FANUC Robotics SYSTEM R-30*i*A HandlingTool Setup and Operations Manual

MAROC77HT01101E REV B

Version 7.70 series

© 2011 FANUC Robotics America Corporation

About This Manual

This manual can be used with controllers labeled R-30*i*A or R-J3*i*C. If you have a controller labeled R-J3*i*C, you should read R-30*i*A as R-J3*i*C throughout this manual. You should also read the "*i*RVision Visual Tracking Start-up Guidance manual" for a deeper understanding of PickTool. PickTool is a value-added application built on top of two key components — *i*RVision Visual Tracking and Line Tracking. This manual contains some descriptions that are taken from the *i*RVision Visual Tracking manual.

Copyrights and Trademarks

This new publication contains proprietary information of FANUC Robotics America Corporation, furnished for customer use only. No other uses are authorized without the express written permission of FANUC Robotics America Corporation.

FANUC Robotics America Corporation
3900 W. Hamlin Road
Rochester Hills, MI 48309-3253

The descriptions and specifications contained in this manual were in effect at the time this manual was approved. FANUC Robotics America Corporation, hereinafter referred to as FANUC Robotics, reserves the right to discontinue models at any time or to change specifications or design without notice and without incurring obligations.

FANUC Robotics manuals present descriptions, specifications, drawings, schematics, bills of material, parts, connections and/or procedures for installing, disassembling, connecting, operating and programming FANUC Robotics' products and/or systems. Such systems consist of robots, extended axes, robot controllers, application software, the KAREL® programming language, INSIGHT® vision equipment, and special tools.

FANUC Robotics recommends that only persons who have been trained in one or more approved FANUC Robotics Training Course(s) be permitted to install, operate, use, perform procedures on, repair, and/or maintain FANUC Robotics' products and/or systems and their respective components. Approved training necessitates that the courses selected be relevant to the type of system installed and application performed at the customer site.

Warning

This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instruction manual, may cause interference to radio communications. As temporarily permitted by regulation, it has not been tested for compliance with the limits for Class A computing devices pursuant to subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference. Operation of the equipment in a residential area is likely to cause interference, in which case the user, at his own expense, will be required to take whatever measure may be required to correct the interference.

FANUC Robotics conducts courses on its systems and products on a regularly scheduled basis at its headquarters in Rochester Hills, Michigan. For additional information contact

FANUC Robotics America Corporation
3900 W. Hamlin Road
Rochester Hills, MI 48309-3253

www.fanucrobotics.com

For customer assistance, including Technical Support, Service, Parts & Part Repair, and Marketing Requests, contact the Customer Resource Center, 24 hours a day, at 1-800-47-ROBOT (1-800-477-6268). International customers should call 011-1-248-377-7159.

Send your comments and suggestions about this manual to:
product.documentation@fanucrobotics.com

The information illustrated or contained herein is not to be reproduced, copied, downloaded, translated into another language, distributed, or published in any physical or electronic format, including Internet, or transmitted in whole or in part in any way without the prior written consent of FANUC Robotics America, Inc.

AccuStat®, ArcTool®, *i*RVision®, KAREL®, PaintTool®,PalletTool®, SOCKETS®, SpotTool®, SpotWorks®, and TorchMate®are Registered Trademarks of FANUC Robotics.

FANUC Robotics reserves all proprietary rights, including but not limited to trademark and trade name rights, in the following names:

AccuAir™, AccuCal™, AccuChop™, AccuFlow™, AccuPath™, AccuSeal™, ARC Mate™, ARC Mate Sr.™, ARC Mate System 1™, ARC Mate System 2™, ARC Mate System 3™, ARC Mate System 4™, ARC Mate System 5™, ARCWorks Pro™, AssistTool™, AutoNormal™, AutoTCP™, BellTool™, BODYWorks™, Cal Mate™, Cell Finder™, Center Finder™, Clean Wall™, DualARM™, LR Tool™, MIG Eye™, MotionParts™, MultiARM™, NoBots™, Paint Stick™, PaintPro™, PaintTool 100™, PAINTWorks™, PAINTWorks II™, PAINTWorks III™, PalletMate™, PalletMate PC™, PalletTool PC™, PayloadID™, RecipTool™, RemovalTool™, Robo Chop™, Robo Spray™, S-420i™, S-430i™, ShapeGen™, SoftFloat™, SOFT PARTS™, SpotTool+™, SR Mate™,

SR ShotTool™, SureWeld™, SYSTEM R-J2 Controller™, SYSTEM R-J3 Controller™, SYSTEM R-J3*i*B Controller™, SYSTEM R-J3*i*C Controller™, SYSTEM R-30*i*A Controller™,TCP Mate™, TorchMate™, TripleARM™, TurboMove™, visLOC™, visPRO-3D™, visTRAC™, WebServer™, WebTP™, and YagTool™.

Patents

One or more of the following U.S. patents might be related to the FANUC Robotics products described in this manual.

FRA Patent List

4,630,567 4,639,878 4,707,647 4,708,175 4,708,580 4,942,539 4,984,745 5,238,029 5,239,739 5,272,805 5,293,107 5,293,911 5,331,264 5,367,944 5,373,221 5,421,218 5,434,489 5,644,898 5,670,202 5,696,687 5,737,218 5,823,389 5,853,027 5,887,800 5,941,679 5,959,425 5,987,726 6,059,092 6,064,168 6,070,109 6,086,294 6,122,062 6,147,323 6,204,620 6,243,621 6,253,799 6,285,920 6,313,595 6,325,302 6,345,818 6,356,807 6,360,143 6,378,190 6,385,508 6,425,177 6,477,913 6,490,369 6,518,980 6,540,104 6,541,757 6,560,513 6,569,258 6,612,449 6,703,079 6,705,361 6,726,773 6,768,078 6,845,295 6,945,483 7,149,606 7,149,606 7,211,978 7,266,422 7,399,363

FANUC LTD Patent List

4,571,694 4,626,756 4,700,118 4,706,001 4,728,872 4,732,526 4,742,207 4,835,362 4,894,596 4,899,095 4,920,248 4,931,617 4,934,504 4,956,594 4,967,125 4,969,109 4,970,370 4,970,448 4,979,127 5,004,968 5,006,035 5,008,834 5,063,281 5,066,847 5,066,902 5,093,552 5,107,716 5,111,019 5,130,515 5,136,223 5,151,608 5,170,109 5,189,351 5,267,483 5,274,360 5,292,066 5,300,868 5,304,906 5,313,563 5,319,443 5,325,467 5,327,057 5,329,469 5,333,242 5,337,148 5,371,452 5,375,480 5,418,441 5,432,316 5,440,213 5,442,155 5,444,612 5,449,875 5,451,850 5,461,478 5,463,297 5,467,003 5,471,312 5,479,078 5,485,389 5,485,552 5,486,679 5,489,758 5,493,192 5,504,766 5,511,007 5,520,062 5,528,013 5,532,924 5,548,194 5,552,687 5,558,196 5,561,742 5,570,187 5,570,190 5,572,103 5,581,167 5,582,750 5,587,635 5,600,759 5,608,299 5,608,618 5,624,588 5,630,955 5,637,969 5,639,204 5,641,415 5,650,078 5,658,121 5,668,628 5,687,295 5,691,615 5,698,121 5,708,342 5,715,375 5,719,479 5,727,132 5,742,138 5,742,144 5,748,854 5,749,058 5,760,560 5,773,950 5,783,922 5,799,135 5,812,408 5,841,257 5,845,053 5,872,894 5,887,122 5,911,892 5,912,540 5,920,678 5,937,143 5,980,082 5,983,744 5,987,591 5,988,850 6,023,044 6,032,086 6,040,554 6,059,169 6,088,628 6,097,169 6,114,824 6,124,693 6,140,788 6,141,863 6,157,155 6,160,324 6,163,124 6,177,650 6,180,898 6,181,096 6,188,194 6,208,105 6,212,444 6,219,583 6,226,181 6,236,011 6,236,896 6,250,174 6,278,902 6,279,413 6,285,921 6,298,283 6,321,139 6,324,443 6,328,523 6,330,493 6,340,875 6,356,671 6,377,869 6,382,012 6,384,371 6,396,030 6,414,711 6,424,883 6,431,018 6,434,448 6,445,979 6,459,958 6,463,358 6,484,067 6,486,629 6,507,165 6,654,666 6,665,588 6,680,461 6,696,810 6,728,417 6,763,284 6,772,493 6,845,296 6,853,881 6,888,089 6,898,486 6,917,837 6,928,337 6,965,091 6,970,802 7,038,165 7,069,808 7,084,900 7,092,791 7,133,747 7,143,100 7,149,602 7,131,848 7,161,321 7,171,041 7,174,234 7,173,213 7,177,722 7,177,439 7,181,294 7,181,313 7,280,687

7,283,661 7,291,806 7,299,713 7,315,650 7,324,873 7,328,083 7,330,777 7,333,879 7,355,725 7,359,817 7,373,220 7,376,488 7,386,367 7,464,623 7,447,615 7,445,260 7,474,939 7,486,816 7,495,192 7,501,778 7,502,504 7,508,155 7,512,459 7,525,273 7,526,121

VersaBell, ServoBell and SpeedDock Patents Pending.

Conventions

This manual includes information essential to the safety of personnel, equipment, software, and data. This information is indicated by headings and boxes in the text.



Warning

Information appearing under WARNING concerns the protection of personnel. It is boxed and in bold type to set it apart from other text.



Caution

Information appearing under **CAUTION** concerns the protection of equipment, software, and data. It is boxed to set it apart from other text.

Note Information appearing next to **NOTE** concerns related information or useful hints.

Contents

About This Manual i

Safety lix

QUICK REFERENCE lxvii

HandlingTool lxix

Chapter 1 OVERVIEW 1–1

1.1 OVERVIEW 1–3

1.2 ROBOT 1–4
1.2.1 Robot Overview 1–4
1.2.2 Robot Models 1–5
1.2.3 HandlingTool End-of-Arm Tooling (EOAT) 1–5
1.2.4 Extended Axes 1–6

1.3 CONTROLLER 1–7
1.3.1 Controller Overview 1–7
1.3.2 Teach Pendant 1–10
1.3.3 Standard Operator Panel 1–10
1.3.4 MODE SELECT Switch 1–12
1.3.5 Robot Stop Variation 1–16
1.3.6 User Operator Panel (UOP) 1–17
1.3.7 CRT/KB 1–17
1.3.8 Emergency Stop Devices 1–17
1.3.9 Communications 1–17
1.3.10 Input/Output (I/O) 1–18
1.3.11 Remote I/O Interfaces 1–19
1.3.12 Motion 1–20
1.3.13 Extended Axes 1–21
1.3.14 Multiple Applications (option) 1–22
1.3.15 Controller Backplane 1–22
1.3.16 Memory 1–23
1.3.17 Line Tracking 1–25

1.4 TEACH PENDANT 1–26
1.4.1 Overview 1–26
1.4.2 Additional *i*Pendant Features 1–39

1.5 FANUC ROBOTICS SOFTWARE 1–65
1.5.1 FANUC Robotics Software Overview 1–65
1.5.2 Set Up 1–66
1.5.3 Program 1–66
1.5.4 Test 1–66
1.5.5 Run Production 1–67

Chapter 2 TURNING ON AND JOGGING THE ROBOT 2–1
2.1 OVERVIEW 2–2
2.2 TURNING ON AND TURNING OFF THE ROBOT 2–2
2.3 JOGGING THE ROBOT 2–5
2.3.1 Overview 2–5
2.3.2 Jog Speed 2–6
2.3.3 Coordinate Systems 2–7
2.3.4 Wrist Jogging 2–10
2.3.5 Remote TCP Jogging 2–11
2.3.6 Motion Groups 2–12
2.3.7 Extended Axes and Sub-Groups 2–13
2.3.8 Jog Menu 2–18

Chapter 3 GENERAL SETUP 3–1
3.1 FRAMES SETUP 3–3
3.1.1 Frames Setup Overview 3–3
3.1.2 Tool Frame 3–6
3.1.3 User Frame 3–27
3.1.4 Remote TCP Frame 3–50
3.1.5 Jog Frame 3–61
3.1.6 Saving Frame Data 3–74
3.2 PRODUCTION OPERATION SETUP 3–76
3.2.1 Production Operation Setup Overview 3–76
3.2.2 Robot Service Request (RSR) Setup 3–82
3.2.3 Program Number Select (PNS) Setup 3–86
3.2.4 Style Name Setup 3–92
3.2.5 OTHER Program Select Mode 3–98
3.3 MACRO COMMANDS 3–99
3.3.1 Overview 3–99
3.3.2 Setting Up Macro Commands 3–99
3.3.3 Executing Macro Commands 3–113
3.3.4 Assigning I/O and Recording Positions in Macro Command Programs 3–117
3.3.5 HandlingTool Macro Commands 3–118
3.4 AXIS LIMITS SETUP 3–119
3.5 BRAKE TIMERS SETUP 3–122
3.6 SETUP GENERAL SETUP SCREEN 3–127
3.6.1 Overview 3–127
3.6.2 Brake on Hold Setup 3–127
3.6.3 Current Language Setup 3–129
3.6.4 Ignore Offset Setup 3–129
3.6.5 Ignore Tool Offset Setup 3–130
3.7 USER ALARM SETUP 3–131
3.8 OVERRIDE SELECT SETUP 3–134
3.9 PASSWORD SETUP 3–137
3.9.1 Password Operations Overview 3–137
3.9.2 Install User Password Operations 3–138
3.9.3 Disabling Passwords 3–142
3.9.4 Program and Set Up User Password Operations 3–143
3.9.5 Configuring Passwords 3–148
3.9.6 Password Log 3–162
3.9.7 Password Level Screen Permissions for HandlingTool 3–165

3.9.8 USB Password Function 3–169
3.9.9 Using KCL with Passwords Enabled 3–171
3.9.10 Password Auto Login Function 3–172
3.10 ERROR SEVERITY TABLE 3–174
3.10.1 Error Severity Table Overview 3–174
3.10.2 Modifying Error Severity 3–174
3.11 ERROR CODE OUTPUT SETUP (OPTION) 3–179
3.11.1 Error Code Output Setup Overview 3–179
3.11.2 Method 1: Output Errors Using 33 Digital Outputs 3–180
3.11.3 Method 2: Output Errors Using 3 Group Outputs 3–185
3.11.4 Output Error Parameters 3–186
3.11.5 Procedure 3–187
3.12 ROBOT PAYLOAD SETTING 3–188
3.12.1 Overview 3–188
3.12.2 Setting the Active Payload Schedule 3–188
3.12.3 Setup of Payload Schedules 3–190
3.12.4 Using Payload Ident. (Option 669) to Set Up Payload Schedules 3–195
3.12.5 Setting Up Arm Load Information 3–209
3.13 STROKE LIMIT SETUP 3–210
3.14 TEACHING KAREL VARIABLES (DispenseTool and HandlingTool) 3–212
3.14.1 Overview 3–212
3.14.2 KAREL Positions 3–212
3.14.3 KAREL Paths 3–215
3.14.4 KAREL Variables 3–224
3.15 SYSTEM CONFIGURATION SETUP 3–226

Chapter 4 INPUT/OUTPUT (I/O) SETUP 4–1
4.1 OVERVIEW 4–3
4.1.1 Overview 4–3
4.1.2 Hardware 4–3
4.1.3 Kinds of I/O 4–3
4.2 SETTING UP I/O 4–5
4.2.1 Overview 4–5
4.2.2 Configuring I/O 4–6
4.2.3 Adding Comments About I/O 4–10
4.2.4 Complementary Output Signals and Polarity 4–10
4.2.5 Simulating I/O 4–10
4.2.6 Controlling Outputs 4–10
4.2.7 Memory Image Port Assignment 4–11
4.2.8 Procedure for Setting Up I/O 4–11
4.2.9 Custom I/O 4–27
4.3 DISTRIBUTED (MODEL B) I/O SETUP 4–29
4.3.1 Overview 4–29
4.3.2 Setting the DIP Switches 4–31
4.3.3 Setting Up the Basic Digital I/O Units 4–34
4.3.4 Setting Up User I/O 4–36
4.4 ROBOT I/O SETUP 4–37
4.5 STANDARD OPERATOR PANEL (SOP) I/O SETUP 4–41
4.6 SETTING UP I/O LINK DEVICES 4–45
4.6.1 Overview 4–45
4.6.2 I/O Link Devices 4–45

4.6.3 Process I/O 4–45
4.6.4 Model A I/O 4–46
4.6.5 Model B I/O 4–47
4.6.6 I/O Link Device Screen 4–47
4.6.7 I/O Link Connection 4–50
4.6.8 FANUC I/O Link Connection Unit 4–52
4.6.9 Setting the Number of Ports 4–60
4.7 I/O INTERCONNECT SETUP 4–62
4.8 CONTROLLING I/O 4–68
4.8.1 Overview 4–68
4.8.2 Forcing Outputs 4–68
4.8.3 Simulating Inputs and Outputs 4–69
4.8.4 Output When WAITing on Input 4–70
4.9 USER OPERATOR PANEL (UOP) SIGNAL DEFINITION 4–72
4.9.1 Overview 4–72
4.9.2 UOP Input Signals 4–75
4.9.3 UOP Output Signals 4–80
4.10 HANDLINGTOOL CELL INTERFACE I/O SIGNALS 4–83
4.10.1 Overview 4–83
4.10.2 HandlingTool Cell Interface I/O Signals 4–84

Chapter 5 PLANNING AND CREATING A PROGRAM 5–1
5.1 OVERVIEW 5–3
5.2 PLANNING A PROGRAM 5–3
5.2.1 Overview 5–3
5.2.2 Motion 5–3
5.2.3 Predefined Positions 5–7
5.3 WRITING AND MODIFYING A PROGRAM 5–15
5.3.1 Overview 5–15
5.3.2 Writing a New Program 5–17
5.3.3 Creating and Writing a New HandlingTool Program 5–22
5.3.4 Modifying a Program 5–28
5.3.5 Setting Up Predefined Positions and Using Them in a Program 5–45
5.3.6 Creating a Macro Program 5–47
5.4 MODIFYING A PROGRAM IN THE BACKGROUND (BACKGROUND EDITING) 5–48
5.4.1 Overview 5–48
5.4.2 Background Edit Process Flow 5–49
5.4.3 Background Editing Using the iPendant 5–51
5.4.4 Modifying a Program in the Background 5–52
5.4.5 Troubleshooting Background Edit 5–54
5.5 SETTING PROGRAM STORAGE CONFIGURATION 5–57
5.5.1 User Interface 5–60
5.5.2 Maximum Program Size 5–62
5.5.3 KAREL Control of STORAGE 5–63
5.5.4 File Formats 5–63
5.5.5 Storage When Loading from a File 5–64
5.5.6 Backup and restore 5–64
5.5.7 Issues relating to SHADOW programs 5–65
5.6 ICON EDITOR 5–65
5.6.1 Using the Icon Editor 5–66
5.6.2 Configuring the Custom Toolbar 5–71

Chapter 6 PROGRAM ELEMENTS 6–1
6.1 OVERVIEW 6–5
6.2 PROGRAM HEADER INFORMATION 6–7
6.2.1 Overview 6–7
6.2.2 Creation Date 6–12
6.2.3 Modification Date 6–13
6.2.4 Copy Source 6–13
6.2.5 Positions and Program Size 6–13
6.2.6 Program Name 6–13
6.2.7 Sub Type 6–14
6.2.8 Program Comment 6–18
6.2.9 Group Mask 6–18
6.2.10 Write Protection 6–19
6.2.11 Ignore Pause 6–20
6.2.12 Stack Size 6–20
6.2.13 Application Mask 6–20
6.2.14 Program Type 6–21
6.2.15 Multi-Arm Main 6–22
6.2.16 Head of Family 6–22
6.2.17 Family Members 6–22
6.3 LINE NUMBER AND PROGRAM END MARKER 6–22
6.4 MOTION INSTRUCTION 6–22
6.4.1 Overview 6–22
6.4.2 Motion Type 6–23
6.4.3 Positional Information 6–28
6.4.4 Position Confirmation 6–29
6.4.5 Motion Status Display 6–30
6.4.6 Frame Number of Positional Data 6–40
6.4.7 Switch Frame Check Type 6–41
6.4.8 Speed 6–42
6.4.9 Termination Type 6–50
6.4.10 Motion Options 6–52
6.4.11 Correspondence Between Teach Pendant Program Motion and KAREL Program Motion 6–117
6.5 REGISTER INSTRUCTIONS 6–118
6.6 POSITION REGISTER INSTRUCTIONS 6–123
6.6.1 Overview 6–123
6.6.2 PR[x] Position Register Instructions 6–123
6.6.3 PR[i,j] Position Register Element Instructions 6–125
6.7 STRING REGISTER INSTRUCTIONS 6–128
6.7.1 Overview 6–128
6.7.2 String Register Assignment and Concatenation 6–128
6.7.3 String Conversion and Precedence 6–130
6.7.4 STRLEN Instruction 6–131
6.7.5 FINDSTR Instruction 6–131
6.7.6 SUBSTR Instruction 6–132
6.8 INPUT/OUTPUT INSTRUCTIONS 6–134
6.8.1 Overview 6–134
6.8.2 Digital Input and Output Instructions 6–134
6.8.3 Robot Digital Input and Output Instructions 6–136
6.8.4 Analog Input and Output Instructions 6–137
6.8.5 Group Input and Output Instructions 6–138
6.9 BRANCHING INSTRUCTIONS 6–139

6.9.1 Overview 6–139
6.9.2 Label Definition Instruction **LBL[x]** 6–140
6.9.3 Unconditional Branching Instructions 6–140
6.9.4 Conditional Branching Instructions 6–142
6.10 WAIT INSTRUCTIONS 6–146
6.11 MISCELLANEOUS INSTRUCTIONS 6–151
6.11.1 Overview 6–151
6.11.2 RSR Enable/Disable Instruction 6–151
6.11.3 User Alarm Instruction 6–151
6.11.4 Timer Instruction 6–152
6.11.5 OVERRIDE Instruction 6–152
6.11.6 Remark Instruction 6–153
6.11.7 Multi-lng Remark Instruction 6–153
6.11.8 Line Remark (Comment out) Instruction 6–154
6.11.9 Message Instruction 6–155
6.11.10 Parameter Name Instruction 6–155
6.11.11 Maximum Speed Instruction 6–157
6.12 SKIP INSTRUCTION 6–159
6.13 QUICK SKIP (HIGH-SPEED SKIP) 6–162
6.13.1 Overview 6–162
6.13.2 Execution 6–162
6.13.3 Limitations 6–163
6.14 OFFSET/FRAME INSTRUCTIONS 6–163
6.15 TOOL OFFSET CONDITION INSTRUCTION 6–166
6.16 MULTIPLE CONTROL INSTRUCTIONS 6–167
6.17 MACRO COMMAND INSTRUCTION 6–168
6.18 PARAMETERS FOR PROGRAM CALL AND MACRO INSTRUCTIONS 6–169
6.18.1 Overview 6–169
6.18.2 Parameter Instruction Syntax 6–170
6.18.3 String Value Selections 6–172
6.18.4 Argument Registers 6–174
6.18.5 Guidelines for Using Parameters 6–175
6.18.6 Including Parameters in Program Call and Macro Instructions 6–177
6.18.7 Including Argument Registers in Sub-Programs 6–181
6.19 PROGRAM CONTROL INSTRUCTIONS 6–182
6.19.1 Overview 6–182
6.19.2 PAUSE Instruction 6–183
6.19.3 ABORT Instruction 6–183
6.19.4 Error Program Instruction 6–183
6.19.5 Resume Program Instruction 6–184
6.19.6 Maintenance Program Instruction 6–184
6.19.7 Clear Resume Program Instruction 6–185
6.19.8 Return Path Disable Instruction 6–185
6.20 **MOTION GROUP INSTRUCTIONS (OPTION)** 6–186
6.20.1 Overview 6–186
6.20.2 Independent Motion Group Instructions 6–186
6.20.3 Simultaneous Motion Group Instructions 6–186
6.20.4 Defining Motion Group Instructions 6–187
6.21 POSITION REGISTER LOOK-AHEAD INSTRUCTIONS 6–189
6.22 CONDITION MONITOR INSTRUCTIONS 6–190

6.23 PAYLOAD INSTRUCTION 6–194
6.24 COLLISION GUARD INSTRUCTIONS (OPTION) 6–196
6.25 BASIC PROCESS AXES INSTRUCTIONS (OPTION) 6–197
6.25.1 Overview 6–197
6.25.2 SET ISDT SPEED Instruction 6–198
6.25.3 STOP ALL ISDT Instruction 6–199
6.26 PALLETIZING INSTRUCTIONS 6–200
6.26.1 Overview 6–200
6.26.2 PALLETIZING-B Instruction 6–203
6.26.3 PALLETIZING-BX Instruction 6–209
6.26.4 PALLETIZING-E Instruction 6–216
6.26.5 PALLETIZING-EX Instruction 6–224
6.26.6 PALLETIZING-END Instruction 6–232
6.27 **ALL-POINT TEACHING FOR PALLETIZING** 6–232
6.28 PALLET REGISTER INSTRUCTIONS 6–235
6.29 TOUCH SENSE INSTRUCTIONS 6–237
6.29.1 Overview 6–237
6.29.2 Search Start Instruction 6–238
6.29.3 Search End Instruction 6–239
6.29.4 Touch Offset Instruction 6–239
6.29.5 Touch Offset End Instruction 6–240
6.30 MIXED LOGIC INSTRUCTIONS 6–241
6.30.1 Overview 6–241
6.30.2 Data Types 6–241
6.30.3 Operators 6–242
6.30.4 Expressions 6–243
6.30.5 Adding Mixed Logic Instructions 6–245
6.30.6 Background Logic 6–247
6.30.7 Backup/Restore 6–260
6.31 PROCESS SYNCHRONIZATION 6–260
6.32 VISION INSTRUCTIONS 6–263
6.32.1 Overview 6–263
6.32.2 RUN_FIND 6–264
6.32.3 GET_OFFSET 6–264
6.32.4 SET_REFERENCE 6–265
6.32.5 CAMERA_CALIB 6–265
6.32.6 VR[].OFFSET 6–265
6.32.7 VR[]. MODELID 6–266
6.32.8 SAMPLE PROGRAM 6–266

Chapter 7 TESTING A PROGRAM AND RUNNING PRODUCTION 7–1
7.1 OVERVIEW 7–2
7.2 PROGRAM PAUSE AND RECOVERY 7–3
7.2.1 Program Pause and Recovery Overview 7–3
7.2.2 EMERGENCY STOP and Recovery 7–3
7.2.3 HOLD and Recovery 7–5
7.2.4 Setting Tolerance for Resuming a Program 7–6
7.2.5 Disable Fault Checking 7–11
7.3 TEST CYCLE 7–16
7.3.1 Test Cycle Overview 7–16
7.3.2 Test Cycle Setup 7–16

7.3.3 Single Step Testing 7–21
7.3.4 Continuous Testing 7–28
7.3.5 Monitoring Programs 7–34
7.4 RELEASE WAIT 7–34
7.5 PRODUCTION OPERATION 7–35
7.5.1 Production Operation Overview 7–35
7.5.2 Standard Operator Panel Cycle Start Production 7–38
7.5.3 User Operator Panel Start 7–39
7.5.4 Robot Service Request (RSR) Production Start 7–41
7.5.5 Program Number Select (PNS) and UOP Production Start 7–42
7.6 ADJUSTING A PROGRAM (PROG ADJUST) 7–44
7.6.1 Overview 7–44
7.6.2 Adjusting a Program or a Schedule 7–44
7.6.3 Error Handling 7–52
7.7 MAINTENANCE AND REPAIR 7–54

Chapter 8 STATUS DISPLAYS AND INDICATORS 8–1
8.1 STATUS INDICATORS 8–3
8.1.1 Overview 8–3
8.1.2 Teach Pendant Status Indicators 8–3
8.1.3 Standard Operator Panel Status Indicators 8–5
8.2 USER SCREEN STATUS 8–7
8.3 REGISTER STATUS 8–8
8.4 POSITION REGISTER STATUS 8–10
8.5 STRING REGISTER STATUS 8–14
8.6 PALLET REGISTER STATUS 8–17
8.7 SYSTEM VARIABLE STATUS 8–18
8.8 SAFETY SIGNAL STATUS 8–20
8.9 PROGRAM TIMER STATUS 8–22
8.10 SYSTEM TIMER STATUS 8–25
8.11 CLOCK STATUS 8–26
8.12 VERSION IDENTIFICATION STATUS 8–27
8.13 MEMORY STATUS 8–31
8.13.1 Checking Memory 8–31
8.14 POSITION STATUS 8–33
8.15 TURN NUMBER DISPLAY 8–36
8.15.1 Overview 8–36
8.15.2 Usual Configuration 8–36
8.15.3 $SCR_GRP[group].$turn_axis[i] System Variable 8–39
8.16 EXECUTION HISTORY 8–40
8.17 PROCESS AXES STATUS (OPTION) 8–42
8.18 PROGRAM or PRODUCTION STATUS 8–43
8.19 ROBOT AXES STATUS 8–45
8.19.1 Overview 8–45
8.19.2 Status 1 Screen 8–45
8.19.3 Status 2 Screen 8–49
8.19.4 Pulse Screen 8–50

8.19.5 Torque Monitor Screen 8–51
8.19.6 Tracking Screen 8–52
8.19.7 Disturbance Torque Screen 8–53
8.19.8 Servo Diagnosis 8–53
8.19.9 Procedure 8–60
8.20 PROGRAM MONITOR STATUS 8–61
8.21 VISUAL DIAGNOSTICS 8–62
8.21.1 Overview 8–62
8.21.2 Creating and Editing Visual Diagnostic Screens 8–64
8.21.3 Managing Screens 8–78
8.21.4 Viewing Visual Diagnostic Screens 8–81

Chapter 9 PROGRAM AND FILE MANIPULATION 9–1
9.1 STORAGE DEVICES 9–3
9.1.1 Overview 9–3
9.1.2 Storage Device Hints 9–9
9.1.3 Setting the Default Device 9–10
9.1.4 Setting Up a Port 9–12
9.1.5 Using a Memory Card or USB Memory Stick Interface 9–19
9.1.6 Formatting Devices 9–23
9.1.7 Subdirectories 9–26
9.2 MANIPULATING PROGRAMS 9–30
9.2.1 Overview 9–30
9.2.2 Selecting Programs On the SELECT Menu 9–31
9.2.3 Saving Programs 9–32
9.2.4 Loading Programs 9–34
9.2.5 Copying Programs Within the SELECT Menu 9–36
9.2.6 Deleting Programs from the SELECT Menu 9–37
9.2.7 Saving Files as ASCII 9–39
9.2.8 Filtering Programs in the Select List 9–41
9.3 MANIPULATING FILES 9–42
9.3.1 Overview 9–42
9.3.2 Altering the FILE Menu View 9–47
9.3.3 Generating a Directory of Files 9–50
9.3.4 Backing Up Files 9–52
9.3.5 Loading and Restoring Files to Controller Memory 9–58
9.3.6 Conversion of PC Files 9–64
9.3.7 Auto Software Update 9–65
9.3.8 Displaying Text (ASCII) Files 9–66
9.3.9 Copying Files 9–67
9.3.10 Deleting Files 9–70
9.3.11 Saving Files 9–71
9.3.12 Checking File Memory 9–74
9.4 AUTOMATIC BACKUP 9–75
9.4.1 Overview 9–75
9.4.2 Automatic Backup 9–75
9.5 TROUBLESHOOTING FILE BACKUP OR AUTOMATIC BACKUP ERRORS 9–80
9.6 IMAGE BACKUP 9–82
9.6.1 Overview 9–82
9.6.2 Performing an Image Backup 9–83
9.6.3 Restoring an Image Backup 9–86
9.7 ASCII UPLOAD 9–89

9.7.1 Overview 9–89
9.7.2 Loading an ASCII Teach Pendant Program From the Teach Pendant 9–90
9.7.3 Loading an ASCII Teach Pendant Program from KCL 9–93
9.7.4 Viewing ASCII Upload Errors 9–94
9.7.5 Example ASCII File 9–95

Chapter 10 ADVANCED FUNCTIONS 10–1

10.1 ADAPTIVE PROCESS CONTROL 10–7
10.1.1 Overview 10–7
10.1.2 Setting Up Input Parameters 10–9
10.1.3 Setting Up Output Parameters 10–10
10.1.4 Setting Up Map Functions 10–13
10.1.5 Setting Up Options 10–15
10.1.6 Adding the APC Instructions To a Program 10–16
10.2 ANGLE ENTRY SHIFT FUNCTION (OPTION NOT AVAILABLE IN NORTH AMERICA) 10–17
10.3 AUTO SINGULARITY AVOIDANCE 10–23
10.3.1 Overview 10–23
10.3.2 How it Works 10–25
10.3.3 Compatibility 10–26
10.3.4 Limitations 10–26
10.3.5 Cautions 10–27
10.3.6 How to Use Auto Singularity Avoidance 10–30
10.3.7 System Variable Setup 10–37
10.3.8 Troubleshooting 10–37
10.4 AUTOMATIC COLLISION RECOVERY 10–40
10.4.1 Overview 10–40
10.4.2 Enabling and Disabling Automatic Collision Recovery 10–41
10.4.3 Automatic Collision Recovery Setup 10–41
10.4.4 Mode 1 of Automatic Collision Recovery 10–43
10.4.5 Mode 2 of Automatic Collision Recovery 10–45
10.4.6 Limitations 10–46
10.5 BASIC PROCESS AXES (OPTION) 10–47
10.5.1 Overview 10–47
10.5.2 Setup 10–48
10.5.3 TP Instructions 10–51
10.5.4 PROCESS AXES STATUS (OPTION) 10–54
10.5.5 Torque Monitor Function 10–55
10.6 BRAKE CHECK FUNCTION 10–56
10.6.1 Overview 10–56
10.6.2 Setup 10–56
10.6.3 Start Brake Check 10–57
10.6.4 Result of Brake Check 10–58
10.6.5 Limitations 10–58
10.6.6 Caution 10–59
10.7 BUMP BOX 10–59
10.7.1 Overview 10–59
10.7.2 Setting Up Bump Box Schedules 10–60
10.7.3 Setting Up Bump Box I/O 10–67
10.7.4 BumpBox with Coordinated Motion - Overview 10–69
10.7.5 Bump Frames for Normal and Coordinated Motion 10–73
10.8 CENTER FINDER 10–82

10.8.1 Overview 10–82
10.8.2 Using Center Finder 10–82
10.8.3 Troubleshooting 10–91
10.9 COLLISION GUARD 10–91
10.9.1 Overview 10–91
10.9.2 Limitations 10–92
10.9.3 Falsely Detected Collisions 10–92
10.9.4 Setup 10–93
10.9.5 Programmed Motion 10–95
10.10 COLLISION SKIP 10–98
10.10.1 Overview 10–98
10.10.2 Requirements 10–98
10.10.3 Operation 10–99
10.11 CONDITION MONITOR FUNCTION 10–103
10.11.1 Overview 10–103
10.11.2 Monitors 10–104
10.11.3 Monitor State 10–105
10.11.4 Monitor Instructions 10–106
10.11.5 Condition Handler Program 10–107
10.11.6 Conditions 10–107
10.11.7 Condition Menu 10–108
10.11.8 Restrictions 10–111
10.12 COORDINATES OFFSET FUNCTION 10–117
10.12.1 Using the Coordinates Offset Function 10–117
10.12.2 Tool Frame Offset Function 10–121
10.12.3 User Frame Offset Function 10–125
10.13 CYCLE TIME LOGGING AND DISPLAY 10–129
10.13.1 Overview 10–129
10.13.2 Modes 10–131
10.13.3 Tracking Cycle Time 10–136
10.14 DATA MONITOR 10–139
10.14.1 Overview 10–139
10.14.2 Data Monitor Setup 10–142
10.14.3 Data Monitor Schedule 10–148
10.14.4 Data Monitor Programming 10–153
10.15 DIAGNOSTIC VIDEO MONITORING 10–154
10.15.1 Overview 10–154
10.15.2 DVM Setup 10–155
10.15.3 Controlling DVM Execution 10–161
10.15.4 Displaying DVM and DVM Logs 10–162
10.16 DISTANCE BEFORE 10–165
10.16.1 Overview 10–165
10.16.2 Specification 10–165
10.16.3 Program Instruction 10–167
10.16.4 Distance Value 10–167
10.16.5 Action 10–169
10.16.6 Changing the Trigger Condition 10–170
10.16.7 Alarms Posted When Distance Before is Not Triggered 10–172
10.16.8 Single Step Execution 10–173
10.16.9 Hold and Resume 10–173
10.16.10 Resume After Jogging 10–176
10.16.11 Power Failure Recovery 10–177
10.16.12 Adding the Distance Before Motion Option (Procedure) 10–177

10.16.13 Cautions and Limitations 10–180
10.16.14 System Variables 10–181
10.16.15 Error Codes 10–182

10.17 EASY NORMAL UTILITY (formerly known as Auto Normal Utility) 10–183
10.17.1 Overview 10–183
10.17.2 Using the Easy Normal Utility 10–184
10.17.3 Setup 10–187
10.17.4 Execution 10–189

10.18 ENHANCED MIRROR IMAGE (OPTION) 10–190

10.19 ERROR RECOVERY (OPTION) 10–209
10.19.1 Overview 10–209
10.19.2 Features 10–210
10.19.3 Limitations 10–211
10.19.4 I/O Interface 10–212
10.19.5 Setup 10–214
10.19.6 Programming 10–220
10.19.7 Testing 10–224
10.19.8 Error Recovery Manual Function 10–224
10.19.9 I/O Timing Sequence 10–227

10.20 EXECUTING MULTIPLE PROGRAMS (MULTI-TASKING) 10–232
10.20.1 Overview 10–232
10.20.2 Guidelines 10–233
10.20.3 Synchronizing the Execution of Multiple Programs 10–234
10.20.4 Effect of Multi-Tasking on Dedicated I/O Signals 10–234
10.20.5 Standard Operator Panel (SOP) Cycle Start Execution 10–234
10.20.6 Program Number Select (PNS) Execution 10–235
10.20.7 RUN Program Instruction Execution 10–237
10.20.8 Single Step Program Execution 10–237

10.21 EXTENDED USER FRAMES 10–239
10.21.1 Configuring the Number of Frames 10–240
10.21.2 Effect on the Jog Menu 10–240
10.21.3 Programming with Extended User Frames 10–241

10.22 GRAVITY COMPENSATION 10–242
10.22.1 System Variables 10–242
10.22.2 MOTION Screen 10–243

10.23 GROUP MASK EXCHANGE 10–244
10.23.1 Overview 10–244
10.23.2 Setting Up Group Mask Exchange 10–245
10.23.3 Troubleshooting Group Mask Exchange 10–246

10.24 HIGH SPEED INPUT FUNCTIONS 10–248
10.24.1 Overview 10–248
10.24.2 Frequency Interface 10–248
10.24.3 High Speed Scanning (for Line Tracking Part Detect). 10–248
10.24.4 Position Snap & Map 10–249

10.25 LIMIT SET 10–249

10.26 MENU UTILITY (OPTION) 10–252
10.26.1 Overview 10–252
10.26.2 Prompt Box Message Menu 10–253
10.26.3 Prompt Box Yes/No Menu 10–259
10.26.4 List Menu 10–265
10.26.5 Status Menu 10–273
10.26.6 Operator Entry Menu 10–285

10.26.7 Using Your Teach Pendant Menus 10–294
10.27 MIRROR IMAGE UTILITY 10–295
10.28 MONITORING PROGRAMS 10–311
10.28.1 Montioring Teach Pendant Programs 10–311
10.29 MOTION GROUP DO OUTPUT FUNCTION (MULTIPLE MOTION GROUP OPTION) 10–313
10.29.1 Overview 10–313
10.29.2 Restrictions 10–314
10.29.3 Operations 10–314
10.29.4 Setup 10–315
10.29.5 Subprogram and Multi-tasking Execution 10–316
10.30 MOTION START DELAY DETECTION FUNCTION 10–317
10.30.1 Overview 10–317
10.30.2 Restrictions 10–317
10.30.3 Detection Information and System Variables 10–318
10.30.4 Motion Start Delay Detection Function Example 10–319
10.31 OFFLINE SINGULARITY AVOIDANCE (Formerly Singularity Avoidance) 10–320
10.31.1 Overview 10–320
10.31.2 How it Works 10–321
10.31.3 Compatibility 10–323
10.31.4 Limitations 10–324
10.31.5 Procedure 10–324
10.31.6 Troubleshooting 10–329
10.32 OPERATION LOG BOOK [OPTION] 10–330
10.32.1 Overview 10–330
10.32.2 Recorded Events 10–333
10.32.3 Operations 10–339
10.33 ORIGINAL PATH RESUME 10–344
10.34 PAYLOAD CONFIRM 10–348
10.34.1 Overview 10–348
10.34.2 Payload Setting and Payload Over Confirm 10–349
10.35 POSITION REGISTER LOOK-AHEAD EXECUTION FUNCTION 10–353
10.35.1 Overview 10–353
10.35.2 Program Instructions 10–354
10.35.3 Program Example 10–355
10.35.4 Execution 10–356
10.36 PROGRAM SHIFT UTILITY 10–356
10.37 REFERENCE POSITION UTILITY 10–368
10.38 RESUME OFFSET 10–372
10.38.1 Limitations of Resume Offset 10–372
10.38.2 Constant Path Resume Offset 10–373
10.38.3 Limitations of Constant Path Resume Offset 10–373
10.38.4 Non-Constant Path Resume Offset 10–374
10.38.5 Limitations of non-Constant Path Resume Offset 10–374
10.38.6 Selecting Resume Offset Type 10–375
10.38.7 Resume Offset Setup Menu 10–375
10.38.8 System Variables 10–377
10.39 ROBOT-PC COMMUNICATIONS MACROS 10–378
10.39.1 Overview 10–378
10.39.2 Send Event ( ) 10–378
10.39.3 Send Data( ) 10–380

10.39.4 Send SysVar( ) 10–381
10.39.5 Get Data( ) 10–383
10.39.6 Request Menu( ) 10–384
10.40 SAFETY SPACE 10–385
10.40.1 Overview 10–385
10.40.2 Requirements 10–385
10.40.3 Operation 10–386
10.41 SERVO TOOL (Option) 10–387
10.41.1 Benefits 10–387
10.41.2 Limitations 10–388
10.41.3 Program Instruction 10–388
10.41.4 Setup 10–390
10.41.5 Detection 10–391
10.42 SHAPE GENERATION II (OPTION) 10–393
10.42.1 Overview 10–393
10.42.2 Quick Start Hints 10–394
10.42.3 Shape Setup 10–395
10.42.4 Setting Up Default Shape Data 10–397
10.42.5 Shape-Specific Shape Data for Standard Shapes 10–400
10.42.6 Setting Up Custom Shape Data 10–413
10.42.7 Generating a Shape Program from UTILITIES Shape Gen 10–416
10.42.8 Servo Schedules 10–424
10.42.9 Tuning Guidelines 10–426
10.43 SIMPLE TEACH SUITE 10–429
10.43.1 Overview 10–429
10.43.2 Auto Path Smoothing 10–430
10.44 SIMULATED INPUT SKIP 10–440
10.44.1 Overview 10–440
10.44.2 Setup 10–440
10.44.3 Operation 10–443
10.45 SINGULARITY CHECK FUNCTION 10–445
10.46 SOFT FLOAT (OPTION) 10–446
10.46.1 Overview 10–446
10.46.2 Restrictions 10–447
10.46.3 Setting Up Joint Soft Float Schedules 10–449
10.46.4 Setting Up Cartesian Soft Float Schedules 10–451
10.46.5 Setting Up Pushout Soft Float Schedules 10–453
10.46.6 Program Instructions 10–456
10.46.7 General Soft Float Guidelines 10–459
10.46.8 Cartesian Soft Float Guidelines 10–462
10.46.9 Other Guidelines 10–468
10.47 SPACE CHECK FUNCTION (OPTION) 10–470
10.47.1 Setting Up Space Check 10–474
10.47.2 Limitations 10–481
10.48 TCP SPEED OUTPUT (OPTION) 10–482
10.48.1 Overview 10–482
10.48.2 Setting Up TCP Speed Output 10–483
10.48.3 TCP Speed Output Program Instruction 10–485
10.49 TCP SPEED PREDICTION (OPTION) 10–487
10.49.1 Overview 10–487
10.49.2 Setting Up and Running TCP Speed Prediction 10–488
10.49.3 System Variables 10–489

10.50 TEACH PENDANT SHIM 10–501
10.50.1 Overview 10–501
10.50.2 Setting Up and Using TP Shim 10–504
10.50.3 Using TP Shim History to Apply Previous Shims 10–507
10.51 TIME BEFORE/AFTER MOTION OPTION INSTRUCTION 10–510
10.51.1 Overview 10–510
10.51.2 Program Execution 10–510
10.51.3 Execution Timing 10–513
10.51.4 Recording a TIME BEFORE/AFTER Instruction 10–514
10.51.5 TIME BEFORE Instruction Program Example 10–516
10.51.6 Programming Hints 10–517
10.52 TORQUE LIMIT 10–518
10.52.1 Introduction 10–518
10.52.2 Specification 10–519
10.52.3 Restriction 10–520

Chapter 11 SERVO TOOL CHANGE FUNCTION 11–1

11.1 OVERVIEW 11–3
11.1.1 1-1 Feature of function 11–3
11.1.2 1-2 Basic specification 11–3
11.1.3 Restrictions 11–3
11.1.4 1-4 System Configuration 11–4
11.1.5 1-5 Outline of installation 11–6
11.2 INITIAL SETUP 11–7
11.3 PRELIMINARY TOOL ATTACH OPERATION 11–7
11.4 TOOL CHANGE SETUP 11–9
11.5 SETTING THE REFERENCE POSITION 11–17
11.5.1 Battery-less type tools 11–17
11.5.2 Battery-mounted type tools 11–17
11.6 TOOL CHANGE INSTRUCTION 11–18
11.6.1 TOOL DETACH instruction 11–18
11.6.2 TOOL ATTACH instruction 11–18
11.6.3 Sample program 11–19
11.6.4 Foward execution 11–19
11.6.5 Backward execution 11–20
11.7 TOOL CHANGE SEQUENCE 11–21
11.8 TOOL CHANGE STATUS 11–21
11.9 TEACHING 11–22
11.9.1 Notice for teaching 11–22
11.9.2 Sample program 11–23
11.10 CONSTRUCTION OF SERVO TOOL CHANGE SCREEN 11–24
11.10.1 INITIAL SETUP SCREEN This screen can be displayed at CONTROL START Menu. 11–24
11.10.2 SETUP SCREEN 11–24
11.10.3 - STATUS SCREEN 11–25
11.11 TOOL CHANGE INITIAL SETUP 11–25
11.11.1 Setting Motion Parameters For Servo Tool Axes 11–25
11.12 TOOL CHANGE REFERENCE POSITION SETUP METHOD (BATTERY-LESS TYPE) 11–27
11.12.1 Reference Position Setup for Calibration Types 3 and 4 11–27
11.12.2 Reference Position Setup for Calibration Types 5 and 6 11–30

11.12.3 Quick Mastering Reference Position Setup 11–31
11.13 TROUBLESHOOTING 11–32
11.13.1 The Attach Instruction Is Executed When the Tool Is Not Attached. 11–32
11.13.2 The Robot Stopped During Calibration. 11–33
11.13.3 Calibration Motion Failed. 11–34
11.13.4 A Different Tool from That Specified by the Attach Instruction Is Attached. 11–34
11.13.5 The Attached Tool Has Been Detached by Mistake (Without Using the Detach Instruction). 11–35
11.13.6 TheTool Axis of a Detached Tool Has Moved. 11–35
11.13.7 The Battery Ran Low While the Tool Was Detached. 11–36

Chapter 12 FANUC SENSOR INTERFACE 12–1
12.1 OVERVIEW 12–2
12.2 COMMUNICATION PROTOCOL 12–2
12.2.1 Overview 12–2
12.2.2 Flow Control Mechanism 12–2
12.2.3 Data Format 12–3
12.2.4 Handshaking 12–5
12.3 PROGRAMMING 12–8
12.4 SENSOR SETUP AND HARDWARE CONNECTIONS 12–9

Chapter 13 MATERIAL HANDLING OPTION 13–1
13.1 OVERVIEW 13–2
13.2 SETTING UP THE MATERIAL HANDLING TOOLS 13–3
13.2.1 Overview 13–3
13.2.2 Selecting the Handling Config (SpotTool+ only) 13–3
13.2.3 Setting Up Tooling Valves 13–6
13.2.4 Setting Up Tooling Valve Signals 13–12
13.2.5 Configuring Tooling Valve I/O 13–17
13.2.6 Controlling Valve I/O 13–19
13.3 CONTROLLING A TOOLING VALVE IN MANUAL MODE 13–22
13.4 MATERIAL HANDLING VALVE MACRO PROGRAMS 13–23
13.4.1 Examples 13–24
13.4.2 Grip Part 13–25
13.4.3 Release Part 13–27
13.4.4 Part Present 13–29
13.4.5 Check No Part 13–30
13.4.6 Prepare to Pick 13–31
13.4.7 Clear to Proceed 13–32
13.4.8 Turn ON Vacuum 13–33
13.4.9 Turn OFF Vacuum 13–36
13.4.10 Turn OFF Blowoff 13–37
13.4.11 Set CurrentValve 13–39
13.4.12 Set MH Tool 13–40
13.5 PRODUCTION 13–41
13.5.1 MH Tryout Mode 13–41
13.5.2 Disable for 20 Cycles 13–41
13.5.3 Alarms 13–42

Chapter 14 MACHINE TOOL OPTIONS 14–1
14.1 OVERVIEW 14–2

14.1.1 Overview 14–2
14.2 MACHINE TOOL EASY SETUP 14–2
14.2.1 Overview 14–2
14.2.2 User Interface 14–5
14.2.3 Machine Tool Programs 14–10
14.2.4 Machine Tool Data 14–18
14.2.5 Customizing Machine Tool Easy Setup 14–25

Chapter 15 *i*RCALIBRATION OPTIONS 15–1
15.1 OVERVIEW 15–3
15.1.1 Overview 15–3
15.1.2 Torque Observer Sensor and Electrical Continuity Sensing 15–5
15.1.3 *i*RCalibration TP Instructions 15–6
15.2 *I*RCALIBRATION MASTER SET 15–8
15.2.1 Overview 15–8
15.2.2 Preliminary Setup 15–9
15.2.3 *i*RCalibration Master Set Setup 15–12
15.2.4 Creating a Robot Calibration Teach Pendant Program 15–22
15.2.5 Running the *i*RCalibration Master Set Program 15–35
15.2.6 Updating the Master Count 15–37
15.2.7 Viewing the Calibration Result 15–38
15.2.8 Troubleshooting *i*RCalibration Master Set 15–40
15.3 *i*RCALIBRATION MASTER RECOVERY 15–42
15.3.1 Overview 15–42
15.3.2 Preliminary Setup 15–43
15.3.3 *i*RCalibration Master Recovery Setup 15–46
15.3.4 Creating a Robot Calibration Teach Pendant Program 15–57
15.3.5 Running the *i*RCalibration Master Recovery Program 15–70
15.3.6 Single Axis Mastering Recovery 15–72
15.3.7 Updating the Master Count 15–73
15.3.8 Viewing the Calibration Result 15–74
15.3.9 Troubleshooting *i*RCalibration Mastering 15–76
15.4 iRCALIBRATION TCP SET 15–78
15.4.1 Overview 15–78
15.4.2 Preliminary Setup 15–78
15.4.3 *i*RCalibration TCP Set Setup 15–83
15.4.4 Running the *i*RCalibration TCP Set Program 15–103
15.4.5 Displaying the UTool Log Screen 15–105
15.4.6 *i*RCalibration TCP Set Error Recovery 15–107
15.4.7 *i*RCalibration TCP Set Troubleshooting 15–108
15.5 iRCALIBRATION TCP SHIFT 15–109
15.5.1 Overview 15–109
15.5.2 Preliminary Setup 15–109
15.5.3 *i*RCalibration TCP Shift Setup 15–113
15.5.4 Running the *i*RCalibration TCP Shift Program 15–133
15.5.5 Displaying the UTool Log Screen 15–135
15.5.6 *i*RCalibration TCP Shift Error Recovery 15–137
15.5.7 *i*RCalibration TCP Shift Troubleshooting 15–138
15.6 *i*RCALIBRATION FRAME SET 15–139
15.6.1 Overview 15–139
15.6.2 Preliminary Setup 15–140
15.6.3 *i*RCalibration Frame Set Setup 15–140
15.6.4 Running the *i*RCalibration Frame Set Program and Updating the User Frame 15–156

15.6.5 Using *i*RCalibration Frame Set with Nominal Data 15–158
15.6.6 Displaying the UFrame Log Screen 15–158
15.6.7 Troubleshooting *i*RCalibration Frame Set 15–160
15.7 iRCALIBRATION FRAME SHIFT 15–163
15.7.1 Overview 15–163
15.7.2 Preliminary Setup 15–164
15.7.3 *i*RCalibration Frame Shift Setup 15–167
15.7.4 Running the *i*RCalibration Frame Shift Program and Updating the User Frame 15–188
15.7.5 Using *i*RCalibration Frame Shift with Dynamic-Uframe 15–191
15.7.6 Displaying the UFrame Log Screen 15–191
15.7.7 Troubleshooting *i*RCalibration Frame Shift 15–193

Chapter 16 INTERFERENCE CHECKING 16–1

16.1 OVERVIEW 16–3
16.1.1 Overview 16–3
16.1.2 Collision Avoidance 16–3
16.1.3 Multiple Controller Interference Checking 16–5
16.1.4 DualARM Single Controller Interference Checking 16–6
16.2 MODELING 16–7
16.2.1 Overview 16–7
16.2.2 Robot Arm Modeling 16–13
16.2.3 User Tool Modeling 16–18
16.2.4 Obstacle Modeling 16–26
16.2.5 How to Teach a Plane Obstacle Effectively 16–32
16.2.6 How to Teach a Box Obstacle Effectively 16–35
16.3 SETTING UP COLLISION CHECKING BETWEEN ROBOT ARM AND STATIC OBSTACLES 16–36
16.3.1 Overview 16–36
16.3.2 Setting Up Combination Pairs 16–37
16.4 SETTING UP INTERFERENCE CHECKING FOR MULTIPLE ROBOT APPLICATIONS 16–39
16.4.1 Overview 16–39
16.4.2 Host Robot and Priority List 16–39
16.4.3 Combination Check Pairs 16–43
16.4.4 Automatic Setup of Default Interference Zone Between Robot Pairs 16–47
16.5 SETTING UP DUALARM SINGLE CONTROLLER INTERFERENCE CHECKING 16–57
16.6 INTERFERENCE CHECK TP INSTRUCTIONS 16–58
16.7 STATUS AND RECOVERY 16–59
16.7.1 Overview 16–59
16.7.2 Recovery and Interference Checking 16–60
16.8 ROBOT PARAMETERS 16–61

Chapter 17 INTERFERENCE AVOIDANCE 17–1

17.1 Overview 17–2
17.2 Define Robot and Tool Model 17–3
17.3 Setting Up the AutoZone Space 17–5
17.4 Set Up the AutoZone Manager Member List and Calibrate Each Member 17–8
17.5 Set Up the AutoZone Schedule 17–10
17.6 Master Each Program Which is to be Protected from Collision Example 17–12

17.7 AutoZone Status 17–14
17.8 AutoZone Graphical Status 17–15
17.9 Interference Avoidance Limitations 17–20

Chapter 18 TOUCH SENSING 18–1
18.1 OVERVIEW 18–3
18.2 ASSIGNING TOUCH SENSING I/O 18–4
18.2.1 Touch Sensing I/O Overview 18–4
18.2.2 Touch Sensing Input Signal 18–5
18.2.3 Touch Sensing Enable/Disable Output Signal 18–5
18.2.4 Assigning the Touch Sensing Inputs and Outputs 18–6
18.3 SETTING UP TOUCH SENSING 18–7
18.3.1 Touch Sensing Setup Overview 18–7
18.3.2 Touch Frames 18–10
18.3.3 Search Pattern 18–16
18.3.4 Touch Schedule 18–22
18.4 TOUCH SENSING PROGRAMMING 18–28
18.4.1 Programming Overview 18–28
18.4.2 Touch Sensing Instructions 18–28
18.4.3 Touch Sensing Motion Option 18–29
18.4.4 Motion Instructions Used with Touch Sensing 18–30
18.4.5 Executing a Touch Sensing Program 18–30
18.4.6 Touch Sensing Robot Position Touchup 18–31
18.4.7 Programming Examples 18–32
18.5 TOUCH SENSING MASTERING 18–38
18.5.1 Mastering Overview 18–38
18.5.2 Mastering 18–38
18.5.3 Remastering 18–39
18.5.4 Offsets 18–40
18.5.5 Patterns 18–41
18.5.6 Master Flag 18–42
18.5.7 Touching Up Path Positions 18–44
18.5.8 Adding New Positions 18–46
18.5.9 Multiple Searches 18–47
18.5.10 Touching Up Search Start Positions 18–49
18.6 TOUCH SENSING HARDWARE 18–51
18.6.1 Hardware Overview 18–51
18.6.2 Touch Sensing Input Signal 18–51
18.6.3 Touch Sensing Enable/Disable Output Signal 18–51
18.6.4 Simple Low Voltage Touch Sense Detection Circuit 18–52
18.7 LASER ANALOG INPUT NON-CONTACT TOUCH SENSING 18–54
18.7.1 Overview 18–54
18.7.2 System Setup 18–54
18.7.3 Analog Input Touch I/O Setup 18–56
18.7.4 Analog Input Search Direction 18–57

Appendix A ERROR CODES AND RECOVERY A–1
A.1 OVERVIEW A–2
A.2 ERROR CODE PROPERTIES A–3
A.2.1 Overview A–3
A.2.2 Facility Name and Code A–8

A.2.3 Severity Descriptions A–12
A.2.4 Error Message Text A–15
A.3 GENERAL ERROR RECOVERY PROCEDURES A–17
A.3.1 Overview A–17
A.3.2 Overtravel Release A–17
A.3.3 Hand Breakage Recovery A–19
A.3.4 Pulse Coder Alarm Recovery A–20
A.3.5 Chain Failure Detection Error Recovery A–22

Appendix B CRT/KB SETUP AND OPERATIONS B–1
B.1 OVERVIEW B–2
B.2 CRT/KB SETUP B–3
B.2.1 Overview B–3
B.2.2 Set the Communications Parameters on the Controller Port B–3
B.2.3 Connect the Controller Port to the Remote CRT/KB B–3
B.2.4 Configure the Terminal Parameters of the Remote CRT/KB B–4
B.2.5 Program the Key Codes for the Function Keys of the Remote CRT/KB B–4
B.3 CRT/KB MENUS B–5
B.4 CRT/KB KEYS B–6

Appendix C SYSTEM OPERATIONS C–1
C.1 STARTUP METHODS C–2
C.1.1 Overview C–2
C.1.2 Hot Start C–3
C.1.3 Cold Start(START COLD) C–6
C.1.4 Cycle Power C–9
C.1.5 Controlled Start C–10
C.1.6 INIT Start C–15
C.2 HARDWARE DIAGNOSIS C–16
C.3 CONTROLLER BACKUP AND RESTORE C–22
C.3.1 Overview C–22
C.3.2 Backing Up a Controller as Images C–22
C.3.3 Restoring Controller Images C–25
C.4 MAINTENANCE OPERATIONS C–28
C.4.1 Overview C–28
C.4.2 Updating Boot Software from the Configuration Menu C–28
C.4.3 Updating *i*Pendant Firmware C–31
C.5 DIAGNOSTIC TOOLS C–42
C.5.1 Development Tools C–42
C.5.2 Diagnostic Log C–42

Appendix D MASTERING D–1
D.1 OVERVIEW D–2
D.2 RESETTING ALARMS AND PREPARING FOR MASTERING D–4
D.3 SAVING AND RESTORING MASTERING DATA D–6
D.4 MASTERING TO A FIXTURE (FIXTURE POSITION MASTER) D–8
D.5 SINGLE AXIS MASTERING D–11
D.6 QUICK MASTERING D–15
D.7 ZERO DEGREE MASTERING D–19

Appendix E PROGRAM EXAMPLES E–1
E.1 Program Examples Overview E–2
E.2 **POSITION REGISTER ELEMENT** E–2
E.3 **POSITION REGISTER VALUE** E–2
E.4 REGISTER ANALOG INPUT E–3
E.5 CONDITIONAL BRANCHING; USING LABELS E–4
E.6 CALL KAREL SOFTPART E–5
E.7 REGISTER INCREMENT E–5
E.8 GROUP OUTPUT; WAIT INSTRUCTION PULSE INSTRUCTION E–6
E.9 LABELS E–6
E.10 LABEL; JUMP LABEL; MESSAGE E–7
E.11 MACRO INSTRUCTION E–8

Appendix F ADVANCED *i*PENDANT FUNCTIONS F–1
F.1 Overview F–2
F.2 *i*Pendant Controls Installation F–2
F.3 Remote Monitoring F–3
F.3.1 Overview F–3
F.3.2 Setup F–3
F.3.3 Operation F–6
F.3.4 Limitations F–9
F.4 Remote Operation F–11
F.4.1 Overview F–11
F.4.2 Setup F–11
F.4.3 Remote *i*Pendant Operation F–12
F.4.4 Limitations F–19
F.5 Screen Customization F–20
F.5.1 Overview F–20
F.5.2 Easy Panel Environment Setup F–20
F.5.3 Panel Wizard F–21
F.5.4 *i*Pendant HMI Screen Customizations F–25
F.5.5 Displaying Web Page Macros F–33
F.5.6 Customizing User Menus F–34

Glossary GL–38

Index Index–1

List of Figures

Figure 1. HandlingTool Quick Reference Flow Chart lxviii
Figure 2. Teach Pendant/*i*Pendant Keys lxix
Figure 1–1. System Overview 1–3
Figure 1–2. Major and Minor Axes 1–4
Figure 1–3. Finger Type Gripper 1–5
Figure 1–4. Vacuum Type Gripper 1–6
Figure 1–5. Controller 1–7
Figure 1–6. Controller Capabilities 1–8
Figure 1–7. Controller Possible Configuration 1–9
Figure 1–8. Controller Standard Operator Panel 1–11
Figure 1–9. Mode Select Switch 1–12
Figure 1–10. Effect of Opening the Safety Fence While in AUTO Mode 1–15
Figure 1–11. CRT/KB 1–17
Figure 1–12. Controller Memory 1–24
Figure 1–13. Monochrome Teach Pendant 1–27
Figure 1–14. iPendant Overview 1–28
Figure 1–15. DEADMAN Switch 1–29
Figure 1–16. Teach Pendant Switches 1–30
Figure 1–17. EMERGENCY STOP Button 1–30
Figure 1–18. Window Display Control Menu 1–39
Figure 1–19. Multiple Window Display 1–40
Figure 1–20. Window and Focus Example 1–42
Figure 1–21. First Level and Fly Out Popup Menus 1–44
Figure 1–22. Status Bar Display 1–45
Figure 1–23. Touch Screen Navigation 1–47
Figure 1–24. User Views 1–51
Figure 1–25. Modifying User Views Screen 1–52
Figure 1–26. Maximize Window 1–54
Figure 1–27. Restore Window 1–55
Figure 1–28. Menu Favorites 1–56
Figure 1–29. Menu Favorites 1–57
Figure 1–30. Modifying the Menu Favorites Screen 1–58
Figure 1–31. Menu History Menu List 1–59

Figure 1–32. *i*Pendant with USB Port 1–65
Figure 2–1. Jog Speed Keys 2–7
Figure 2–2. COORD Display 2–8
Figure 2–3. COORD Display for the iPendant 2–8
Figure 3–1. World Frame 3–4
Figure 3–2. Moving a Frame 3–6
Figure 3–3. HandlingTool Tool Frame 3–7
Figure 3–4. Defining the Orientation of the Origin 3–20
Figure 3–5. World and User Frames 3–29
Figure 3–6. World and User Frames for ArcTool 3–30
Figure 3–7. Defining the Origin 3–35
Figure 3–8. Defining the X Direction Point 3–36
Figure 3–9. Defining the X-Y Plane 3–37
Figure 3–10. Defining the Origin 3–41
Figure 3–11. Defining the X Direction Point 3–42
Figure 3–12. Defining the X-Y Plane 3–42
Figure 3–13. Defining the Second Origin 3–43
Figure 3–14. Remote TCP Frame 3–51
Figure 3–15. Touching the TCP of the Robot Tool to the Remote TCP 3–54
Figure 3–16. Touching the TCP of the Robot Tool to the Remote TCP 3–57
Figure 3–17. Jog Frame Defined Parallel to Part 3–62
Figure 3–18. Defining the Origin 3–67
Figure 3–19. Defining the X Direction Point 3–67
Figure 3–20. Defining the X-Y Plane 3–68
Figure 3–21. Production SETUP Screen 3–77
Figure 3–22. RSR Timing Diagram 3–83
Figure 3–23. PNS Timing Diagram for DispenseTool (default and recommended configuration) 3–88
Figure 3–24. Typical Style Initiate Sequence 3–93
Figure 3–25. Fault Recovery During Style Execution 3–94
Figure 3–26. Password Configuration Screen 3–148
Figure 3–27. Error Severity Table 3–174
Figure 3–28. Example Output Definition 3–183
Figure 3–29. Timing - One Alarm 3–184
Figure 3–30. Timing - Multiple Alarms 3–184
Figure 3–31. Inertia Equations 3–194
Figure 3–32. Valid Payload Configuration for Accurate Estimation 3–196
Figure 3–33. J5 and J6 Orientation for POS1 3–205
Figure 3–34. Acceleration Equation 3–207
Figure 3–35. Path with Four Path Nodes 3–216
Figure 4–1. Rack, Slot, Channel, and Starting Point 4–7
Figure 4–2. Example Distributed I/O Setup Block Diagram 4–30

Figure 4–3. Interface Unit DIP Switches 4–31
Figure 4–4. Basic Digital I/O Module DIP Switches 4–33
Figure 4–5. I/O Link Device Screen 4–48
Figure 4–6. I/O Link Diagram 4–51
Figure 4–7. System that Uses FANUC I/O Link Connection Units 4–52
Figure 4–8. LED Locations 4–54
Figure 4–9. Outline Drawing 4–55
Figure 4–10. Mounting Location 4–56
Figure 4–11. Example Connection Diagram 4–57
Figure 4–12. Electrical Signal Cable Connectors 4–58
Figure 4–13. Power Supply Cable Connector 4–59
Figure 4–14. RSR Timing Diagram 4–79
Figure 4–15. PNS Timing Diagram 4–80
Figure 5–1. Continuous Termination Type for Movement Around Obstacles 5–5
Figure 5–2. Minimizing Wrist Orientation Changes 5–6
Figure 5–3. Home Position 5–11
Figure 5–4. Purge Position 5–13
Figure 5–5. Pounce Position 5–14
Figure 5–6. Safe Position 5–15
Figure 5–7. Writing and Modifying a Program 5–16
Figure 5–8. Background Edit Process 5–50
Figure 5–9. Background Edit Process (Continued) 5–51
Figure 5–10. Configuration/Detail Screens 5–61
Figure 5–11. Editor Related Views 5–66
Figure 5–12. Icon Editor 5–67
Figure 5–13. Instruction Icons 5–68
Figure 5–14. TOOLBAR Icons 5–69
Figure 5–15. INPUT Icon 5–70
Figure 5–16. Context Sensitive Icon Editing 5–71
Figure 6–1. Program Example 6–5
Figure 6–2. Program Example 6–6
Figure 6–3. Program Header Information 6–12
Figure 6–4. Typical Motion Instruction Example 6–23
Figure 6–5. Joint Motion Type 6–24
Figure 6–6. Linear Motion Type 6–25
Figure 6–7. Linear Motion Type Used to Rotate About the Tool Center Point 6–26
Figure 6–8. Circular Motion Type 6–27
Figure 6–9. Restart of Circular Motion Instruction 6–28
Figure 6–10. Restart of Circular Motion Instruction 6–28
Figure 6–11. Positional Information 6–29
Figure 6–12. An Example Program in Display Mode 1 6–31

Figure 6–13. Motion Info is Toggled ON and OFF with EDCMD 6–32
Figure 6–14. Resume Distance Display 6–35
Figure 6–15. Frame Number of Positional Data Example Program 6–42
Figure 6–16. Example of the Sec Speed Feature 6–43
Figure 6–17. Variable Motion Speed Program Execution Example 6–45
Figure 6–18. Syntax for Changing the Motion Speed 6–47
Figure 6–19. Robot Motion with Fine Termination Type 6–51
Figure 6–20. Robot Motion with Continuous Termination Type 6–52
Figure 6–21. Acceleration Override 6–56
Figure 6–22. Shortest Motion Within Axis Limit 6–57
Figure 6–23. SKIP LBL[x] Motion Option Example 6–59
Figure 6–24. Tool Offset Instruction 6–63
Figure 6–25. Position Representation Screen 6–64
Figure 6–26. TIME BEFORE / TIME AFTER Motion Option Instructions 6–67
Figure 6–27. Remote TCP Motion Option 6–69
Figure 6–28. Remote TCP (RTCP) Motion Option Example 6–70
Figure 6–29. 6–74
Figure 6–30. 6–79
Figure 6–31. 6–80
Figure 6–32. 6–81
Figure 6–33. 6–82
Figure 6–34. 6–83
Figure 6–35. 6–84
Figure 6–36. PICK and PLACE Application 6–85
Figure 6–37. PICK and PLACE Application with CNT100 6–85
Figure 6–38. Adjusting P3 and P5 with Linear Distance 6–86
Figure 6–39. RT_LD: Effect of CNT Value 6–88
Figure 6–40. Place Motion: Two Possible Traces 6–89
Figure 6–41. Corner Path 6–92
Figure 6–42. Half Segment Length 6–93
Figure 6–43. Program Speed Changes 6–94
Figure 6–44. Corner Path determined by CRy if Linear Dist is satisfied 6–95
Figure 6–45. Corner path is determined by Linear Distance 6–96
Figure 6–46. Travel Angle Definitions (Push and Drag) 6–103
Figure 6–47. Work Angle Definitions (XZ and XY) 6–105
Figure 6–48. Teaching a Reference Position 6–106
Figure 6–49. Torch Angle TA_REF Instruction 6–107
Figure 6–50. Torch Angle TA_REF PR[ ] Instruction 6–108
Figure 6–51. Web Page Main Display 6–109
Figure 6–52. Web Page Torch Angle Files 6–110
Figure 6–53. Web Page Torch Angle Report File 6–112

Figure 6–54. Torch Angle Report in a Spreadsheet Application 6–113
Figure 6–55. Travel Angle Progress Display 6–114
Figure 6–56. Work Angle Progress Display 6–115
Figure 6–57. Torch Angle System Variables 6–116
Figure 6–58. Torch Angle Graphical Display 6–117
Figure 6–59. Direct and Indirect Addressing Example 6–119
Figure 6–60. R[x] = [value] 6–119
Figure 6–61. R[x] = [value] [operator] [value] 6–121
Figure 6–62. PR[GRPn:x] = [value] 6–123
Figure 6–63. PR[GRPn:x] = [value] [operator] [value] 6–124
Figure 6–64. Position Register Element PR[i,j] 6–125
Figure 6–65. PR[i,j] = [value] 6–125
Figure 6–66. PR[i,j] = [value] [operator] [value] 6–127
Figure 6–67. String Register Assignment 6–129
Figure 6–68. String Register Concatenation 6–129
Figure 6–69. STRLEN Instruction 6–131
Figure 6–70. FINDSTR Instruction 6–132
Figure 6–71. SUBSTR Instruction 6–133
Figure 6–72. R[x] = DI[x] 6–134
Figure 6–73. DO[x] = ON/OFF 6–135
Figure 6–74. DO[x] = PULSE [,width] 6–135
Figure 6–75. DO[x] = R[x] 6–135
Figure 6–76. R[x] = RI[x] 6–136
Figure 6–77. RO[x] = ON/OFF 6–136
Figure 6–78. RO[x] = PULSE [,width] 6–137
Figure 6–79. RO[x] = R[x] 6–137
Figure 6–80. R[x] = AI[x] 6–138
Figure 6–81. AO[x] = value 6–138
Figure 6–82. R[x] = GI[x] 6–139
Figure 6–83. GO[x] = value 6–139
Figure 6–84. LBL[x] 6–140
Figure 6–85. JMP LBL[x] 6–140
Figure 6–86. CALL program 6–141
Figure 6–87. Program End Instruction 6–141
Figure 6–88. Register IF Instruction 6–142
Figure 6–89. I/O IF Instruction for DI/DO, RI/RO, SI/SO and UI/UO 6–143
Figure 6–90. I/O IF Instruction for PL 6–143
Figure 6–91. I/O IF Instruction for R, AI/AO, GI/GO and System Variable 6–143
Figure 6–92. Select Instruction 6–146
Figure 6–93. Wait Time 6–146
Figure 6–94. WAIT Condition for DI/DO, RI/RO, SI/SO, and UI/UO 6–147

Figure 6–95. WAIT Condition for DI/DO, RI/RO, SI/SO, UI/UO, and WI/WO 6–147
Figure 6–96. WAIT Condition for R, GI/GO, AI/AO, and Parameters 6–148
Figure 6–97. WAIT Condition 6–148
Figure 6–98. Mixed Logic WAIT Example 6–150
Figure 6–99. RSR Enable/Disable 6–151
Figure 6–100. User Alarm 6–152
Figure 6–101. Timer 6–152
Figure 6–102. OVERRIDE 6–153
Figure 6–103. Message Instruction 6–155
Figure 6–104. Parameter Name Write Instruction 6–156
Figure 6–105. Parameter Name Read Instruction 6–157
Figure 6–106. JOINT_MAX_SPEED Instruction - Multiple Motion Group Syntax 6–158
Figure 6–107. LINEAR_MAX_SPEED Instruction - Multiple Motion Group Syntax 6–158
Figure 6–108. JOINT_MAX_SPEED Instruction - Single Motion Group Syntax 6–158
Figure 6–109. LINEAR_MAX_SPEED Instruction - Single Motion Group Syntax 6–158
Figure 6–110. Skip Condition for DO/DI, RO/RI, SO/SI, and UO/UI 6–159
Figure 6–111. Skip Condition for DI/DO, RI/RO, SI/SO, UI/UO, and WI/WO 6–159
Figure 6–112. Skip Condition for R, GI/GO, AI/AO, and Parameters 6–160
Figure 6–113. Skip Condition 6–160
Figure 6–114. Offset Condition 6–164
Figure 6–115. UFRAME_NUM=[value] 6–164
Figure 6–116. UTOOL_NUM=[value] 6–165
Figure 6–117. UFRAME[i] = PR[x] 6–165
Figure 6–118. UTOOL[i] = PR[x] 6–166
Figure 6–119. Tool Offset Condition Instruction 6–167
Figure 6–120. RUN program 6–167
Figure 6–121. Macro Command Instruction 6–168
Figure 6–122. Parameter Example 6–169
Figure 6–123. CALL Program with Parameters 6–171
Figure 6–124. Parameters for Conditional Branching Example 6–172
Figure 6–125. MACRO Program with Parameters 6–172
Figure 6–126. Argument Registers 6–174
Figure 6–127. Use No More than Ten Parameters in an Instruction 6–175
Figure 6–128. Make Sure Data Types Match 6–175
Figure 6–129. Define All Parameter Elements 6–176
Figure 6–130. Use Parameters Defined in the Main Program 6–176
Figure 6–131. Cursor Position to Insert Parameters 6–178
Figure 6–132. PAUSE 6–183
Figure 6–133. ABORT 6–183
Figure 6–134. Error Program 6–184
Figure 6–135. RESUME_PROG = program 6–184

Figure 6–136. MAINT_PROG = program 6–185
Figure 6–137. CLEAR_RESUME_PRO 6–185
Figure 6–138. RETURN_PATH_DSBL 6–185
Figure 6–139. LOCK PREG Instruction 6–189
Figure 6–140. UNLOCK PREG Instruction 6–190
Figure 6–141. MONITOR Instruction 6–190
Figure 6–142. MONITOR END Instruction 6–191
Figure 6–143. Condition for Register, System Variable, and I/O Parameters 6–191
Figure 6–144. Condition2 for I/O 6–192
Figure 6–145. Condition2 for I/O 6–193
Figure 6–146. Condition for Error Status 6–193
Figure 6–147. Payload Instruction 6–194
Figure 6–148. Inertia Equations 6–196
Figure 6–149. SET ISDT SPEED A...=... 6–198
Figure 6–150. Example of PALLETIZING-B Instruction 6–203
Figure 6–151. Example of PALLETIZING-BX Instruction 6–209
Figure 6–152. Example of PALLETIZING-E Instruction 6–217
Figure 6–153. Example of PALLETIZING-EX Instruction 6–225
Figure 6–154. Palletizing Example 6–233
Figure 6–155. Direct and Indirect Addressing Example 6–236
Figure 6–156. PL[x] = [value] 6–236
Figure 6–157. PL[x] = [value] [operator] [value] 6–237
Figure 6–158. SEARCH START [i] PR[x] 6–238
Figure 6–159. SEARCH END 6–239
Figure 6–160. TOUCH OFFSET PR[x] 6–239
Figure 6–161. Backward Execution Example 6–240
Figure 6–162. TOUCH OFFSET END 6–240
Figure 6–163. SYNC_SCHED 6–262
Figure 6–164. INPOS 6–262
Figure 6–165. PR_STRT[n] 6–263
Figure 6–166. PR_END[n] 6–263
Figure 6–167. PR_SYNC[n] 6–263
Figure 7–1. Resume Tolerance Example 7–8
Figure 7–2. ALARM Recovery Without Specific Error and Action Text 7–12
Figure 7–3. Example Program Showing Backward Execution 7–23
Figure 8–1. Teach Pendant Status Indicators 8–3
Figure 8–2. iPendant Status Indicators 8–4
Figure 8–3. Standard Operator Panel 8–6
Figure 8–4. String Register Simple Import Text File 8–17
Figure 8–5. String Register Full Import Text File 8–17
Figure 8–6. Turn Number and Joint Placement Display on Position Screen 8–36

Figure 8–7. Turn Number Display Configuration 8–37
Figure 8–8. Joint Placement Configuration Examples for Fully Articulated Robots 8–37
Figure 8–9. Joint Placement Configuration Examples for Fully Articulated Paint Robots 8–38
Figure 8–10. Joint Placement Configuration Examples for Horizontally Articulated Robots 8–39
Figure 8–11. $SCR_GRP[group].$turn_axis[i] for Turn Number Display Configuration 8–40
Figure 8–12. Process Axes Status Screen 8–42
Figure 8–13. Status 1 Screen 8–46
Figure 8–14. Status 2 Screen 8–49
Figure 8–15. Pulse Screen 8–50
Figure 8–16. Torque Monitor Screen 8–51
Figure 8–17. Tracking Screen 8–52
Figure 8–18. Disturbance Torque Screen 8–53
Figure 8–19. Servo Diagnosis Main Screen 8–54
Figure 8–20. Servo Diagnosis Reducer Screen 8–56
Figure 8–21. Servo Diagnosis Over Heat Screen 8–57
Figure 8–22. Servo Diagnosis Torque Screen 8–58
Figure 8–23. Servo Diagnosis Disturbance Screen 8–58
Figure 8–24. Servo Diagnosis OVC Screen 8–59
Figure 8–25. Servo Diagnosis Collision Detection Screen 8–60
Figure 8–26. Simple Gripper Example 8–63
Figure 8–27. Select a Screen to Configure 8–66
Figure 8–28. Screen Settings Page 8–67
Figure 8–29. Selecting an Image File 8–70
Figure 8–30. File Devices 8–71
Figure 8–31. Adjust 8–72
Figure 8–32. Indicator Settings Page 8–73
Figure 8–33. Indicator Settings 8–75
Figure 8–34. Adjust the Location of LED Indicator State: 0 8–76
Figure 8–35. Custom Logic Program Field 8–77
Figure 8–36. SETUP Visual Diagnostics 8–79
Figure 8–37. Reposition the Screen 8–80
Figure 8–38. Select a Screen to View 8–82
Figure 8–39. Invalid Indicator 8–83
Figure 8–40. Visual Diagnostics Screen Display 8–84
Figure 9–1. Location of Ports on the Controller 9–13
Figure 9–2. Memory Card Interface Location 9–21
Figure 9–3. USB Memory Stick Device Location 9–22
Figure 9–4. New *i*Pendant USB Port Interface Location 9–22
Figure 9–5. Making Subdirectories 9–27
Figure 9–6. Displaying Subdirectories 9–28
Figure 9–7. Subdirectories 9–29

Figure 9–8. File Screen 9–42
Figure 9–9. Example ASCII File 9–96
Figure 10–1. User-definable Table 10–8
Figure 10–2. Change Speed Override Based on Torque (Analog Input) 10–15
Figure 10–3. Two Methods of Angle Entry Shift Function 10–19
Figure 10–4. Angle Entry Shift Screen Structure 10–20
Figure 10–5. Taught Configuration vs. Actual Configuration 10–26
Figure 10–6. Single Step Forward (FWD) vs. Backward (BWD) 10–28
Figure 10–7. Program Motion vs. Jogging: Behavior Might Be Different Near Singularity 10–29
Figure 10–8. Unforeseeable Singularity at a Corner Path 10–30
Figure 10–9. Process Axes Status Screen 10–54
Figure 10–10. BumpBox with coordinated Motion 10–69
Figure 10–11. BumpBox Schedule 10–70
Figure 10–12. BBox with Table 10–76
Figure 10–13. Coordinated motion setup screen 10–78
Figure 10–14. Leader Frame PopUp menu 10–79
Figure 10–15. Leader Frame Setup - 3 point Method 10–80
Figure 10–16. CD Leader Frame Example 10–81
Figure 10–17. Shape with a Perpendicular Axis 10–83
Figure 10–18. Orientation Setup Approach Positions 10–90
Figure 10–19. Condition Monitor Function 10–103
Figure 10–20. Condition for Register, System Variable, and I/O Parameters 10–107
Figure 10–21. Condition2 for I/O 10–108
Figure 10–22. Condition for Error status 10–108
Figure 10–23. Program Monitor Menu 10–109
Figure 10–24. System Monitor Menu 10–110
Figure 10–25. Coordinates Offset Screens 10–118
Figure 10–26. TCP Fixed Method 10–119
Figure 10–27. Robot Fixed Method - Example 1 10–120
Figure 10–28. Robot Fixed Method - Example 2 10–121
Figure 10–29. Total Cycle Time in One Cycle Mode 10–132
Figure 10–30. Ten Cycle Mode 10–133
Figure 10–31. Hundred Cycle Mode 10–134
Figure 10–32. One Hundred Hour Mode 10–135
Figure 10–33. Target Cycle Time 10–138
Figure 10–34. Reference Program Recorded Display 10–139
Figure 10–35. Process Limits 10–141
Figure 10–36. Sample Report 10–145
Figure 10–37. DVM Schedule List Screen 10–160
Figure 10–38. Log Viewer 10–164
Figure 10–39. Execution Timing of Distance Before 10–165

Figure 10–40. Normal Execution when $distbf_ver = 4 10–166
Figure 10–41. Distance Before Motion Option, DB 10–167
Figure 10–42. Checking Whether the TCP Goes into the Trigger Region 10–168
Figure 10–43. Size of Trigger Region 10–169
Figure 10–44. TCP Does not Go into Trigger Region 10–171
Figure 10–45. Penetration 10–172
Figure 10–46. DISTBF_TTS = 0 10–174
Figure 10–47. Execution Timing of an Action Program 10–174
Figure 10–48. Hold During the Execution of an Action Program 10–175
Figure 10–49. Signal Output Timing After Resume 10–175
Figure 10–50. DISTBF_TTS = 2 10–176
Figure 10–51. Resume After Jogging 10–177
Figure 10–52. Laser Height Sensor 10–184
Figure 10–53. Search Motion 10–185
Figure 10–54. Positions and Frames Defined After the Search 10–186
Figure 10–55. Shape Frames 10–187
Figure 10–56. Parallel Mirror Image with Mirror Plane in Center of Robot 10–192
Figure 10–57. Parallel Mirror Image with Mirror Plane Offset from Center of Robot 10–193
Figure 10–58. Parallel Mirror Image with Offset 10–193
Figure 10–59. Positional Mirror Image 10–194
Figure 10–60. Rotational Mirror Image 10–195
Figure 10–61. Rotational Mirror Image 10–195
Figure 10–62. Mirror Image Shift with Orientation Mirrored (Mirror Method) 10–196
Figure 10–63. Enhanced Mirror Image with Orientation Controlled (Fixture Method) 10–196
Figure 10–64. Mirror Image Key 10–197
Figure 10–65. Example of Robot Axes Only Mirror Image 10–197
Figure 10–66. Example of Extended Axes Integrated Mirror Image 10–198
Figure 10–67. Example of With Extended Axes Mirror Image 10–198
Figure 10–68. Mirroring an Entire Program 10–201
Figure 10–69. Mirroring a Portion of a Program 10–202
Figure 10–70. Error Recovery Setup Screen 10–214
Figure 10–71. Setting User Alarm Screen 10–219
Figure 10–72. RESUME_PROGRAM Instruction 10–221
Figure 10–73. CLEAR_RESUME_PROG Instruction 10–221
Figure 10–74. WELD.TP Example Program 10–221
Figure 10–75. WIRE_CUT.TP (Resume Program) Example Program 10–221
Figure 10–76. MAINT_PROGRAM Instruction 10–222
Figure 10–77. RETURN_PATH_DSBL Instruction 10–222
Figure 10–78. WELD.TP Example Program 10–223
Figure 10–79. Fast Exit/Entry Feature Enabled and Disabled in a Program 10–223
Figure 10–80. Normal Operation Auto Start Mode 10–228

Figure 10–81. Normal Operation without Execution of Resume Program 10–229
Figure 10–82. Resume Program Aborted 10–230
Figure 10–83. Normal Operation (Automatic Start DISABLED) 10–231
Figure 10–84. Auto Mode When an Undefined Alarm Occurs 10–232
Figure 10–85. Using Register Instructions to Synchronize Program Execution 10–234
Figure 10–86. Multi-Tasking Using the RUN Program Instruction 10–237
Figure 10–87. Single Step Execution Example 10–238
Figure 10–88. Single Step Backward Execution 10–238
Figure 10–89. Backward Execution of a RUN Instruction Example 10–239
Figure 10–90. Teach Pendant Menus Example 10–252
Figure 10–91. Prompt Box Message Menu 10–253
Figure 10–92. Prompt Box Message Menu Does not Exist Screen 10–254
Figure 10–93. Prompt Box Yes/No Menu 10–259
Figure 10–94. Prompt Box Yes/No Menu Does not Exist Screen 10–260
Figure 10–95. List Menu 10–265
Figure 10–96. List Menu Does not Exist Screen 10–266
Figure 10–97. Status Menu 10–273
Figure 10–98. Default Status Menu 10–274
Figure 10–99. Operator Entry Menu 10–285
Figure 10–100. Operator Entry Menu Does Not Exist Screen 10–286
Figure 10–101. Parallel Mirror Image with Mirror Plane in Center of Robot 10–297
Figure 10–102. Parallel Mirror Image with Mirror Plane Offset from Center of Robot 10–297
Figure 10–103. Parallel Mirror Image with Offset 10–298
Figure 10–104. Positional Mirror Image 10–299
Figure 10–105. Rotational Mirror Image 10–300
Figure 10–106. Rotational Mirror Image 10–300
Figure 10–107. Mirror Image Key 10–301
Figure 10–108. Example of Robot Axes Only Mirror Image 10–301
Figure 10–109. Example of Extended Axes Integrated Mirror Image 10–302
Figure 10–110. Example of With Extended Axes Mirror Image 10–302
Figure 10–111. Mirroring an Entire Program 10–305
Figure 10–112. Mirroring a Portion of a Program 10–305
Figure 10–113. Motion Start Delay Memory Buffers 10–319
Figure 10–114. Modified Program Listing Using Method 2 10–322
Figure 10–115. J5 After Configuration Change 10–323
Figure 10–116. Hold Without Original Path Resume — Path Resumes from Current Position 10–345
Figure 10–117. Hold With Original Path Resume — Path Resumes from Stop Position 10–346
Figure 10–118. Emergency Stop With Original Path Resume — Path Resumes from Stop Position 10–347
Figure 10–119. Payload Check Screen 10–349
Figure 10–120. Payload Setting Screen 10–350
Figure 10–121. Payload Setting Error 10–350

Figure 10–122. Modify Program Screen 10–351
Figure 10–123. Modify Program Screen 10–352
Figure 10–124. Power On/Off screen 10–352
Figure 10–125. Shifting an Entire Program 10–357
Figure 10–126. Shifting Portions of a Program 10–357
Figure 10–127. Parallel Shift 10–358
Figure 10–128. Parallel and Rotational Shift 10–359
Figure 10–129. Program Shift Key 10–359
Figure 10–130. Example of Robot Axes Only Shift 10–360
Figure 10–131. Example of Extended Axes Integrated Shift 10–360
Figure 10–132. Example of With Extended Axes Shift 10–361
Figure 10–133. Example of With Extended Axes Only Shift 10–361
Figure 10–134. Example of a Replace Extended Axes Shift 10–362
Figure 10–135. Turn Numbers 10–367
Figure 10–136. Hold With Constant Path Resume Offset — Path Resumes from Offset Stop Position 10–373
Figure 10–137. Hold with non-Constant Path Resume Offset — Path Resumes from Offset Stop Position 10–374
Figure 10–138. Resume Offset Setup Menu 10–376
Figure 10–139. Resume Offset Setup Menu Screen Items 10–377
Figure 10–140. Servo Tool 10–390
Figure 10–141. SETUP Circle DETAIL Screen 10–401
Figure 10–142. Circle Detail Terminology 10–402
Figure 10–143. SETUP Hexagon DETAIL Screen 10–403
Figure 10–144. Hexagon Detail Terminology 10–404
Figure 10–145. SETUP Rectangle DETAIL Screen 10–405
Figure 10–146. Rectangle Detail Terminology 10–406
Figure 10–147. SETUP Slot DETAIL Screen 10–407
Figure 10–148. Slot Detail Terminology 10–408
Figure 10–149. SETUP Keyhole DETAIL Screen 10–409
Figure 10–150. Keyhole Detail Terminology 10–410
Figure 10–151. SETUP Pommel DETAIL Screen 10–411
Figure 10–152. Pommel Detail Terminology 10–413
Figure 10–153. SETUP Cust Shape DETAIL Screen 10–413
Figure 10–154. UTILITIES Shape Gen DETAIL Screen Example 10–418
Figure 10–155. UTILITIES Shape Gen Screen 10–419
Figure 10–156. Shape Generated Program Example ACIR1 10–422
Figure 10–157. Shape Generated Program Example ACIR2 10–423
Figure 10–158. Production Program Example 10–424
Figure 10–159. Servo Schedules 10–425
Figure 10–160. Lemon Shape with a Bump on the Right Side 10–427
Figure 10–161. Lemon Shape with Outward Bump and Ripples 10–428

Figure 10–162. Flat Spot on Each Side 10–428
Figure 10–163. Two Flat Spots and an Inward Bump 10–429
Figure 10–164. Simulated Input Skip Enabled Screen 10–444
Figure 10–165. Soft Float in the X Direction 10–456
Figure 10–166. Soft Float[n] Independent Instruction Example 10–457
Figure 10–167. Soft Float[n] Motion Option Example 10–458
Figure 10–168. Improving Softness Example 10–461
Figure 10–169. Improving Softness 10–461
Figure 10–170. Layout and Tool Selection 10–463
Figure 10–171. Insertion Motion 10–464
Figure 10–172. Face-to-Face Motion 10–465
Figure 10–173. TCP Setup Example 10–467
Figure 10–174. Space Check - Single Robot Controller 10–471
Figure 10–175. Space Check - Single Robot Controller 10–472
Figure 10–176. Common Space Inside/Outside 10–476
Figure 10–177. Common Space Outside Boundary 10–477
Figure 10–178. Common Space Vertexes 10–478
Figure 10–179. MultiARM Controller 10–481
Figure 10–180. Max Value 10–482
Figure 10–181. Max Speed 10–483
Figure 10–182. Typical TP Shim Example 10–502
Figure 10–183. Advanced TP Shim Example Using USEMAST 10–503
Figure 10–184. Advanced TP Shim Example Using HISTORY 10–504
Figure 10–185. TIME BEFORE / TIME AFTER Motion Option Instructions 10–511
Figure 10–186. Normal Execution when $timebf_ver=4 10–512
Figure 10–187. Timing Sequence (TIME BEFORE instruction) 10–513
Figure 10–188. Timing Sequence (AFTER instruction) 10–514
Figure 10–189. Timing Sequence (TIME BEFORE instruction) 10–514
Figure 10–190. Main and Sub Program Examples 10–516
Figure 10–191. Program Example for TIME BEFORE Instruction 10–517
Figure 11–1. Example: Servo tool is the 2nd group and the number of tools is four. 11–4
Figure 11–2. 11–6
Figure 11–3. Calibration Motion Type 11–14
Figure 11–4. Tool Change Sequence 11–21
Figure 11–5. 11–24
Figure 11–6. 11–24
Figure 11–7. 11–25
Figure 11–8. 11–27
Figure 12–1. Composition of FANUC Sensor Data Packet 12–3
Figure 12–2. Handshaking When the Contents of a Register is Sent to the Sensor 12–6
Figure 12–3. Handshaking When a Register Number and Its Data is Received 12–6

Figure 12–4. Handshaking When Position Register Data is Received 12–7
Figure 12–5. Handshaking When Three Point Data is Received 12–7
Figure 12–6. Handshaking When the Transformation Matrix is Received 12–7
Figure 13–1. Currently Selected Application 13–6
Figure 13–2. HandlingTool User Keys 13–22
Figure 14–1. Define The Cell 14–6
Figure 14–2. Setup Machine List 14–7
Figure 14–3. Teach Cell 14–8
Figure 14–4. Run Production 14–9
Figure 14–5. CNC Tending Flowchart 14–10
Figure 14–6. Grid Example 14–14
Figure 14–7. Grid Example 14–16
Figure 14–8. Machine Tool Setup 14–26
Figure 15–1. Standard *i*RCalibration Mastering Calibration Plates 15–12
Figure 15–2. Example of First Three Positions 15–29
Figure 15–3. Example of Second Three Positions 15–29
Figure 15–4. Standard *i*RCalibration Mastering Calibration Plates 15–46
Figure 15–5. Example of First Three Positions 15–64
Figure 15–6. Example of Second Three Positions 15–64
Figure 15–7. Typical Arc Welding tool definition 15–79
Figure 15–8. Typical Waterjet Tool Definition 15–80
Figure 15–9. Recommended *i*RCalibration TCP Set Touch Plate and Pointer Specifications 15–81
Figure 15–10. Installing the Touch Plate 15–82
Figure 15–11. Example of First Three Positions 15–96
Figure 15–12. Example of Second Three Positions 15–97
Figure 15–13. Cal-Plate Calibration Positions 15–98
Figure 15–14. *i*RCalibration TCP Set NEW-XYZWPR Calibration Program Example 15–101
Figure 15–15. *i*RCalibration TCP Set Cal-Plate Calibration Program Example 15–102
Figure 15–16. Typical Arc Welding tool definition 15–110
Figure 15–17. Typical Waterjet Tool Definition 15–111
Figure 15–18. Recommended *i*RCalibration TCP Shift Touch Plate and Pointer Specifications 15–112
Figure 15–19. Installing the Touch Plate 15–113
Figure 15–20. *i*RCalibration TCP Shift XYZWPR Calibration Program Example 15–127
Figure 15–21. Example of Manually Created XYZ Search TPP Program 15–128
Figure 15–22. Example of Manually Created XY Search TP Program 15–128
Figure 15–23. Example of Manually Created Z Search Teach Pendant Program 15–129
Figure 15–24. Example of Manually Created XYZ Search TP Program Using Detect Circle Instruction (with XY and Z offsets) 15–129
Figure 15–25. Example of Manually Created XY Search TP Program Using Detect Circle Instruction (without XY and Z offsets) 15–130
Figure 15–26. Example of Standard iRCalibration TCP Shift-Generated Program with Update Frame Added 15–130

Figure 15–27. 6 Points Search (3-2-1) (Corresponds to) 15–154
Figure 15–28. 6 points (3-2-1) for x, y, z, w, p, r Offset 15–155
Figure 15–29. Example Program for Automatic Operation 15–156
Figure 15–30. Typical Arc Weld Tool Definition 15–166
Figure 15–31. Typical Waterjet Tool Definition 15–167
Figure 15–32. Non-uniform Surface (Example 1) 15–183
Figure 15–33. Non-uniform Surface (Example 2) 15–183
Figure 15–34. 6 Points Search (3-2-1) (Corresponds to) 15–184
Figure 15–35. 6 points (3-2-1) for x, y, z, w, p, r Offset 15–185
Figure 15–36. Example program for automatic operation. 15–186
Figure 16–1. Example of Collision Avoidance 16–4
Figure 16–2. Robots on Different Controllers with Interference Zone and I/O 16–6
Figure 16–3. Typical DualARM System with Ferris Wheel Positioner 16–7
Figure 16–4. Sphere Element 16–8
Figure 16–5. Cylinder Element 16–9
Figure 16–6. Plane Element 16–9
Figure 16–7. Teaching An Obstacle of a Sphere Type 16–11
Figure 16–8. Teaching An Obstacle of a Cylinder Type 16–11
Figure 16–9. Teaching An Obstacle of a Plane Type 16–12
Figure 16–10. Teaching an Obstacle of a Box Type 16–12
Figure 16–11. Element Data 16–17
Figure 16–12. Tool Model Example 16–19
Figure 16–13. Obstacle Model Example 16–27
Figure 16–14. Right Hand Rule Example 16–33
Figure 16–15. Infinite Plane and Finite Plane Examples 16–34
Figure 16–16. Plane Obstacle Example 16–35
Figure 16–17. Box Obstacle Example 16–36
Figure 16–18. Robots on Different Controllers with Interference Zone and I/O 16–48
Figure 16–19. Side view of the R-2000i/165F Robot and DH Frame Assignment and Dimensions 16–62
Figure 16–20. Isometric view of R-2000i/165F Robot and DH Frame Assignment 16–63
Figure 16–21. Side View of R-2000i/165F Robot with default robot model elements 16–64
Figure 16–22. Front view of R-2000i/165F robot with default robot model elements 16–65
Figure 17–1. Interference Avoidance Display 17–16
Figure 17–2. Interference Avoidance Display 2 17–17
Figure 17–3. Interference Avoidance Display 3 17–18
Figure 17–4. Interference Avoidance Display 4 17–19
Figure 17–5. Interference Avoidance Display 5 17–19
Figure 18–1. Example Program Including Touch Sensing Routine 18–4
Figure 18–2. Search Using Searches in One Direction 18–8
Figure 18–3. Search Using Offsets in Two Dimensions 18–9

Figure 18–4. Search Using Two Search Motions in Two Different Directions to Obtain X and Y Offset and Rotation about Z 18–10
Figure 18–5. Touch Frame Used in a Program 18–11
Figure 18–6. Simple Search Routine Using Searches in Two Directions 18–17
Figure 18–7. Fillet Search in One Direction (x) with Rotation about Z 18–18
Figure 18–8. Fillet Search in Two Directions (x and y) with Rotation about Z 18–19
Figure 18–9. Fillet Search in Three Directions (x, y, z) with Rotation about Z 18–19
Figure 18–10. V-Groove Search 18–20
Figure 18–11. OD/ID Search in Two Directions (X and Y) 18–20
Figure 18–12. Touch Sensing Motion Option Example 18–29
Figure 18–13. Points that Require Touching Up 18–31
Figure 18–14. Simple Search Example Program 18–32
Figure 18–15. One-Dimensional Search Ex. Prog. (Fillet/Lap, V-Groove) 18–33
Figure 18–16. Two Dimensional Search Example Program 18–33
Figure 18–17. Two Dimensional Search with Coordinated Motion Example Program (See and for illustrations) 18–34
Figure 18–18. First Illustration of Two Dimensional Search with Coordinated Motion Program Example 18–35
Figure 18–19. Second Illustration of Two Dimensional Search with Coordinated Motion Program Example 18–35
Figure 18–20. Simple Search with Coordinated Motion Example Program 18–36
Figure 18–21. Three Dimensional Search with Rotation Example Program (See for an illustration) 18–37
Figure 18–22. Illustration of Three Dimensional Search with Rotation Program Example 18–37
Figure 18–23. Part in Mastered Position and Offset Applied Illustration 18–39
Figure 18–24. Offset Value Illustration 18–41
Figure 18–25. Part with One Touch Sense Start Position, 2, and Three Points along a Path, 5, 6, 7 18–43
Figure 18–26. Illustration of the Path when an Offset is Applied 18–44
Figure 18–27. Offset Path Touch Up to Adjust Location of Points 6 and 7 18–44
Figure 18–28. New Master Touch Up Illustration 18–45
Figure 18–29. Incorrect Touch Up of a Path 18–45
Figure 18–30. Path Followed After Altering One Point 18–46
Figure 18–31. New Point Taught while Executing the Offset Path 18–47
Figure 18–32. Multiple Searches can be Performed for Complex Shapes 18–49
Figure 18–33. Illustration of Part Shape Change and the Effect on Multiple Searches Performed 18–49
Figure 18–34. Moving a Search Start Position along the Search Direction 18–50
Figure 18–35. Search Start Position moved to a New Location Off the Axis of the Search Direction 18–50
Figure 18–36. Simple Low Voltage Touch Sense Detection Circuit 18–53
Figure B–1. Remote CRT/KBs B–2
Figure B–2. CRT/KB Full Menus B–5
Figure F–1. Robot HOME page F–6
Figure F–2. Remote *i*Pendant LOGON Screen F–8
Figure F–3. Remote *i* Pendant Monitoring Screen F–9
Figure F–4. Highlighted Links F–10

Figure F–5. Remote *i* Pendant Operation Password Screen F–13
Figure F–6. Remote *i* Pendant LOGON Screen F–14
Figure F–7. Remote *i*Pendant Operation Screen F–15
Figure F–8. Function Keys and Auxiliary Keys F–16
Figure F–9. Main Panel Wizard Screen F–22
Figure F–10. Create Operation Panel Wizard Screen F–23
Figure F–11. Create Operation Panel Flowchart F–24
Figure F–12. *i*Pendant General Setup Screen F–26
Figure F–13. *i*Pendant HMI Setup Menu F–27
Figure F–14. *i*Pendant General Setup Screen F–29
Figure F–15. *i*Pendant HMI Setup Screen F–30
Figure F–16. *i*Pendant HMI FULL Menu F–31
Figure F–17. *i*Pendant HMI QUICK Menu F–32
Figure F–18. *i*Pendant HMI Setup Screen Disabling the DATA Key F–33

List of Tables

Table 1–1. Standard Operator Panel Buttons 1–11
Table 1–2. Robot Servo Status 1–16
Table 1–3. Navigation and Data Entry Keys 1–32
Table 1–4. Robot Motion Keys 1–33
Table 1–5. Execution Keys 1–34
Table 1–6. Editing Keys 1–35
Table 1–7. HandlingTool-Specific Keys 1–36
Table 1–8. Help and Diagnostic Key (*i*Pendant only) 1–37
Table 1–9. FCTN Menu 1–37
Table 1–10. 1–38
Table 1–11. Navigating Web Pages 1–43
Table 1–12. Status Bar Display Items 1–45
Table 1–13. Navigating Web Pages 1–48
Table 1–14. HELP/DIAGNOSTICS Menu Items 1–48
Table 1–15. Help and Diagnostics Operations 1–49
Table 1–16. Help/Diag Information Display Locations 1–49
Table 1–17. $UI_CONFIG.$recovermenu 1–61
Table 1–18. Favorites Screen Function Keys 1–62
Table 1–19. Favorites Screen Operations 1–62
Table 2–1. Utilities Hints Screen Items 2–4
Table 2–2. Jog Speed Values 2–6
Table 2–3. LCD Indicators 2–8
Table 2–4. Coordinate Systems 2–9
Table 2–5. SubGroup Example 2–13
Table 3–1. Tool Frame Setup Screen Items 3–7
Table 3–2. Tool Frame Setup Three Point DETAIL Screen Items 3–9
Table 3–3. Tool Frame Setup Six Point DETAIL Screen Items 3–9
Table 3–4. Tool Frame Setup Direct Entry DETAIL Screen Items 3–10
Table 3–5. User Frame Setup Screen Items 3–28
Table 3–6. User Frame Setup Three Point DETAIL Screen Items 3–30
Table 3–7. User Frame Setup Four Point DETAIL Screen Items 3–31
Table 3–8. User Frame Setup Direct Entry DETAIL Screen Items 3–32
Table 3–9. Jog Frame Setup Screen Items 3–62

Table 3–10. Jog Frame Setup Three Point DETAIL Screen Items 3–63
Table 3–11. Jog Frame Setup Direct Entry DETAIL Screen Items 3–63
Table 3–12. Production Setup Screen 3–77
Table 3–13. RSR Setup Item Description 3–84
Table 3–14. PNS Setup Item Description 3–90
Table 3–15. Style Name Setup Items 3–94
Table 3–16. Prog Select STYLE Setup DETAIL Screen Items 3–95
Table 3–17. Prog Select OTHER DETAIL Screen Items 3–98
Table 3–18. Macro Command Screen Items 3–99
Table 3–19. Application-Specific Teach Pendant User Keys 3–101
Table 3–20. Macro Command Assignments (ArcTool and HandlingTool) 3–111
Table 3–21. Macro Command Assignments (PaintTool) 3–111
Table 3–22. Macro Command Assignments (DispenseTool and SpotTool+) 3–112
Table 3–23. MANUAL Macros Screen Items 3–115
Table 3–24. SYSTEM Axis Limit Items 3–121
Table 3–25. SETUP General Screen Items 3–127
Table 3–26. Brake On Hold Settings 3–128
Table 3–27. $UALRM_SEV[n] Severity Values 3–132
Table 3–28. Setting/User Alarm Screen Items 3–132
Table 3–29. Override Select Menu Listing 3–135
Table 3–30. Password Levels 3–137
Table 3–31. SETUP Passwords Screen Items — Screen 3–144
Table 3–32. SETUP Passwords Screen Items — Screen 3–144
Table 3–33. Password Configuration Setup Items 3–149
Table 3–34. XML Command Syntax 3–150
Table 3–35. Password Log Screen Items 3–163
Table 3–36. Devices for Password Level Screen Permissions 3–165
Table 3–37. Password Level Screen Permissions 3–165
Table 3–38. Error Severity Table Items 3–175
Table 3–39. Error Code Output Signal Definition ($ER_OUT_PUT.$out_num=1) 3–180
Table 3–40. Error Code Severity Definition ($ER_OUT_PUT.$out_num = 1) 3–181
Table 3–41. Program Control: DO[25] and DO[26] 3–181
Table 3–42. Motion Control: DO[27] and DO[28] 3–182
Table 3–43. MOTION PERFORMANCE Screen Items 3–189
Table 3–44. MOTION/PAYLOAD SET Screen Items 3–192
Table 3–45. MOTION/PAYLOAD ID Items 3–196
Table 3–46. MOTION/ID POS1 ID Items 3–197
Table 3–47. MOTION/ARMLOAD SET Items 3–209
Table 3–48. Stroke Limit Setup Screen Items 3–210
Table 3–49. DATA KAREL Posns Screen Items 3–212
Table 3–50. DATA KAREL Vars Screen Items 3–224

Table 3–51. System Configuration Setup Screen Items 3–226
Table 3–52. Use PPABN Signal DETAIL Screen Items 3–235
Table 3–53. Default Logical Command DETAIL Screen Items 3–236
Table 3–54. Output when waiting on Input DETAIL Screen Items 3–236
Table 3–55. Hand Broken DETAIL Screen Items 3–237
Table 4–1. Process I/O Board Default Digital Input and Output Configuration 4–7
Table 4–2. I/O Hardware Eligible for Default UOP Assignment 4–8
Table 4–3. Default UOP Input Configuration for Process I/O Boards 4–8
Table 4–4. Default UOP Output Configuration for Process I/O Boards 4–8
Table 4–5. Rack Assignments for Different Kinds of I/O 4–9
Table 4–6. Slot Assignments for Different Kinds of I/O 4–9
Table 4–7. I/O Analog In/Out Monitor Screen Items 4–11
Table 4–8. I/O Analog In/Out Configure Screen Items 4–12
Table 4–9. I/O Digital In/Out Monitor Screen Items 4–13
Table 4–10. I/O Digital In/Out Configure Screen Items 4–14
Table 4–11. I/O Digital In/Out DETAIL Screen Items 4–16
Table 4–12. I/O Group In/Out Monitor Screen Items 4–16
Table 4–13. I/O Group In/Out Configure Screen Items 4–17
Table 4–14. CONFIG Screen I/O Status 4–21
Table 4–15. Range Setting Limitations 4–22
Table 4–16. Custom I/O Setup Items 4–28
Table 4–17. Communication Speed Settings for Switches Q and H 4–32
Table 4–18. Unit Number Settings of Switches 16, 8, 4, 2, and 1 4–33
Table 4–19. I/O Robot In/Out Screen Items 4–38
Table 4–20. I/O Robot In/Out DETAIL Screen Items 4–38
Table 4–21. I/O SOP In/Out Screen Items 4–41
Table 4–22. Standard Operator Panel Input Signals 4–42
Table 4–23. Standard Operator Panel Output Signals 4–43
Table 4–24. Process I/O Assignments 4–46
Table 4–25. Kinds of Process I/O available on HandlingTool 4–46
Table 4–26. Model A I/O Assignments 4–46
Table 4–27. Model B I/O Assignments 4–47
Table 4–28. I/O Link Device List Screen Items 4–47
Table 4–29. Device Names 4–49
Table 4–30. Devices that have Access to the DETAIL Screen 4–49
Table 4–31. FANUC I/O Link Specifications 4–53
Table 4–32. FANUC I/O Link Ordering Information 4–53
Table 4–33. LED Status Descriptions 4–54
Table 4–34. I/O InterConnect Screen Items 4–63
Table 4–35. Relationship Between the MODE SELECT Switch Signals and Modes of Operation 4–67
Table 4–36. I/O UOP In/Out Monitor Screen Items 4–72

Table 4–37. I/O UOP In/Out Configure Screen Items 4–73
Table 4–38. UOP UI to Process I/O Board DI 4–75
Table 4–39. UOP Input Signals 4–76
Table 4–40. UOP Outputs to Process I/O Board DO 4–81
Table 4–41. UOP Output Signals 4–82
Table 4–42. Cell Interface Input Signals 4–85
Table 4–43. Cell Interface Output Signals 4–86
Table 4–44. Check Items with Robot Ready 4–89
Table 5–1. Predefined Positions Used by Each Tool 5–9
Table 5–2. Rack 1 5–12
Table 5–3. Paste Methods 5–30
Table 5–4. Troubleshoot Background Edit - Problem Cause and Remedy 5–55
Table 5–5. Permanent Program Storage States 5–58
Table 5–6. FILE Programs Limitations and Attributes 5–59
Table 5–7. Temporary Program Storage States 5–59
Table 5–8. Load/Convert Times for 1,000 Point Program 5–59
Table 5–9. Storage Configuration Status and Settings 5–62
Table 5–10. Maximum Program Size for Typical CMOS Allocations 5–63
Table 6–1. Select Screen Items and Operations 6–7
Table 6–2. Program DETAIL Screen Items and Operations 6–8
Table 6–3. Program Instruction Summary for DispenseTool and SpotTool+ with Dispense Application Enabled 6–15
Table 6–4. Job and Process Program Element Summary for PaintTool 6–16
Table 6–5. $MNDSP_POSCF Configuration Settings 6–30
Table 6–6. Display Modes 6–33
Table 6–7. Display Mode Examples 6–36
Table 6–8. $FRM_CHKTYP Values 6–41
Table 6–9. Example Program Operation 6–42
Table 6–10. Range of Register Values to Specify a Variable Motion Speed 6–46
Table 6–11. Correspondence between $GROUP System Variables and the Teach Pendant Motion Instruction 6–118
Table 6–12. SR[x]=R[y] 6–130
Table 6–13. R[x]=SR[y] 6–130
Table 6–14. SR[x]=R[y]+SR[z] 6–130
Table 6–15. SR[x]=SR[y]+R[z] 6–131
Table 6–16. R[x]=STRLEN SR[y]: 6–131
Table 6–17. R[x]=FINDSTR SR[y],SR[z] 6–132
Table 6–18. SR[n]=SUBSTR SR[x],R[y],R[z] 6–133
Table 6–19. Parameter Instructions 6–170
Table 6–20. Parameter Data Types 6–171
Table 6–21. String Parameter System Variables 6–173

Table 6–22. Instructions That Can Use AR[] 6–174
Table 6–23. Instructions that can Use AR[] 6–182
Table 6–24. PALLETIZING-B Pallet Editor Items 6–204
Table 6–25. PALLETIZING-BX Pallet Editor Items 6–210
Table 6–26. PALLETIZINGE Pallet Editor Items 6–218
Table 6–27. PALLETIZING-EX Pallet Editor Items 6–225
Table 6–28. Data Types 6–241
Table 6–29. Arithmetical Operators 6–242
Table 6–30. Logical Operators 6–242
Table 6–31. Comparison Operators 6–242
Table 6–32. Priority of Operators 6–243
Table 6–33. Data Assignments 6–244
Table 6–34. Mixed Logic Error Messages 6–246
Table 6–35. Background Logic Execution Modes 6–248
Table 6–36. Background Logic Screen Items 6–250
Table 6–37. Background Logic Screen Operations 6–250
Table 6–38. TC Online Instruction 6–258
Table 6–39. Schedule and Robot Combinations 6–261
Table 7–1. Tolerance Setup Items 7–9
Table 7–2. Status Disabled Faults Screen Items 7–12
Table 7–3. Faults that can be Disabled 7–14
Table 7–4. SpotTool+ Faults that can be Disabled 7–14
Table 7–5. Test Cycle Conditions 7–17
Table 7–6. Test Cycle Spot Description 7–18
Table 7–7. Program Select and Program Start 7–36
Table 7–8. UTILITIES Prog Adj Screen Items 7–45
Table 7–9. UTILITIES Prog Adj DETAIL Screen Items 7–45
Table 8–1. Teach Pendant Status Indicators 8–4
Table 8–2. Standard Operator Panel Status Indicators 8–6
Table 8–3. DATA Registers Screen Items 8–8
Table 8–4. DATA Position Reg Screen Items 8–10
Table 8–5. Data String Reg Screen Items 8–14
Table 8–6. Safety Signals 8–21
Table 8–7. Program Timer Listing Screen Items 8–23
Table 8–8. Program Timer DETAIL Screen Items 8–23
Table 8–9. System Timer Screen Items 8–25
Table 8–10. Clock Screen Items 8–26
Table 8–11. STATUS Version ID SOFTWARE Screen Items 8–27
Table 8–12. STATUS Version ID CONFIG Screen Items 8–28
Table 8–13. STATUS Version ID MOTOR Screen Items 8–28
Table 8–14. STATUS Version ID SERVO Screen Items 8–29

Table 8–15. Memory Status 8–32
Table 8–16. POSITION Joint Screen Items 8–34
Table 8–17. POSITION User Screen Items 8–34
Table 8–18. POSITION World Screen Items 8–34
Table 8–19. Execution History Screen 8–40
Table 8–20. Process Axes Status Items 8–43
Table 8–21. Program Status or Production Status Items 8–43
Table 8–22. Status 1 Screen Items 8–46
Table 8–23. Servo Alarm Status 1; Address: FC80h (L-axis), FCC0h (M-axis) 8–46
Table 8–24. Alarm Terminology 8–46
Table 8–25. Description of Alarm Combinations 8–47
Table 8–26. Servo Alarm Status 2; Address: FC81h (L-axis), FCC1h (M-axis) 8–48
Table 8–27. Alarm Terminology 8–48
Table 8–28. Status 2 Screen Items 8–49
Table 8–29. Pulse Coder Alarm Status 8–49
Table 8–30. Alarm Terminology 8–49
Table 8–31. Pulse Screen Items 8–50
Table 8–32. Torque Monitor Items 8–51
Table 8–33. Tracking Screen Items 8–52
Table 8–34. Disturbance Torque Screen Items 8–53
Table 8–35. Servo Diagnosis Main Screen Items 8–54
Table 8–36. Diagnosis Reducer Screen Items 8–56
Table 8–37. Diagnosis Over Heat Screen Items 8–57
Table 8–38. Diagnosis Torque Screen Items 8–58
Table 8–39. Diagnosis Disturbance Screen Items 8–59
Table 8–40. Diagnosis OVC Screen Items 8–59
Table 8–41. Diagnosis Last Detection Screen Items 8–60
Table 8–42. Program Monitor Screen Items 8–61
Table 8–43. Indicator States 8–64
Table 8–44. SETUP Visual Diagnostics Screen Settings Setup Items 8–68
Table 8–45. SETUP Visual Diagnostics Indicator Setup Items 8–73
Table 9–1. Ports P1 - P4 9–14
Table 9–2. Default Communications Settings for Devices 9–14
Table 9–3. SETUP Port Init Screen Items 9–17
Table 9–4. File Output Using PRINT 9–40
Table 9–5. System variables to control filtering 9–41
Table 9–6. $PGINP_FLTR values 9–41
Table 9–7. Types of Files 9–43
Table 9–8. Robot Data File Groups 9–46
Table 9–9. Altering the FILE MENU [VIEW] Display 9–48
Table 9–10. System Files 9–52

Table 9–11. Application Files 9–53
Table 9–12. Error Log Files 9–54
Table 9–13. File Types Listings and Descriptions 9–57
Table 9–14. Back Up Operations 9–57
Table 9–15. File Types 9–63
Table 9–16. Auto Update Operations Items 9–65
Table 9–17. Valid SAVE Function Screens 9–72
Table 9–18. FILE Memory Screen Items 9–74
Table 9–19. Automatic Backup Setup Items 9–76
Table 9–20. Viewing ASCII Upload Error Screen Items 9–94
Table 9–21. Viewing ASCII Upload Error Screen Operations 9–94
Table 10–1. Input Parameters 10–9
Table 10–2. Output Parameters 10–11
Table 10–3. Map Functions 10–13
Table 10–4. Options 10–15
Table 10–5. Troubleshooting Solutions 10–37
Table 10–6. Automatic Collision Recovery Variable Data 10–42
Table 10–7. Basic Process Axes setup menu items 10–49
Table 10–8. Process Axes Status Items 10–54
Table 10–9. Bump Box Schedule Items 10–60
Table 10–10. Bump Box I/O Items 10–67
Table 10–11. Bump Frames 10–73
Table 10–12. Center Finder Menu Items 10–83
Table 10–13. Center Finder Motion SETUP Menu 10–85
Table 10–14. Troubleshooting Information 10–91
Table 10–15. Collision Guard Setup Items 10–93
Table 10–16. State of Condition Monitoring 10–106
Table 10–17. Program Monitor Menu Items 10–109
Table 10–18. System Monitor Menu Items 10–110
Table 10–19. Tool Offset Screen Items 10–122
Table 10–20. User Frame Offset Screen Items 10–126
Table 10–21. Tracked Cycle Time Categories 10–129
Table 10–22. Cycle Time Tracking Screen Items 10–130
Table 10–23. Data Monitor SETUP Screen Menu Items 10–142
Table 10–24. Data Monitor ITM DETAIL Screen Items 10–144
Table 10–25. Data Monitor Schedule Menu Items 10–148
Table 10–26. Setting Up DVM Time Based Schedules 10–155
Table 10–27. Event Based Schedules 10–157
Table 10–28. Log File Data 10–163
Table 10–29. Distance Before Specification 10–166
Table 10–30. Distance Before Signal Output Instructions 10–170

Table 10–31. $DB_CONDTYP 10–170
Table 10–32. Distance Before Alarms 10–172
Table 10–33. DISTBF_TTS for a CALL Program Action 10–173
Table 10–34. Distance Before System Variables 10–181
Table 10–35. Easy Normal Setup Items and Operations 10–187
Table 10–36. ENHANCED MIRROR IMAGE Screen Items 10–190
Table 10–37. Error Recovery Features 10–210
Table 10–38. Error Recovery Setup Items 10–215
Table 10–39. Auto Error Recovery Manual Function Screen Items 10–225
Table 10–40. Auto Error Recovery Manual Function Detail Screen Items 10–226
Table 10–41. Programming Considerations 10–242
Table 10–42. Group Mask Exchange Screen Items 10–245
Table 10–43. Group Mask Exchange Troubleshooting 10–246
Table 10–44. SETUP Menu Utility Screen Items 10–253
Table 10–45. Prompt box msg Menus Screen Items 10–255
Table 10–46. Prompt msg menus DETAIL Screen Items 10–255
Table 10–47. Prompt Box Yes/No Menu Screen Items 10–261
Table 10–48. Prompt Box Yes/No Menu DETAIL Screen Items 10–261
Table 10–49. Select from a list Screen Items 10–267
Table 10–50. List Menu DETAIL Screen Items 10–268
Table 10–51. Status Menu Variable Detail Information 10–275
Table 10–52. Status Menu Screen Items 10–278
Table 10–53. Status Menu DETAIL Screen Items 10–278
Table 10–54. Status Menu Sub-DETAIL Screen Items 10–278
Table 10–55. Operator Entry Menu Variable Detail Information 10–286
Table 10–56. Operator Entry Menu Screen Items 10–288
Table 10–57. Operator Entry Menu DETAIL Screen Items 10–288
Table 10–58. Operator Entry Menu Sub-DETAIL Screen Items 10–288
Table 10–59. MIRROR IMAGE SHIFT Screen Items 10–296
Table 10–60. Motion group DO Screen Items 10–315
Table 10–61. Offline Singularity Avoidance System Variables 10–322
Table 10–62. Singularity Avoid Screen Items 10–324
Table 10–63. Singularity Avoidance Func Screen Items 10–325
Table 10–64. Singularity Avoid DETAIL Screen Items 10–325
Table 10–65. Error Messages, Descriptions and Remedies 10–329
Table 10–66. Events Recorded by Log Book 10–334
Table 10–67. System Variables used for Filtering 10–341
Table 10–68. Alarm Filtering Example 10–343
Table 10–69. System Variables for Screen Filtering 10–344
Table 10–70. PROGRAM SHIFT Screen Items 10–362
Table 10–71. Reference Position LISTING Screen Items 10–369

Table 10–72. Reference Position DETAIL Screen Items 10–370
Table 10–73. Selecting Resume Offset Type 10–375
Table 10–74. Resume Offset Setup Menu Screen Items 10–376
Table 10–75. Send Event Macro Description 10–379
Table 10–76. Send Data Macro Description 10–380
Table 10–77. Send SysVar Macro Description 10–381
Table 10–78. Get Data Macro Description 10–383
Table 10–79. Request Menu Macro Description 10–384
Table 10–80. 10–391
Table 10–81. Shape Setup Information 10–395
Table 10–82. SETUP DETAIL Screen and UTILITIES DETAIL Screen Common Items 10–398
Table 10–83. Circle DETAIL Items (Circle-Specific Items) 10–401
Table 10–84. Hexagon DETAIL Items (Hexagon-Specific Items) 10–403
Table 10–85. Rectangle DETAIL Items (Rectangle-Specific Items) 10–405
Table 10–86. Slot DETAIL Items (Slot-Specific Items) 10–407
Table 10–87. Keyhole DETAIL Items (Keyhole-Specific Items) 10–409
Table 10–88. Pommel DETAIL Items (Pommel-Specific Items) 10–411
Table 10–89. Cust Shape DETAIL Screen Items 10–414
Table 10–90. Limitations for Custom Shapes 10–414
Table 10–91. UTILITIES Shape Gen DETAIL Screens Additional Items 10–417
Table 10–92. DATA Servo Screen Items 10–425
Table 10–93. Simple Teach Suite Main Setup Screen Items 10–431
Table 10–94. Auto Path Smoothing Menu Items 10–432
Table 10–95. Joint Soft Float Schedule Setup Items 10–449
Table 10–96. Cartesian Soft Float Setup Items 10–451
Table 10–97. Pushout Setup Items 10–453
Table 10–98. Softness Settings 10–464
Table 10–99. Softness Settings 10–465
Table 10–100. 10–466
Table 10–101. 10–467
Table 10–102. Common Space I/O 10–473
Table 10–103. Space Check Function Screen Items 10–474
Table 10–104. 10–477
Table 10–105. Teach Pendant Shim Screen Items and Function Keys 10–504
Table 10–106. HISTORY Screen Items 10–508
Table 11–1. 11–3
Table 11–2. Common setup 11–10
Table 11–3. Calibration Motion Type 11–12
Table 11–4. Calibration Speed and Acceleration Rate 11–16
Table 11–5. Position Register No. 11–16
Table 11–6. Touch Torque 11–16

Table 11–7. Detection Signal 11–16
Table 11–8. Status Tool Change Screen Items 11–22
Table 11–9. Servo Tool Change Initial Setup Screen Items 11–26
Table 11–10. Servo Tool Change Setup Screen Items 11–27
Table 12–1. FANUC Sensor Data Formats 12–4
Table 12–2. NULL Modem Cable Pin Connector Layout 12–9
Table 13–1. Handling Config Menu Options 13–5
Table 13–2. Material Handling Setup Screen Items 13–7
Table 13–3. Valve Signals 13–12
Table 13–4. Valve DETAIL Screen Items 13–13
Table 13–5. SETUP MH Tool Signal DETAIL Screen Items 13–15
Table 13–6. Handling Manual Operations 13–23
Table 13–7. Material Handling Valve Macro Programs 13–23
Table 13–8. Grip Part Macro 13–25
Table 13–9. Release Part Macro 13–28
Table 13–10. Part Present Macro 13–30
Table 13–11. Check No Part Macro 13–31
Table 13–12. Prepare to Pick Macro 13–31
Table 13–13. Clear to Proceed Macro 13–32
Table 13–14. Turn ON Vacuum Macro 13–33
Table 13–15. Turn OFF Vacuum Macro 13–36
Table 13–16. Turn OFF Blowoff Macro 13–38
Table 13–17. Set CurrentValve Macro 13–40
Table 13–18. Set MH Tool 13–40
Table 14–1. Relation of Machine Tool Options 14–2
Table 14–2. Process Navigator 14–3
Table 14–3. Define the Cell 14–4
Table 14–4. Teach Cell 14–5
Table 14–5. Run Production 14–5
Table 14–6. Teach Pendant Programs 14–11
Table 14–7. User Tools 14–18
Table 14–8. User Frames 14–19
Table 14–9. Macros 14–19
Table 14–10. Registers 14–20
Table 14–11. Position Registers 14–22
Table 14–12. Robot Input/Output 14–24
Table 14–13. User Alarms 14–24
Table 14–14. Process Navigator 14–28
Table 14–15. Setup Machine List 14–28
Table 14–16. Setup Gripper 14–28
Table 14–17. Setup Part Infeed Station 14–29

Table 14–18. Add a Pre Process Operation 14–29
Table 14–19. Setup Machine Station 14–29
Table 14–20. Add a Post Process Operation 14–30
Table 14–21. Setup Finished Part Station 14–30
Table 14–22. Setup User Frame 14–30
Table 15–1. Relation of *i*RCalibration Options 15–3
Table 15–2. Calibration Hardware 15–11
Table 15–3. *i*RCalibration Master Set Schedule Screen Description 15–13
Table 15–4. *i*RCalibration Master Set Detection Schedule Description 15–15
Table 15–5. Program Creation Screen Items for Standard *i*RCalibration Master Set Programs 15–23
Table 15–6. Robot Calibration Program Taught Points 15–27
Table 15–7. Troubleshooting 15–41
Table 15–8. Calibration Hardware 15–45
Table 15–9. *i*RCalibration Mastering Schedule Screen Description 15–47
Table 15–10. *i*RCalibration Mastering Detection Schedule Description 15–50
Table 15–11. Program Creation Screen Items for Standard *i*RCalibration Master Recovery Programs 15–58
Table 15–12. Robot Calibration Program Taught Points 15–62
Table 15–13. Troubleshooting 15–77
Table 15–14. Calibration Mode Descriptions 15–83
Table 15–15. *i*RCalibration TCP Set UTOOL Schedule Item Description 15–84
Table 15–16. *i*RCalibration TCP Set Detection Schedule Description 15–85
Table 15–17. New TCP Program Points 15–94
Table 15–18. *i*RCalibration TCP Set Fit Errors 15–107
Table 15–19. Calibration Mode Descriptions 15–114
Table 15–20. *i*RCalibration TCP Shift UTOOL Schedule Item Description 15–114
Table 15–21. *i*RCalibration TCP Shift Detection Schedule Description 15–116
Table 15–22. New TCP Program Points 15–124
Table 15–23. *i*RCalibration TCP Shift Fit Errors 15–137
Table 15–24. Frame Schedule Detail Schedule Setup 15–141
Table 15–25. *i*RCalibration Frame Set Detection Schedule Description 15–143
Table 15–26. Troubleshooting *i*RCalibration Frame Set 15–161
Table 15–27. Frame Schedule Detail Schedule Setup 15–168
Table 15–28. *i*RCalibration Frame Shift Detection Schedule Description 15–170
Table 15–29. Troubleshooting *i*RCalibration Frame Shift 15–194
Table 16–1. Interference Check Element Data Screen Items 16–10
Table 16–2. Interference Check Element Data Screen Items 16–14
Table 16–3. Interference Check Model Setup Screen Items 16–14
Table 16–4. Interference Check Robot Model Setup Screen Items 16–14
Table 16–5. Element Parameters 16–20
Table 16–6. Interference Check Element Data Screen Items 16–20
Table 16–7. Interference Check Tool Model Setup Screen Items 16–20

Table 16–8. Interference Check Tool Model List Screen Items 16–21
Table 16–9. Interference Check Tool Model Element List Screen Items 16–21
Table 16–10. Obstacle Model Parameters 16–27
Table 16–11. Interference Check Obstacle Model Setup Screen Items 16–28
Table 16–12. Interference Check Obstacle Model List Screen Items 16–28
Table 16–13. Interference Check Obstacle Model Data Screen Items 16–28
Table 16–14. Interference Check Obstacle Element Data Screen Items 16–28
Table 16–15. Setting Element Data 16–32
Table 16–16. Interference Check Combination Setup Screen Items 16–37
Table 16–17. Interference Combination to Check Screen Items 16–37
Table 16–18. Interference Combination to Check DETAIL Screen Items 16–37
Table 16–19. Interference Check Robot No. and Priority Table Screen Items 16–40
Table 16–20. Interference Check Remote Robot Setup Screen Items 16–40
Table 16–21. Interference Check Multiple Controller Setup Screen Items 16–48
Table 16–22. Interference Check Multiple Controller Setup Screen Items 16–49
Table 16–23. Interference Check Zone Setup Screen Items 16–54
Table 16–24. Interference Check Multi-Robot Zone Setup Screen Items 16–57
Table 16–25. Interference Check Status Screen Items 16–59
Table 16–26. R-2000I/165F Parameters 16–65
Table 17–1. Interference Check Setup Items 17–3
Table 17–2. Interference Check Model Setup Items 17–3
Table 17–3. AutoZone Space Setup Items 17–6
Table 17–4. AutoZone Space DETAIL Setup Items 17–6
Table 17–5. AutoZone Schedule Setup Items 17–11
Table 17–6. AutoZone Test-Run Setup Items 17–13
Table 18–1. Touch Frame Setup Items 18–12
Table 18–2. Search Pattern and Valid Pattern Type 18–21
Table 18–3. Touch Sensing Schedule Setup Items 18–22
Table 18–4. Touch I/O Setup Screen Items 18–57
Table A–1. Startup Methods A–3
Table A–2. Alarm Log Screen A–4
Table A–3. Application Alarm Screen A–4
Table A–4. Comm Log Screen A–4
Table A–5. Fault Recovery Screen A–4
Table A–6. Motion Alarm Screen A–5
Table A–7. System Alarm Screen A–5
Table A–8. Error Facility Codes A–8
Table A–9. Severity Descriptions A–12
Table A–10. Effects of Error Severity A–14
Table A–11. Hexadecimal Error Message Display A–16
Table A–12. Hexadecimal Notation and Axis in Error Examples A–16

Table A–13. MANUAL OT Release Items A–17
Table B–1. Port Settings for the Built-In CRT/KB and the FANUC Factory Terminal B–3
Table B–2. Key Codes and Function Key Mapping B–4
Table B–3. Correspondenc e Between Teach Pendant and CRT/KB Keys B–6
Table C–1. Startup Methods C–2
Table C–2. Controlled Start Options for All Software Applications C–10
Table C–3. Controlled Start Options for FlexTool C–12
Table C–4. Controlled Start Options for HandlingTool C–13
Table C–5. Handling Config Menu Options C–13
Table C–6. Hardware Diagnostic Functions C–16
Table C–7. Backup and Restore Controller Items C–22
Table D–1. SYSTEM Master/Cal Items D–3
Table F–1. Remote Operation Key Mapping F–16
Table F–2. Remote Operation Mouse Events F–17
Table F–3. System Variable Settings F–36

Safety

FANUC Robotics is not and does not represent itself as an expert in safety systems, safety equipment, or the specific safety aspects of your company and/or its work force. It is the responsibility of the owner, employer, or user to take all necessary steps to guarantee the safety of all personnel in the workplace.

The appropriate level of safety for your application and installation can best be determined by safety system professionals. FANUC Robotics therefore, recommends that each customer consult with such professionals in order to provide a workplace that allows for the safe application, use, and operation of FANUC Robotic systems.

According to the industry standard ANSI/RIA R15-06, the owner or user is advised to consult the standards to ensure compliance with its requests for Robotics System design, usability, operation, maintenance, and service. Additionally, as the owner, employer, or user of a robotic system, it is your responsibility to arrange for the training of the operator of a robot system to recognize and respond to known hazards associated with your robotic system and to be aware of the recommended operating procedures for your particular application and robot installation.

Ensure that the robot being used is appropriate for the application. Robots used in classified (hazardous) locations must be certified for this use.

FANUC Robotics therefore, recommends that all personnel who intend to operate, program, repair, or otherwise use the robotics system be trained in an approved FANUC Robotics training course and become familiar with the proper operation of the system. Persons responsible for programming the system-including the design, implementation, and debugging of application programs-must be familiar with the recommended programming procedures for your application and robot installation.

The following guidelines are provided to emphasize the importance of safety in the workplace.

CONSIDERING SAFETY FOR YOUR ROBOT INSTALLATION

Safety is essential whenever robots are used. Keep in mind the following factors with regard to safety:

- The safety of people and equipment
- Use of safety enhancing devices
- Techniques for safe teaching and manual operation of the robot(s)
- Techniques for safe automatic operation of the robot(s)
- Regular scheduled inspection of the robot and workcell
- Proper maintenance of the robot

Keeping People and Equipment Safe

The safety of people is always of primary importance in any situation. However, equipment must be kept safe, too. When prioritizing how to apply safety to your robotic system, consider the following:

- People
- External devices
- Robot(s)
- Tooling
- Workpiece

Using Safety Enhancing Devices

Always give appropriate attention to the work area that surrounds the robot. The safety of the work area can be enhanced by the installation of some or all of the following devices:

- Safety fences, barriers, or chains
- Light curtains
- Interlocks
- Pressure mats
- Floor markings
- Warning lights
- Mechanical stops
- EMERGENCY STOP buttons
- DEADMAN switches

Setting Up a Safe Workcell

A safe workcell is essential to protect people and equipment. Observe the following guidelines to ensure that the workcell is set up safely. These suggestions are intended to supplement and **not** replace existing federal, state, and local laws, regulations, and guidelines that pertain to safety.

- Sponsor your personnel for training in approved FANUC Robotics training course(s) related to your application. Never permit untrained personnel to operate the robots.
- Install a lockout device that uses an access code to prevent unauthorized persons from operating the robot.
- Use anti-tie-down logic to prevent the operator from bypassing safety measures.

- Arrange the workcell so the operator faces the workcell and can see what is going on inside the cell.
- Clearly identify the work envelope of each robot in the system with floor markings, signs, and special barriers. The work envelope is the area defined by the maximum motion range of the robot, including any tooling attached to the wrist flange that extend this range.
- Position all controllers outside the robot work envelope.
- Never rely on software or firmware based controllers as the primary safety element unless they comply with applicable current robot safety standards.
- Mount an adequate number of EMERGENCY STOP buttons or switches within easy reach of the operator and at critical points inside and around the outside of the workcell.
- Install flashing lights and/or audible warning devices that activate whenever the robot is operating, that is, whenever power is applied to the servo drive system. Audible warning devices shall exceed the ambient noise level at the end-use application.
- Wherever possible, install safety fences to protect against unauthorized entry by personnel into the work envelope.
- Install special guarding that prevents the operator from reaching into restricted areas of the work envelope.
- Use interlocks.
- Use presence or proximity sensing devices such as light curtains, mats, and capacitance and vision systems to enhance safety.
- Periodically check the safety joints or safety clutches that can be optionally installed between the robot wrist flange and tooling. If the tooling strikes an object, these devices dislodge, remove power from the system, and help to minimize damage to the tooling and robot.
- Make sure all external devices are properly filtered, grounded, shielded, and suppressed to prevent hazardous motion due to the effects of electro-magnetic interference (EMI), radio frequency interference (RFI), and electro-static discharge (ESD).
- Make provisions for power lockout/tagout at the controller.
- Eliminate *pinch points* . Pinch points are areas where personnel could get trapped between a moving robot and other equipment.
- Provide enough room inside the workcell to permit personnel to teach the robot and perform maintenance safely.
- Program the robot to load and unload material safely.
- If high voltage electrostatics are present, be sure to provide appropriate interlocks, warning, and beacons.
- If materials are being applied at dangerously high pressure, provide electrical interlocks for lockout of material flow and pressure.

Staying Safe While Teaching or Manually Operating the Robot

Advise all personnel who must teach the robot or otherwise manually operate the robot to observe the following rules:

- Never wear watches, rings, neckties, scarves, or loose clothing that could get caught in moving machinery.
- Know whether or not you are using an intrinsically safe teach pendant if you are working in a hazardous environment.
- Before teaching, visually inspect the robot and *work envelope* to make sure that no potentially hazardous conditions exist. The work envelope is the area defined by the maximum motion range of the robot. These include tooling attached to the wrist flange that extends this range.
- The area near the robot must be clean and free of oil, water, or debris. Immediately report unsafe working conditions to the supervisor or safety department.
- FANUC Robotics recommends that no one enter the work envelope of a robot that is on, except for robot teaching operations. However, if you must enter the work envelope, be sure all safeguards are in place, check the teach pendant DEADMAN switch for proper operation, and place the robot in teach mode. Take the teach pendant with you, turn it on, and be prepared to release the DEADMAN switch. Only the person with the teach pendant should be in the work envelope.

> **Warning**
>
> **Never bypass, strap, or otherwise deactivate a safety device, such as a limit switch, for any operational convenience. Deactivating a safety device is known to have resulted in serious injury and death.**

- Know the path that can be used to escape from a moving robot; make sure the escape path is never blocked.
- Isolate the robot from all remote control signals that can cause motion while data is being taught.
- Test any program being run for the first time in the following manner:

> **Warning**
>
> **Stay outside the robot work envelope whenever a program is being run. Failure to do so can result in injury.**

  — Using a low motion speed, single step the program for at least one full cycle.

  — Using a low motion speed, test run the program continuously for at least one full cycle.

  — Using the programmed speed, test run the program continuously for at least one full cycle.

- Make sure all personnel are outside the work envelope before running production.

Staying Safe During Automatic Operation

Advise all personnel who operate the robot during production to observe the following rules:

- Make sure all safety provisions are present and active.
- Know the entire workcell area. The workcell includes the robot and its work envelope, plus the area occupied by all external devices and other equipment with which the robot interacts.
- Understand the complete task the robot is programmed to perform before initiating automatic operation.
- Make sure all personnel are outside the work envelope before operating the robot.
- Never enter or allow others to enter the work envelope during automatic operation of the robot.
- Know the location and status of all switches, sensors, and control signals that could cause the robot to move.
- Know where the EMERGENCY STOP buttons are located on both the robot control and external control devices. Be prepared to press these buttons in an emergency.
- Never assume that a program is complete if the robot is not moving. The robot could be waiting for an input signal that will permit it to continue activity.
- If the robot is running in a pattern, do not assume it will continue to run in the same pattern.
- Never try to stop the robot, or break its motion, with your body. The only way to stop robot motion immediately is to press an EMERGENCY STOP button located on the controller panel, teach pendant, or emergency stop stations around the workcell.

Staying Safe During Inspection

When inspecting the robot, be sure to

- Turn off power at the controller.
- Lock out and tag out the power source at the controller according to the policies of your plant.
- Turn off the compressed air source and relieve the air pressure.
- If robot motion is not needed for inspecting the electrical circuits, press the EMERGENCY STOP button on the operator panel.
- Never wear watches, rings, neckties, scarves, or loose clothing that could get caught in moving machinery.
- If power is needed to check the robot motion or electrical circuits, be prepared to press the EMERGENCY STOP button, in an emergency.
- Be aware that when you remove a servomotor or brake, the associated robot arm will fall if it is not supported or resting on a hard stop. Support the arm on a solid support before you release the brake.

Staying Safe During Maintenance

When performing maintenance on your robot system, observe the following rules:

- Never enter the work envelope while the robot or a program is in operation.
- Before entering the work envelope, visually inspect the workcell to make sure no potentially hazardous conditions exist.
- Never wear watches, rings, neckties, scarves, or loose clothing that could get caught in moving machinery.
- Consider all or any overlapping work envelopes of adjoining robots when standing in a work envelope.
- Test the teach pendant for proper operation before entering the work envelope.
- If it is necessary for you to enter the robot work envelope while power is turned on, you must be sure that you are in control of the robot. Be sure to take the teach pendant with you, press the DEADMAN switch, and turn the teach pendant on. Be prepared to release the DEADMAN switch to turn off servo power to the robot immediately.
- Whenever possible, perform maintenance with the power turned off. Before you open the controller front panel or enter the work envelope, turn off and lock out the 3-phase power source at the controller.
- Be aware that an applicator bell cup can continue to spin at a very high speed even if the robot is idle. Use protective gloves or disable bearing air and turbine air before servicing these items.
- Be aware that when you remove a servomotor or brake, the associated robot arm will fall if it is not supported or resting on a hard stop. Support the arm on a solid support before you release the brake.



Warning

Lethal voltage is present in the controller WHENEVER IT IS CONNECTED to a power source. Be extremely careful to avoid electrical shock. HIGH VOLTAGE IS PRESENT at the input side whenever the controller is connected to a power source. Turning the disconnect or circuit breaker to the OFF position removes power from the output side of the device only.

- Release or block all stored energy. Before working on the pneumatic system, shut off the system air supply and purge the air lines.
- Isolate the robot from all remote control signals. If maintenance must be done when the power is on, make sure the person inside the work envelope has sole control of the robot. The teach pendant must be held by this person.

- Make sure personnel cannot get trapped between the moving robot and other equipment. Know the path that can be used to escape from a moving robot. Make sure the escape route is never blocked.
- Use blocks, mechanical stops, and pins to prevent hazardous movement by the robot. Make sure that such devices do not create pinch points that could trap personnel.



Warning

Do not try to remove any mechanical component from the robot before thoroughly reading and understanding the procedures in the appropriate manual. Doing so can result in serious personal injury and component destruction.

- Be aware that when you remove a servomotor or brake, the associated robot arm will fall if it is not supported or resting on a hard stop. Support the arm on a solid support before you release the brake.
- When replacing or installing components, make sure dirt and debris do not enter the system.
- Use only specified parts for replacement. To avoid fires and damage to parts in the controller, never use nonspecified fuses.
- Before restarting a robot, make sure no one is inside the work envelope; be sure that the robot and all external devices are operating normally.

KEEPING MACHINE TOOLS AND EXTERNAL DEVICES SAFE

Certain programming and mechanical measures are useful in keeping the machine tools and other external devices safe. Some of these measures are outlined below. Make sure you know all associated measures for safe use of such devices.

Programming Safety Precautions

Implement the following programming safety measures to prevent damage to machine tools and other external devices.

- Back-check limit switches in the workcell to make sure they do not fail.
- Implement “failure routines” in programs that will provide appropriate robot actions if an external device or another robot in the workcell fails.
- Use *handshaking* protocol to synchronize robot and external device operations.
- Program the robot to check the condition of all external devices during an operating cycle.

Mechanical Safety Precautions

Implement the following mechanical safety measures to prevent damage to machine tools and other external devices.

- Make sure the workcell is clean and free of oil, water, and debris.
- Use software limits, limit switches, and mechanical hardstops to prevent undesired movement of the robot into the work area of machine tools and external devices.

KEEPING THE ROBOT SAFE

Observe the following operating and programming guidelines to prevent damage to the robot.

Operating Safety Precautions

The following measures are designed to prevent damage to the robot during operation.

- Use a low override speed to increase your control over the robot when jogging the robot.
- Visualize the movement the robot will make before you press the jog keys on the teach pendant.
- Make sure the work envelope is clean and free of oil, water, or debris.
- Use circuit breakers to guard against electrical overload.

Programming Safety Precautions

The following safety measures are designed to prevent damage to the robot during programming:

- Establish *interference zones* to prevent collisions when two or more robots share a work area.
- Make sure that the program ends with the robot near or at the home position.
- Be aware of signals or other operations that could trigger operation of tooling resulting in personal injury or equipment damage.
- In dispensing applications, be aware of all safety guidelines with respect to the dispensing materials.

Note Any deviation from the methods and safety practices described in this manual must conform to the approved standards of your company. If you have questions, see your supervisor.

QUICK REFERENCE

Quick Reference Overview

The flowcharts in this section contain steps that will help you set up and use your system. These flowcharts do not include all the features and advanced functions that your application tool offers, however, they provide a basic framework for successfully implementing your particular program.

Use this section as a quick reference guide to help you locate information in the manual. Refer to the appropriate section for more detailed information.

Conditions

Be sure that the following conditions are met before using the flowcharts:

- The robot is installed and all surrounding equipment is placed properly. Refer to the *Mechanical Connection and Maintenance Manual,* specific for your robot, for more information.
- All interfaces are connected properly, including emergency stop circuits and hand breaks. Refer to the *Electrical Connection and Maintenance Manual* for more information.
- If you are using PalletTool, the gripper is installed and wired properly. Refer to the *Electrical Connection and Maintenance Manual* for more information.
- The application software is loaded and configured for the correct hardware. Refer to the *Software Installation Manual* for more information.
- The robot is mastered properly. Refer to the *Mechanical Unit Connection and Maintenance Manual* for more information.

HandlingTool Quick Reference Flow Chart

Figure 1. HandlingTool Quick Reference Flow Chart

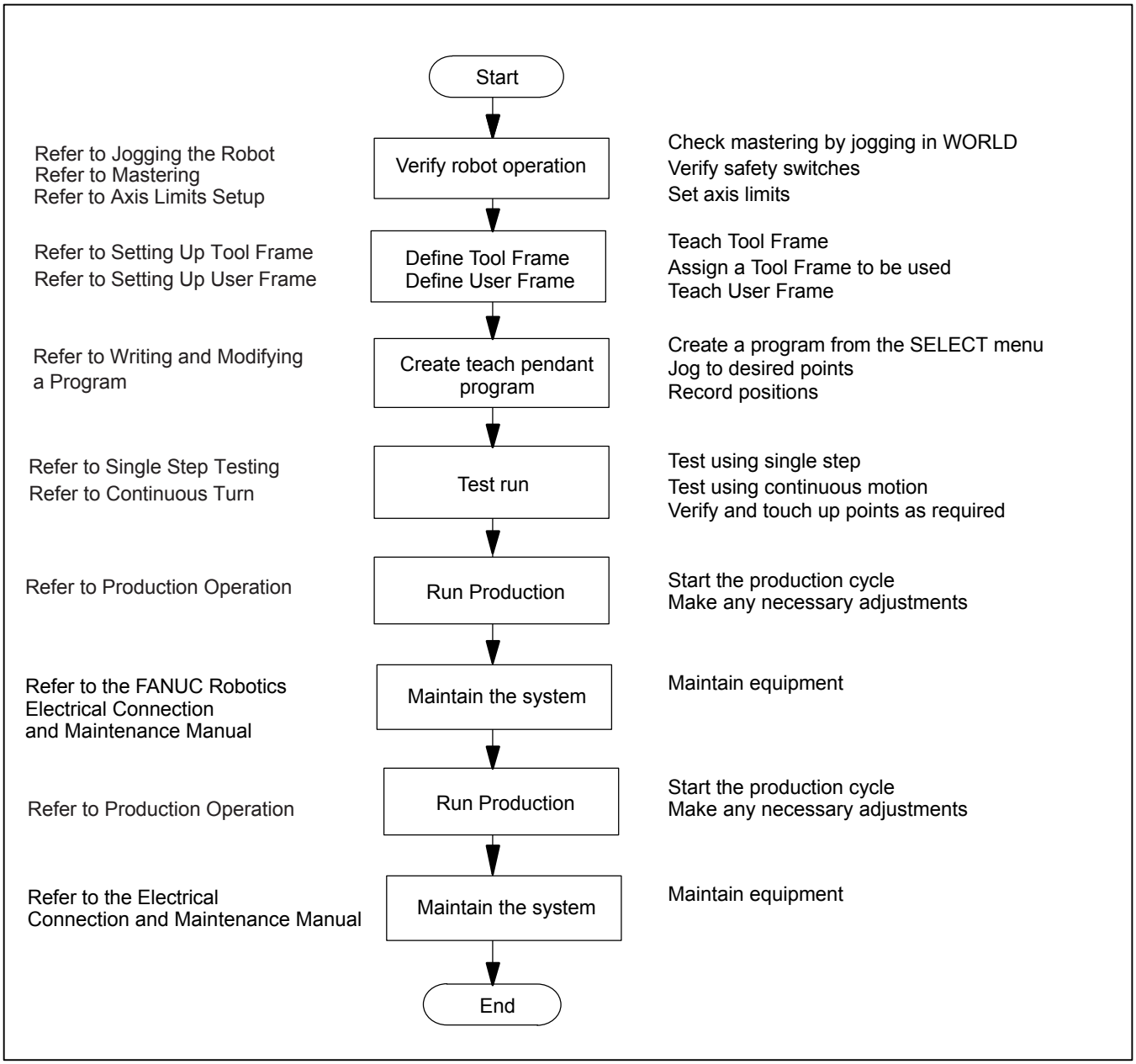


Figure 2. Teach Pendant/*i*Pendant Keys



Chapter 1

OVERVIEW

Contents

Chapter 1 OVERVIEW 1–1
1.1 OVERVIEW 1–3
1.2 ROBOT 1–4
1.2.1 Robot Overview 1–4
1.2.2 Robot Models 1–5
1.2.3 HandlingTool End-of-Arm Tooling (EOAT) 1–5
1.2.4 Extended Axes 1–6
1.3 CONTROLLER 1–7
1.3.1 Controller Overview 1–7
1.3.2 Teach Pendant 1–10
1.3.3 Standard Operator Panel 1–10
1.3.4 MODE SELECT Switch 1–12
1.3.5 Robot Stop Variation 1–16
1.3.6 User Operator Panel (UOP) 1–17
1.3.7 CRT/KB 1–17
1.3.8 Emergency Stop Devices 1–17
1.3.9 Communications 1–17
1.3.10 Input/Output (I/O) 1–18
1.3.11 Remote I/O Interfaces 1–19
1.3.12 Motion 1–20
1.3.13 Extended Axes 1–21
1.3.14 Multiple Applications (option) 1–22
1.3.15 Controller Backplane 1–22
1.3.16 Memory 1–23
1.3.17 Line Tracking 1–25
1.4 TEACH PENDANT 1–26
1.4.1 Overview 1–26
1.4.2 Additional *i*Pendant Features 1–39
1.5 FANUC ROBOTICS SOFTWARE 1–65
1.5.1 FANUC Robotics Software Overview 1–65

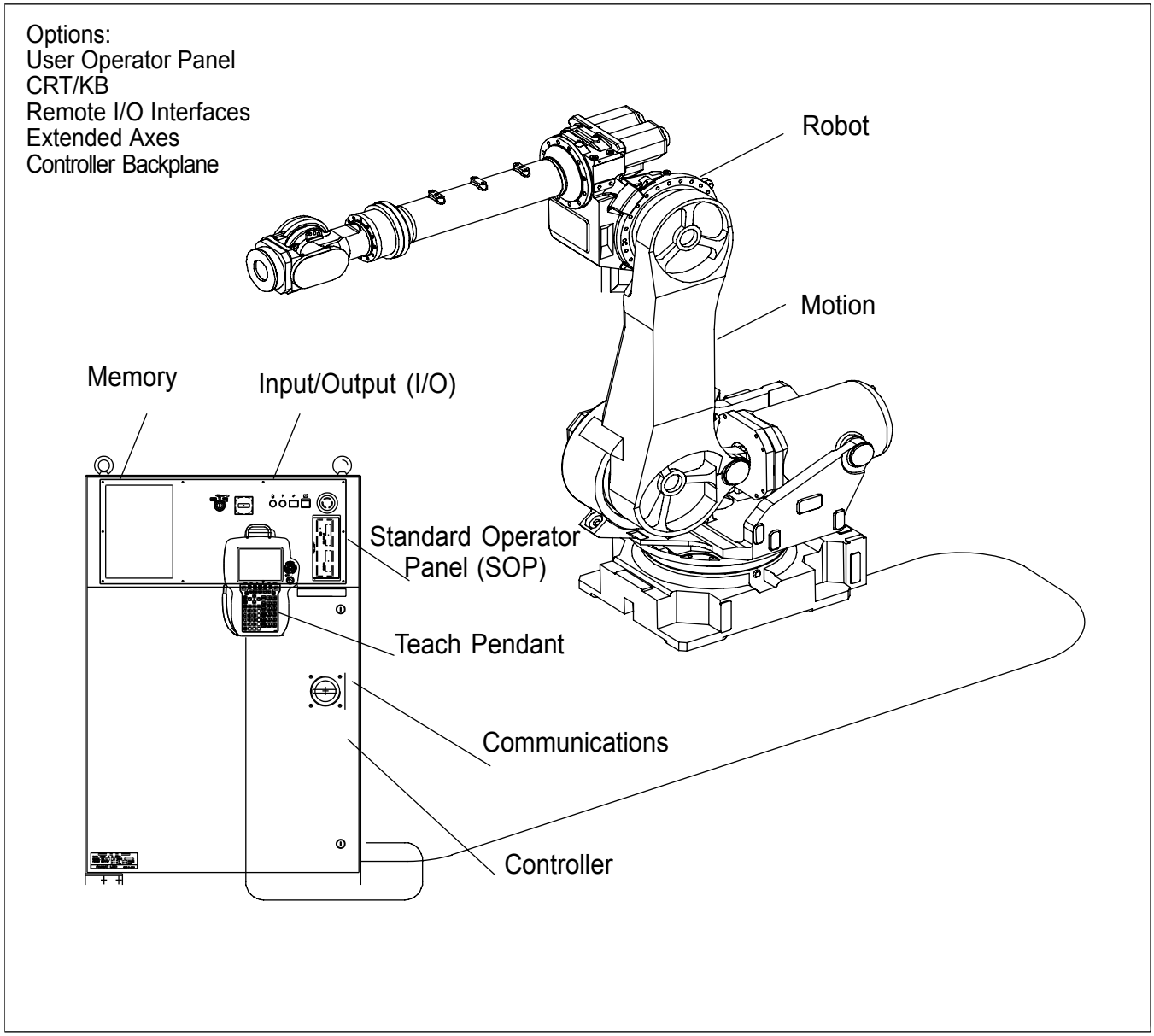
1.5.2 Set Up 1–66
1.5.3 Program 1–66
1.5.4 Test 1–66
1.5.5 Run Production 1–67

1.1 OVERVIEW

The robot system consists of FANUC Robotics software, and the SYSTEM R-30*i*A controller, referred to as the R-30*i*A controller, controller, or robot controller. The robot system provides you with the total solution for all your robotic needs.

Figure 1–1 displays a typical robot system: the robot, controller, and external controller devices. Your system layout might vary depending on the kind of equipment you are using. Each of the items called out in Figure 1–1 are described in more detail further in this chapter.

Figure 1–1. System Overview



1.2 ROBOT

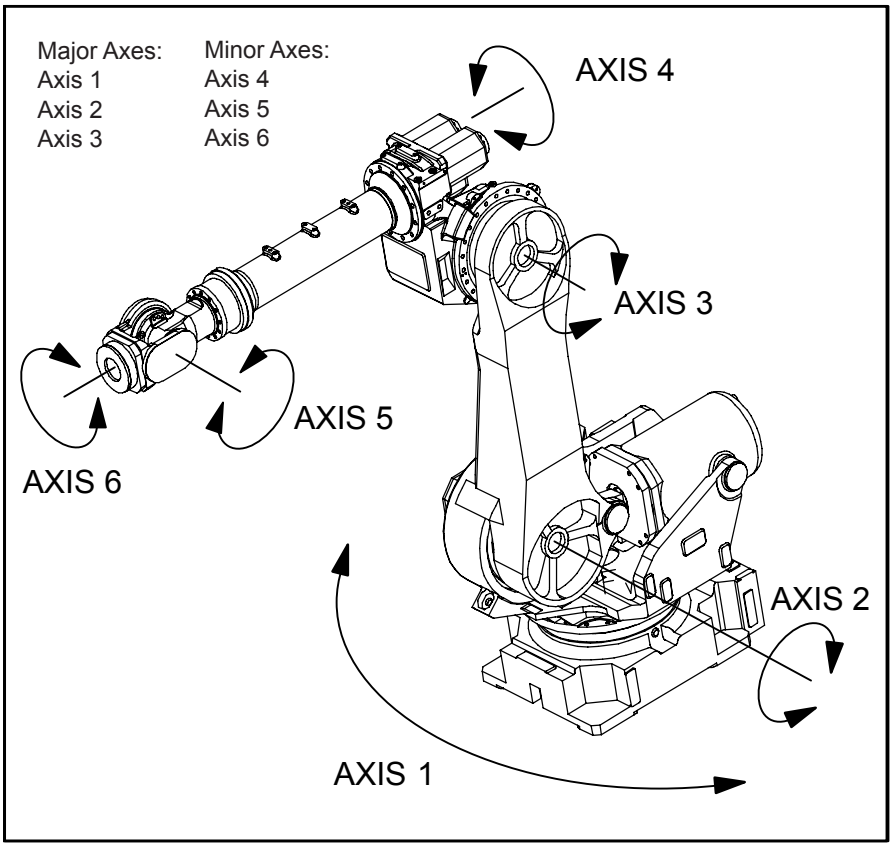
1.2.1 Robot Overview

A robot is a series of mechanical links driven by servomotors. The area at each junction between the links is a *joint*, or *axis* . The first three axes make up the major axes. The last three axes are the minor axes. A robot is classified by the number of linear and rotational major axes.

The major axes (1, 2, and 3) and minor axes (4, 5, and 6) move the tooling at the end of the robot arm. The movements are rotational twisting, up-and-down, and side-to-side motions.

The major axes and minor axes are shown in Figure 1–2 .

Figure 1–2. Major and Minor Axes



1.2.2 Robot Models

A variety of FANUC Robotics robot models can be used. Contact your FANUC Robotics Technical Representative for more information about the kinds of robots you can use.

1.2.3 HandlingTool End-of-Arm Tooling (EOAT)

If you are using HandlingTool with the PalletTool option , single and double case grippers are available as options. Figure 1–3 and Figure 1–4 show examples of the kinds of grippers you might use with PalletTool.

Figure 1–3. Finger Type Gripper

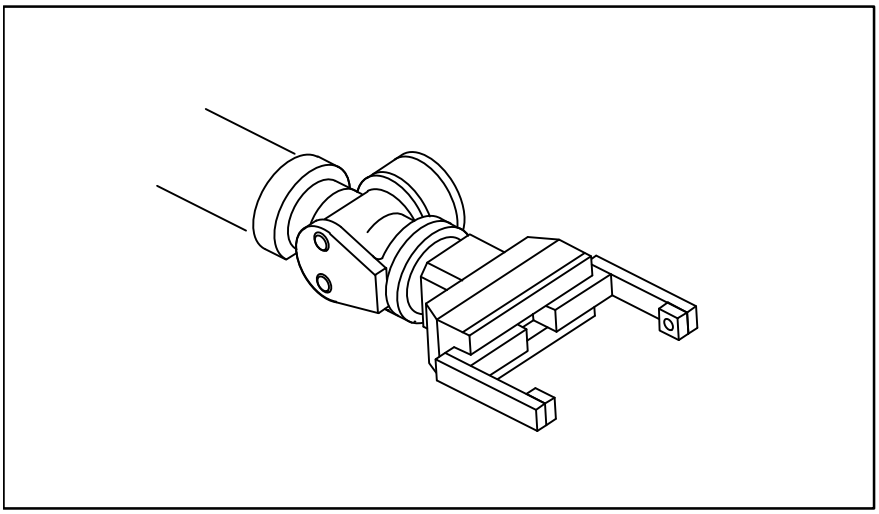
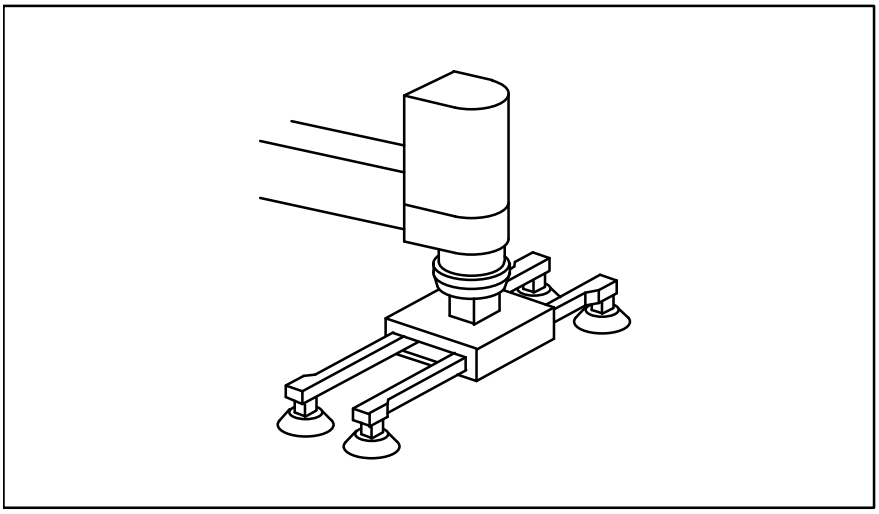


Figure 1–4. Vacuum Type Gripper



1.2.4 Extended Axes

Extended axes are the available axes controlled by the controller beyond the standard number of axes. There is a limit of three extended axes per motion group. The controller can control a maximum of 40 axes.

HandlingTool applications generally use extended axes on jobs that require a sliding axis, rail tracking, or line tracking.

Refer to Section 1.3.13 for more information about extended axes.

If you are using ArcTool, extended axes can include the following:

- Positioner Tables
- Head-tail Stock Positioners
- Ferris Wheel Positioners
- Tracks
- Combinations

1.3 CONTROLLER

1.3.1 Controller Overview

The controller contains the power supply, operator controls, control circuitry, and memory that direct the operation and motion of the robot and communication with external devices. You control the robot using a *teach pendant* or an *operator panel*.

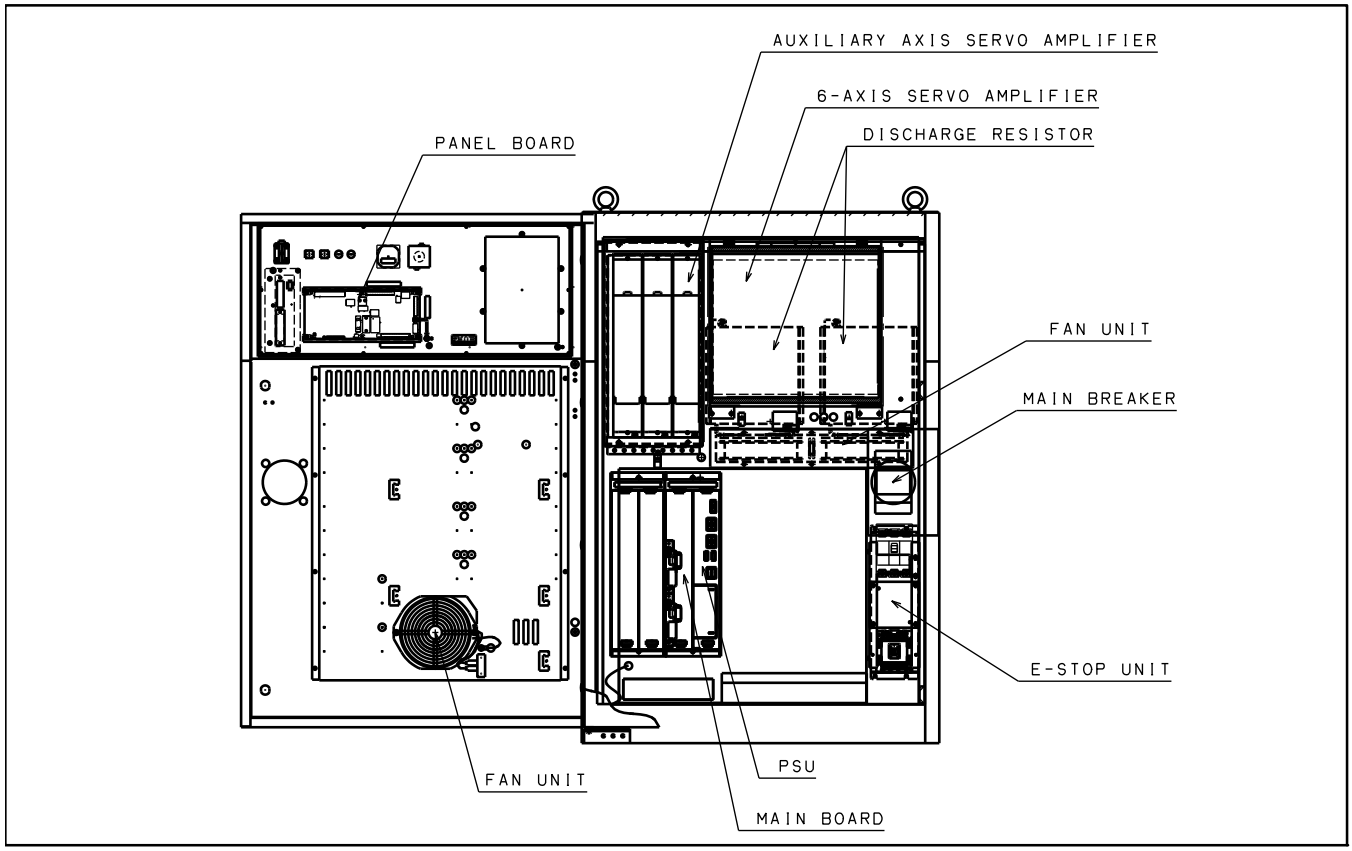
Some systems contain an optional *cathode ray tube/keyboard (CRT/KB)* or an optional user operator panel (UOP) that provides a remote user interface to the controller.

The motion system directs robot motion for all robot axes, including any *extended* axes and up to seven additional motion groups.

Controller memory stores the software in addition to any user-defined programs and data.

The controller is shown in Figure 1–5 .

Figure 1–5. Controller

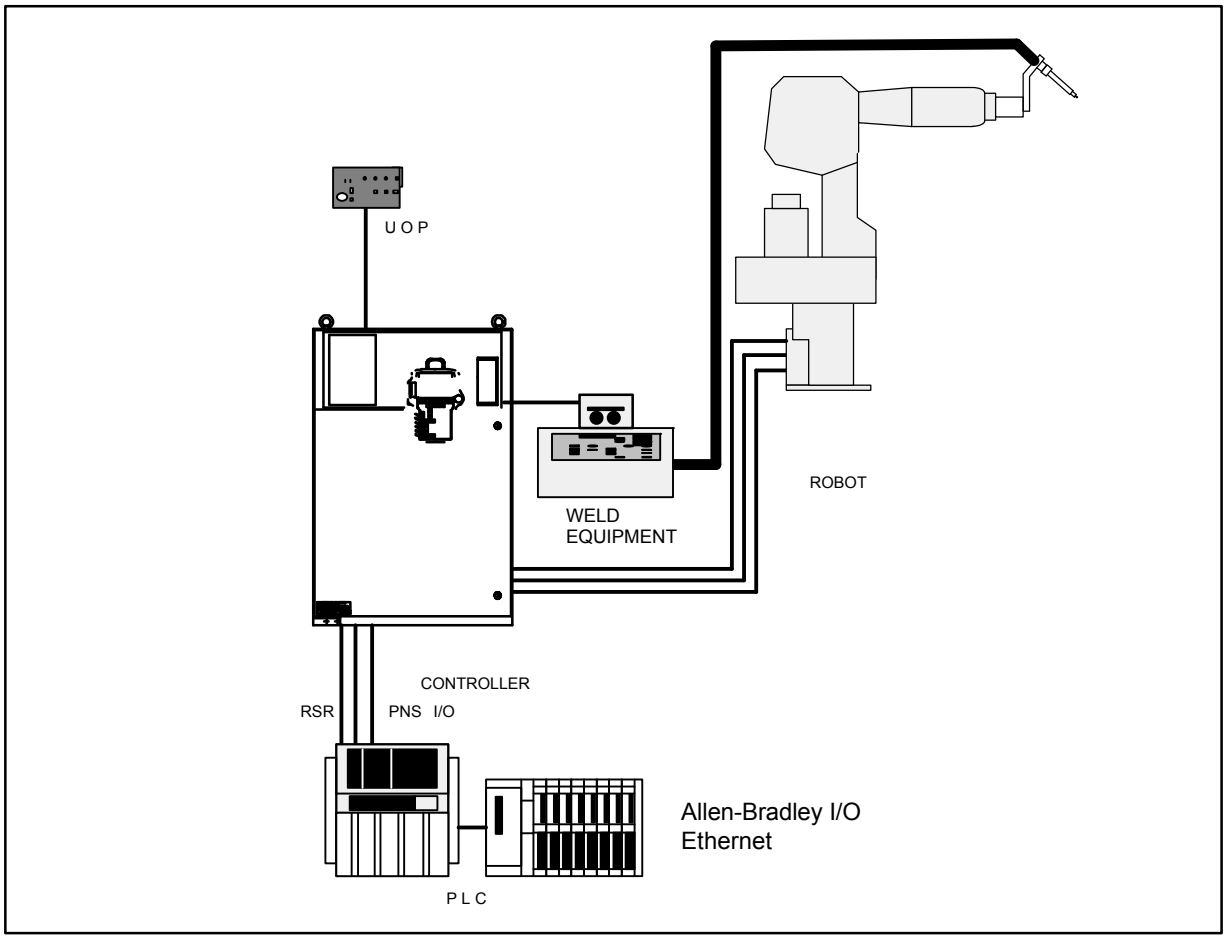


The controller can communicate with a variety of devices. Its I/O system provides an interface between the system software through I/O signals and serial communication ports to external devices. Remote I/O interfaces allow the controller to send signals to a remote device over a single cable. Consult your FANUC Robotics representative for more information.

The controller also provides the capability to work with external or *off-line devices*. An off-line device is any device, other than the controller, used for programming.

See Figure 1–6 .

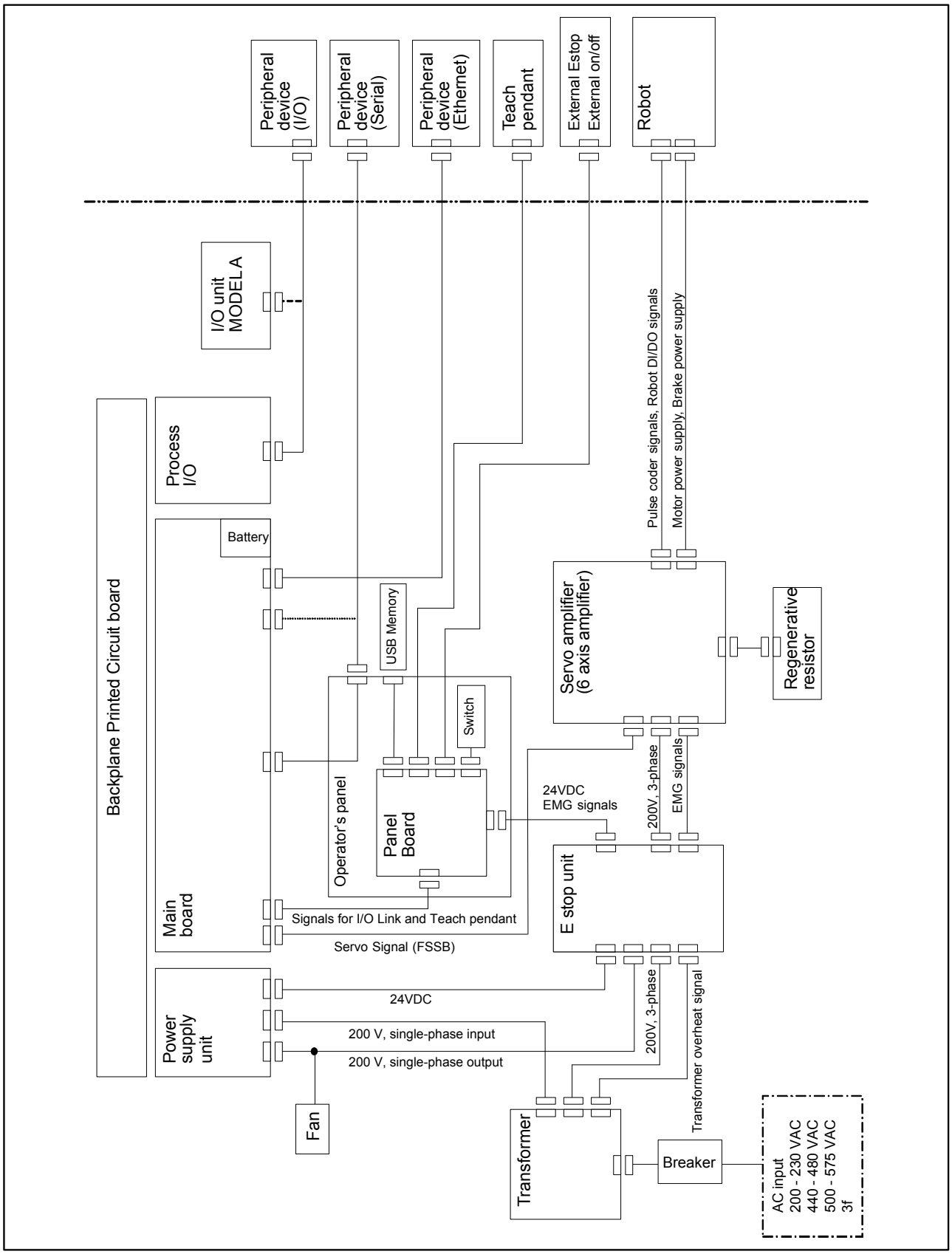
Figure 1–6. Controller Capabilities



The controller is configurable internally depending on the number and kinds of external devices that you have in your system.

See Figure 1–7 .

Figure 1–7. Controller Possible Configuration



1.3.2 Teach Pendant

The teach pendant is a hand-held operator interface device that displays the software menus.

There are two kinds of teach pendants available:

- Monochrome
- *i*Pendant (two models are available)

Refer to Section 1.4 for more information about the teach pendant.

1.3.3 Standard Operator Panel

The operator panel contains buttons, keyswitches, and connector ports and is part of the controller. Refer to the "Status Indicators" section of your application-specific *Setup and Operations Manual* for information on operator panel status indicators. See Figure 1–8 .

Figure 1–8. Controller Standard Operator Panel

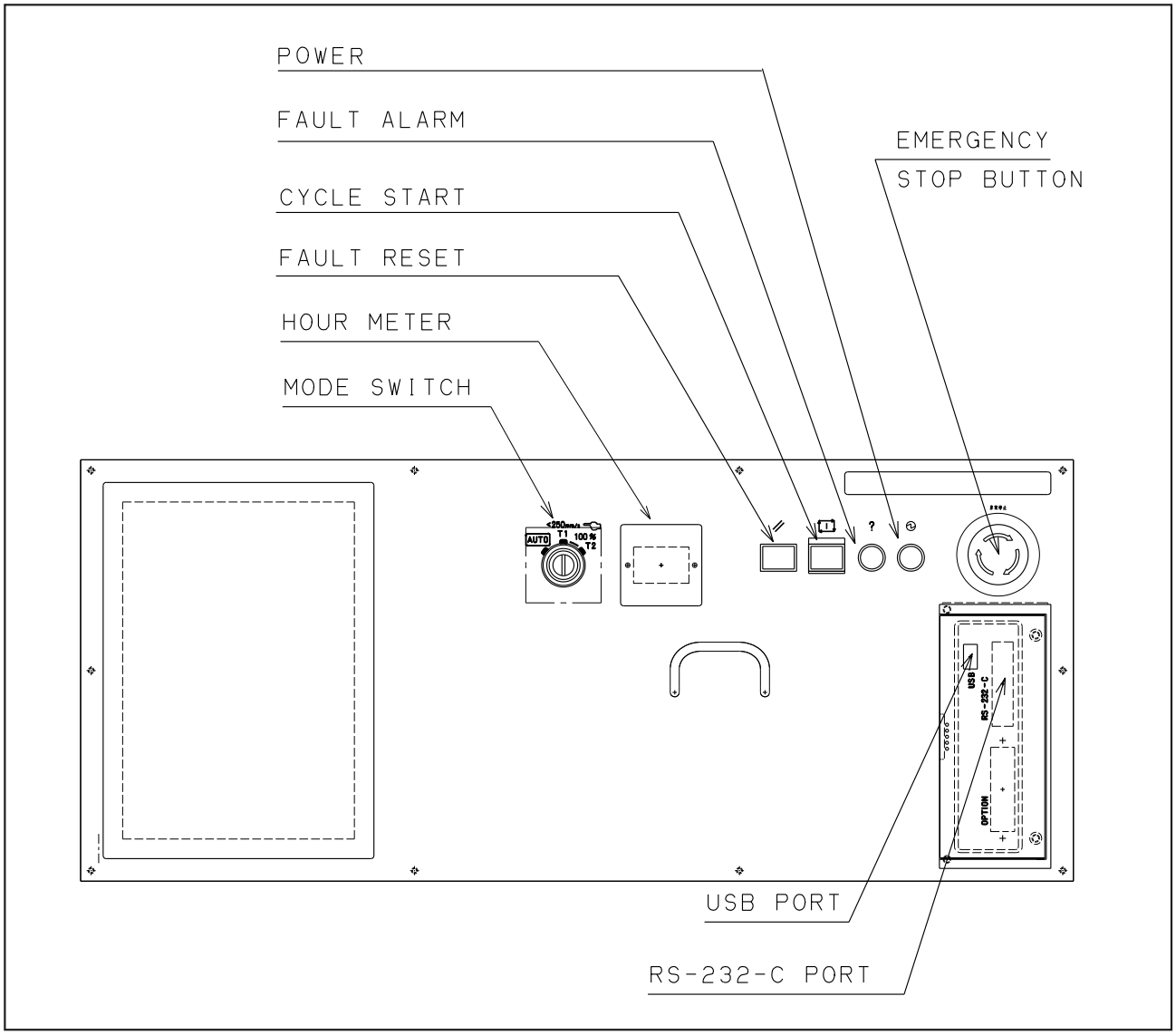


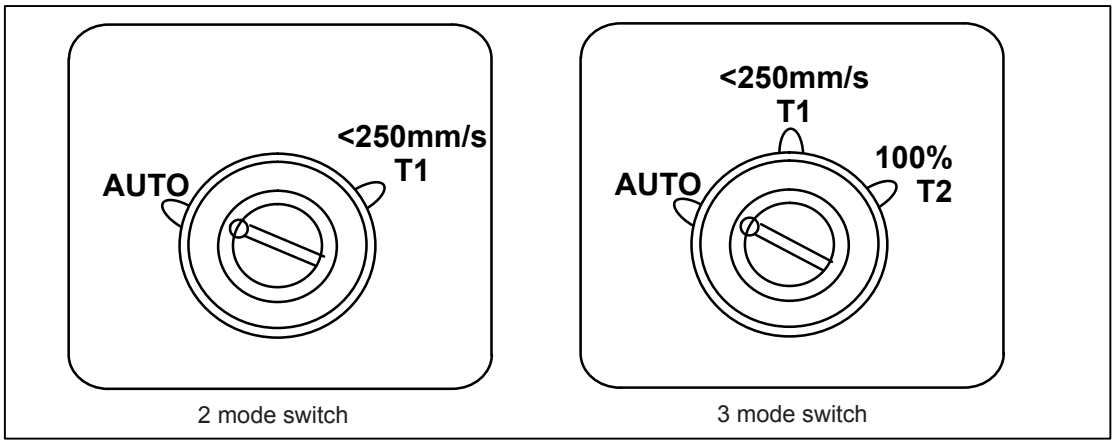
Table 1–1. Standard Operator Panel Buttons

ITEM	DESCRIPTION
EMERGENCY STOP BUTTON	This button stops the robot immediately. Turn the Emergency stop button clockwise to release it.
FAULT RESET BUTTON	This button releases the alarm state.
CYCLE START BUTTON	This button starts the currently selected program This button is lit when the program is being started.

1.3.4 MODE SELECT Switch

The MODE SELECT switch is a keyswitch installed on the controller operator panel. You use the MODE SELECT switch to select the most appropriate way to operate the robot, depending on the conditions and situation. The operation modes are AUTO, T1, and T2. See Figure 1–9 .

Figure 1–9. Mode Select Switch



When you change the mode using the MODE SELECT switch, the robot is paused and a message indicating which mode is selected is displayed at the top of the teach pendant screen. This message will be overwritten by status or error messages during operation. The mode that is currently selected is displayed immediately below the first message, and remains displayed. For example, if T1 mode is selected, you will see a screen similar to the following:

```
SYST-038 Operation mode T1 Selected
TEST1              LINE 9999  T1  PAUSED
UTILITIES Hints               JOINT  10 %
```

You can lock the keyswitch in the AUTO or T1 modes by removing the key from the switch. You cannot remove the key from the keyswitch when the key is in the T2 position.

Note If you change the mode from T1 or T2 to AUTO and the DEADMAN switch is pressed, a system error will occur and the mode will not change to AUTO until the DEADMAN switch is released

Note When an invalid combination is detected by the MODE SELECT switch, programs can be activated only from the teach pendant and the robot cannot operate at speeds of more than 250 mm/s. If a program is being executed at that time, it is forcibly terminated. Refer to the description of the SYST-037 error message for more information.

The operation modes you can select using the MODE SELECT switch are described in the following sections.

T1 (<250mm/s): Test Mode 1

Program activation - Programs can be activated from the teach pendant only. However, programs can be activated only when the teach pendant is enabled and when the DEADMAN switch is in the center position.

Robot speed

- During Cartesian jogging, Cartesian speed is less than 250 mm/sec and joint speed is less than 10% of the maximum joint speed.
- During joint jogging, face plate speed is less than 250 mm/sec.
- During program test run at 100% override, the robot's speed will be the program speed if the program speed is below the*T1 mode safe speed* . T1 mode safe speed is defined as 250 mm/sec for TCP and 10% of maximum joint speeds. If the program speed is above the T1 mode safe speed, robot motion will be executed at the T1 mode safe speed. With lower overrides, the robot speed is reduced proportionally according to the override setting.

Robot motion with Line Tracking software option installed

- When the conveyor is stationary, the robot behavior is exactly as described above under "Robot speed."
- During program test run at any override, if the conveyor begins to move, robot motion will immediately stop, and the error LNTK-041 "Encoder is moved in T1 mode" will be displayed.
- If program test run is attempted while the conveyor is moving, the robot will not move; instead, the error LNTK-041 "Encoder is moved in T1 mode" will be displayed.

Safety equipment - The safety fence is bypassed.

Locking the mode - You can lock the switch in T1 mode by removing the key from the switch.

Possible errors

- If you turn the teach pendant ON/OFF switch to OFF when the switch is in T1 mode, the robot stops and an error message is displayed. To remove the error, turn the teach pendant ON/OFF switch to ON and press RESET.
- If you have set the singularity stop system variable, $PARAM_GROUP[n].$T1T2_SNGSTP, to TRUE, the robot will stop at singularity points while in T1 mode. If you change the value of this variable, you must turn the controller off and then on again for the change to take effect.

T2 (100%): Test Mode 2

Program activation - Programs can be activated from the teach pendant only. However, programs can be activated only when the teach pendant is enabled and the DEADMAN switch is in the center position.

Robot speed

- During Cartesian jogging, Cartesian speed is less than 250 mm/sec and joint speed is less than 10% of the maximum joint speed.
- During joint jogging, face plate speed is less than 250 mm/sec.
- During program test run, full program speed is allowed, and the override can be changed from low to 100%.

Safety equipment - The safety fence is bypassed.

Locking the mode - You cannot lock the switch in T2. You cannot remove the key from the switch in this mode.

Possible errors

- If you turn the teach pendant ON/OFF switch to OFF when the switch is in T2 mode, the robot stops and an error message is displayed. To remove the error, turn the teach pendant ON/OFF switch to ON and press RESET.
- If you have set the singularity stop system variable, $PARAM_GROUP[n].$T1T2_SNGSTP, to TRUE, the robot will stop at singularity points while in T2 mode. If you change the value of this variable, you must turn the controller off then on again for the change to take effect.

AUTO: Automatic Mode

Program activation - You must select AUTO mode and satisfy all other required conditions to enable the activation of programs from remote devices connected through the peripheral I/O. When the switch is in AUTO mode, you cannot start programs using the teach pendant. Refer to the “Test Cycle” section in the “Testing a Program and Running Production” chapter.

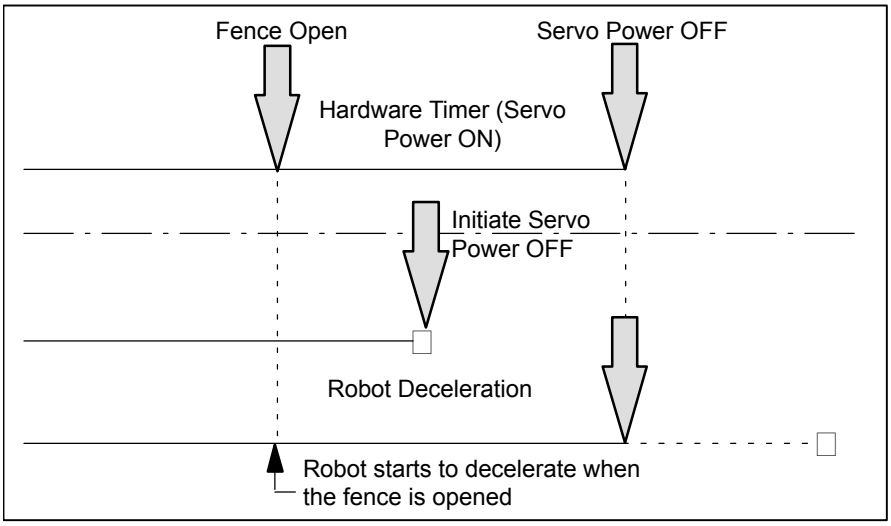
Robot speed - The robot can be operated at the specified maximum speed.

Safety equipment - The safety fence is monitored. If the safety fence is opened during program execution (Figure 1–10):

- **Case [1]** - If the robot deceleration time is less than the *hardware timer*, then the robot will decelerate to a stop. This is a controlled stop. When the robot stops, servo power OFF is initiated.

- **Case [2]** - If the robot deceleration time is greater than the hardware timer, then the robot will decelerate for the duration of the hardware timer and then stop abruptly when the hardware timer expires. When the hardware timer expires, servo power is turned OFF.

Figure 1–10. Effect of Opening the Safety Fence While in AUTO Mode



The system variable $PARAM_GROUP.$LC_QST P_ENB defines whether the condition specified by the condition monitor (condition handler) function will be triggered during robot deceleration. By default, the condition, if it exists, is triggered during deceleration ($LC_QSTP_ENB = TRUE). When $LC_QSTP_ENB = FALSE, a condition, if it exists, is not triggered during deceleration.

Refer to the*System Software Reference Manual* for more information on these system variables

Locking the mode - You can lock the switch in AUTO mode by removing the key from the switch.

Possible errors

- If you turn the teach pendant ON/OFF switch to ON when in AUTO mode, the robot stops and an error message is displayed. To remove the error, turn the teach pendant ON/OFF switch OFF and press RESET.
- If you have set the singularity stop system variable, $PARAM_GROUP[n].$AUTO_SNGSTP, to FALSE, the robot will pass through singularity points while in AUTO mode. If you change the value of this variable, you must cycle power for the change to take effect.

1.3.5 Robot Stop Variation

When the EMERGENCY STOP button on the operator panel or teach pendant is pressed, the robot stops immediately.

An emergency stop condition can be created not only when the EMERGENCY STOP button is pressed, but also by a combination of operation mode selection, teach pendant ON/OFF switch, DEADMAN switch, and safety fence open and close. Refer to Table 1–2 .

Note If the DEADMAN switch is fully pressed, robot motion will not be allowed. This is the same as when the DEADMAN switch is released.

Table 1–2. Robot Servo Status

Mode	TP-ON/OFF Switch	DEADMAN Switch	Fence	SERVO Status	Motion Possible
AUTO	ON	pressed	open	OFF	No
			close	ON	No
		released or pressed extremely firmly	open	OFF	No
			close	OFF	No
	OFF	pressed	open	OFF	No
			close	ON	Yes
		released or pressed extremely firmly	open	OFF	No
			close	ON	Yes
T1 or T2	ON	pressed	open	ON	Yes
			close	ON	Yes
		released or pressed extremely firmly	open	OFF	No
			close	OFF	No
	OFF	pressed	open	OFF (n/a for PaintTool)	No
			close	OFF (n/a for PaintTool)	No
		released or pressed extremely firmly	open	OFF	No
			close	OFF	No

1.3.6 User Operator Panel (UOP)

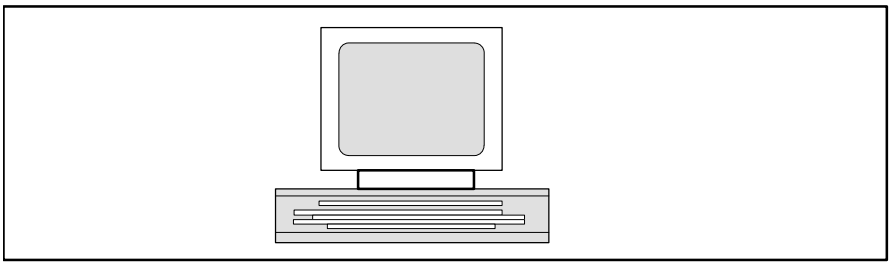
Your system might be equipped with a user operator panel (UOP). A UOP is a customized operator panel that is wired to the controller. It can be a custom control panel, a cell controller, or a host computer. Your company should provide the information necessary to operate this panel.

1.3.7 CRT/KB

The CRT/KB provides an optional alternative operator device to the teach pendant. The CRT/KB is external to the controller. The CRT/KB connects to the controller via a cable connected to the RS-232-C port.

The CRT/KB can be used as an alternative menu to display the FANUC Robotics software. Figure 1–11 shows the CRT/KB. The CRT/KB allows you to perform most teach pendant functions except those that are related to robot motion. Functions that cause robot motion can only be performed using the teach pendant.

Figure 1–11. CRT/KB



1.3.8 Emergency Stop Devices

Your FANUC Robotics robot has the following emergency stop devices.

- Two emergency stop buttons (one installed on the operator panel and one on the teach pendant)
- External emergency stop (input signal)

When the EMERGENCY STOP button is pressed, the robot stops immediately in all cases. The external emergency stop outputs or inputs the emergency stop signal for peripheral devices (such as a safety fence or gate). The signal terminal is on the controller and inside the operator panel.

1.3.9 Communications

The controller has the capability of serial communication using:

- RS-422 serial port, which is used for the teach pendant
- RS-232- C and RS-422 serial ports, which can be used for
  - — CRT/KB
  - — FANUC Robotics Industrialized Terminal
  - — DEC VT-220 terminal
  - — IBM PC compatibles
  - — Debug monitor

Refer to the "Program and File Manipulation" chapter in this manual for more information about setting up the RS-232-C and RS-422 serial ports.

The controller also has the capability of using standard TCP/IP and UDP/IP protocols. FANUC Robotics-supplied options are as follows:

- FTP
- Internet Connectivity and Customization
- PCIF
- Cimplicity/HMI
- User Socket Messaging

In addition, the following I/O products are available:

- ControlNet
- DeviceNet Interface
- CC-link
- Ethernet/IP
- Ethernet Global Data
- Interbus
- Profibus DP

1.3.10 Input/Output (I/O)

The I/O system provides the interface between the controller, teach pendant, robot, and any other external device in your workcell. Controller I/O can consist of the following kinds of I/O:

- User Operator Panel (UOP) Inputs (UI)
- User Operator Panel (UOP) Outputs (UO)

- Standard Operator Panel (SOP) Inputs (SI)
- Standard Operator Panel (SOP) Outputs (SO)
- Robot Inputs (RI)
- Robot Outputs (RO)
- Digital Inputs (DI)
- Digital Outputs (DO)
- Group Inputs (GI)
- Group Outputs (GO)
- Analog Inputs (AI)
- Analog Outputs (AO)
- PLC Inputs
- PLC Outputs

These kinds of I/O are provided by devices, including

- CC-link
- ControlNet
- Distributed Modular I/O (ArcTool and PaintTool only)
- DeviceNet
- Ethernet Global Data (EGD)
- Ethernet/IP
- Interbus
- Modular I/O (Model A and Model B)
- Process I/O (option)
- Profibus-DP

Note that not all Tools support all of these options.

The **quantity** of I/O can change, except for RI/RO, UOP, and SOP I/O signals, which are fixed. The number of RI and RO signals can vary slightly depending on the number of axes in your system.

1.3.11 Remote I/O Interfaces

The controller has the capability to use certain signals from a remote device. These signals can include

- UOP signals

- Safety fence
- RSR and PNS
- External Emergency stop

1.3.12 Motion

The robot system uses the motion system to control robot motion. The motion system regulates the characteristics of robot movement including path trajectory, acceleration/deceleration, termination and speed of the robot.

In robotic applications, single segment motion is the movement of the*tool center point* (TCP) from an initial position to a desired destination position. The TCP is the point on the end-of-arm tooling at which the work is to be done.

Motion Type

There are three different types of motion: *linear*, *circular*, and*joint* . You use these motion types to perform certain tasks. For example, you use linear motion if the robot must move in a straight line between two positions. You use circular motion when the positions must be along the arc of a circle. Joint motion is generally the motion type used at each position when it is not important how the robot moves from position to position.

Note Joint motion is not allowed in Line Tracking.

Termination Type and Speed

Termination type can be specified as fine and continuous. Speed can be specified in either length (mm/sec, cm/min, inch/min), degrees of angle units, or length of time to execute a move.

TCP Speed Prediction (DispenseTool, DispenseTool Plug-in, and SpotTool+ only)

TCP speed prediction improves the control of speed-dependent devices, such as dispensing equipment. The speed prediction lead-time can be adjusted to accommodate a variety of equipment and materials. When you set up the dispensing equipment, you can specify whether to use TCP speed prediction in the computation of the flow command.

Motion Groups

The controller optionally allows you to create up to eight *motion groups.* By default, the first motion group is a robot arm. Additional motion groups can be set up to perform tasks that are executed simultaneously with those of the robot.

Each motion group can contain up to a maximum of nine axes. The total number of axes cannot exceed 40. Motion groups are independent, but a maximum of two motion groups can perform Cartesian coordinated linear or circular interpolated motion **within a single program.**

The R-30iA controller can support

- Up to four full kinematics devices (robot mechanical units)
- Up to eight motion groups
- Up to 40 axes
  - — Up to 24 axes can be supported on the Main CPU PCB.
  - — Up to 16 axes can be supported on an Auxiliary Axis PCB.

In some cases multiple robot followers will be present in a robot program. **If** the part paths are identical or mirrored (i.e. a motion segment on the first follower robot and the second, etc. are the same length), all the follower robots will be executing coordinated motion. In practice, there are small path differences between the follower robots, so the follower with the longest segment will dictate the time for the move AND will be exactly coordinated with the leader. The other follower robots will execute at a lower speed than the programmed speed.

Note Multiple motion groups for dispensing applications are not supported on the SpotTool+ application software.

1.3.13 Extended Axes

Extended axes are the available axes controlled by the controller beyond the standard number of robot axes. Depending on your system setup, many applications use extended axes on jobs that require a rotary, sliding axis, rail tracking, opening devices, or line tracking.

For the R30iA controller:

- Up to four full kinematics devices (robot mechanical units) are supported on one controller
- Non-robot motion groups of one to four axes can be defined. The maximum number includes extended axes. This includes General Positioner, Basic Positioner, and Independent Axis devices.
- One to three extended axes can be added to a motion group.
  - — The extended axes cannot be used independently of the motion group.
  - — Each extended axis adds a position data field (E1, E2, E3) to the motion group data.
  - — The Index axis device is one axis by definition and the Arc Positioner device is a two axes motion group device. Neither of these can have additional axes installed.

    Note Starting with V7.40, the limitation of a maximum of two Index Devices has been removed

- You cannot exceed nine axes per robot motion group; you cannot exceed four axes per non-robot motion group.

If you are using ArcTool, welding applications generally use extended axes on jobs that require a head-tail stock (1 axis) or tables (2–3 axes).

If you are using PaintTool, the P-120 and P-145 robots only support 6 axis configurations.

1.3.14 Multiple Applications (option)

If the multiple application option is installed, you can use the following applications with the SpotTool+ software:

- DispenseTool plug-in to SpotTool+
- HandlingTool plug-in to SpotTool+

At a controlled start, you specify which one of the applications you want to use on the controller. After you select the application at a controlled start, you can

- Select the application you want to use for teach pendant operation by choosing an item on the FUNCTIONS menu; then, all teach pendant keys and screens will correspond to the application you selected.
- Select the application you want to use in the program header information of a teach pendant program; then, all program instructions will correspond to the application you selected.

1.3.15 Controller Backplane

A 4-slot backplane is available on the controller. The 4-slot backplane comes equipped with the following:

- Power supply PCB
  - — One wide mini slot
- Main PCB
  - — Two mini slots
- Two full-size slots

1.3.16 Memory

There are three kinds of controller memory:

- Dynamic Random Access Memory (DRAM)
- A limited amount of battery-backed static/random access memory (SRAM)
- Flash Programmable Read Only Memory (FROM)

In addition, the controller is capable of storing information externally.

DRAM

DRAM memory is volatile. Memory contents do not retain their stored values when power is removed. DRAM memory is also referred to as temporary memory (TEMP). The system software is executed in DRAM memory. KAREL programs and most KAREL variables are loaded into DRAM and executed from here also.

Note Even though DRAM variables are in volatile memory, you can control their value at startup. Any time that a the program .VR or .PC file is loaded, the values in DRAM for that program are set to the value in the .VR file. This means that there is not a requirement to re-load the VR file itself at every startup to set initial values. If the value of that variable changes during normal operation it will revert to the value it was set to the last time the .VR or .PC file was loaded.

If you want the DRAM variables to be uninitialized at start up you can use the IN UNINIT_DRAM clause on any variable you want to insure is uninitialized at startup. You can use the %UNINITDRAM directive to specify that all the variables in a program are to be uninitialized at startup.

If you have a SHADOW variables and DRAM variables in the same KAREL program, there is a possibility that the power up settings of the DRAM variables could change without loading a .PC/.VR File. In this case the programmer must pay particular attention to the reliance of KAREL software on a particular setting of a DRAM variable at startup. Specifically, the DRAM startup values will **always** retain the values that they had at the end of controlled start. If SHADOW memory is full, the DRAM startup values could be set during normal system operation.

SRAM

SRAM memory is nonvolatile. Memory contents retain their stored values when power is removed. SRAM memory is also referred to as CMOS or as permanent memory (PERM).

The TPP memory pool (used for teach pendant programs) is allocated from PERM. KAREL programs can designate variables to be stored in CMOS. A portion of SRAM memory can be defined as a user storage device called RAM Disk (RD:).

Flash memory (FROM)

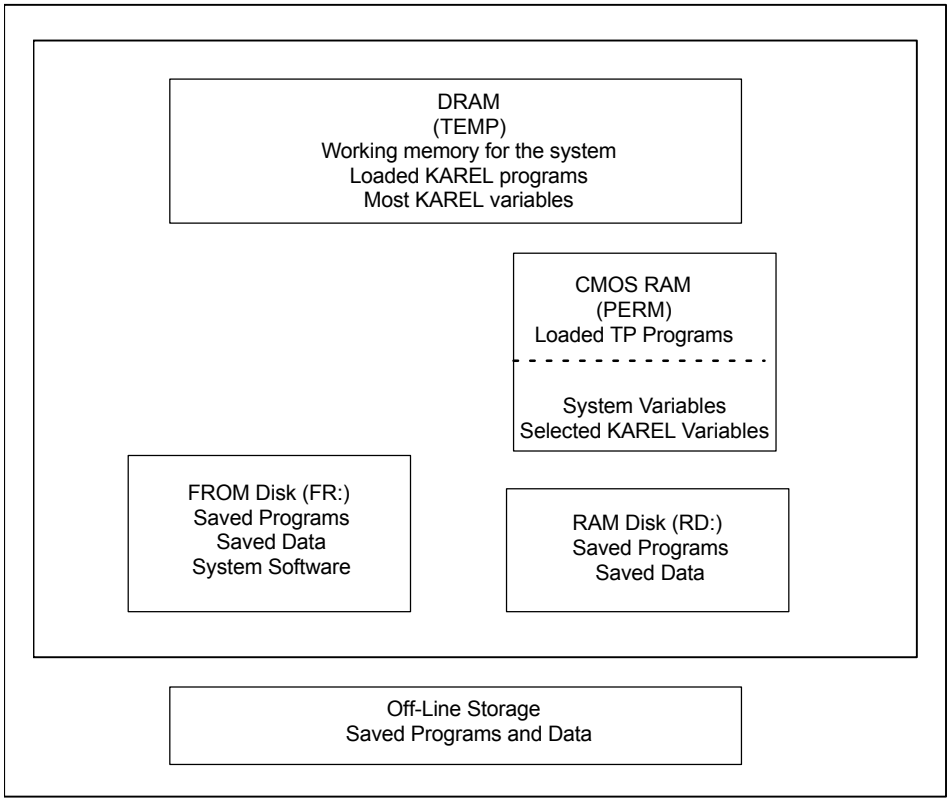
FROM memory is nonvolatile. Memory contents retain their stored values when power is removed.

FROM is used for permanent storage of the system software. FROM is also available for user storage as the FROM device (FR:).

SHADOW

Shadow memory provides the same capabilities as SRAM. Any values set in shadow are non-volatile and will maintain their state through power cycle. Shadow memory is intended for data which tends to be static. Storing dynamic variables in shadow memory, such as FOR loop indexes or other rapidly changing data, is not efficient.

Figure 1–12. Controller Memory



External Storage

You can back up and store files on external devices. You can use the following devices:

- Memory card
- Ethernet via FTP
- USB Memory Stick

1.3.17 Line Tracking

Line tracking is an optional feature that enables a robot to treat a moving workpiece as a stationary object. The option is used in conveyor applications, where the robot must perform tasks on moving workpieces without stopping the assembly line.

The DispenseTool line tracking feature supports all of the TPP line tracking features, plus some additional features specific to dispensing application. These are:

- Tracking schedule number and boundary number are set in the process program header

This eliminates the needs to specify the tracking schedule and the boundary set inside the program.

- Automatic detection of extreme position

The system automatically scans the process programs to determine the most upstream (extreme) position within a sealing path (positions between SS and SE instructions) when the program is invoked for the first time after an edit. The robot will not start dispensing until all points in the seam comes within the tracking boundary.

- Part detect queue

For the system where the conveyor length is long, and more than one workpiece could come in the same conveyor before one job finishes, the system keeps track of the part locations on the conveyor so that the correct tracking frame become effective when the job starts.

Note Job Queue is no longer supported. The cell controller will be required to keep track of jobs corresponding to each part detect.

- GET PART DETECT Macro

In conjunction with the job queue, the GET PART DETECT macro will get the part location on the conveyor, and make the appropriate value effective for the tracking frame.

With these features, a tracking job can be as simple as,

```
1: ! Move to the Pounce position
2: J P[1: POUNCE] 100% FINE
3: ! Wait for part to come in
4: GET PART DETECT
5: ! Now call the tracking process.
6: ! Tracking schedule in program
7: ! header of PROC0001 will be used
8: CALL PROC0001
```

1.4 TEACH PENDANT

1.4.1 Overview

1.4.1.1 Styles

The teach pendant is a hand-held operator interface device that displays the software menus. It is connected to the controller via a cable that plugs into either the MAIN CPU board inside the controller or, if it is a disconnectable teach pendant, to the operator panel.

The teach pendant is the device you use to

- Move the robot
- Set up your application
- Create and edit programs
- Test programs
- Set up production
- Check status
- Perform manual functions
- Configure the application
- Access diagnostic tools (optional)
- Access Internet/intranet websites (optional)

There are two kinds of teach pendants available:

- Monochrome
- *i*Pendant

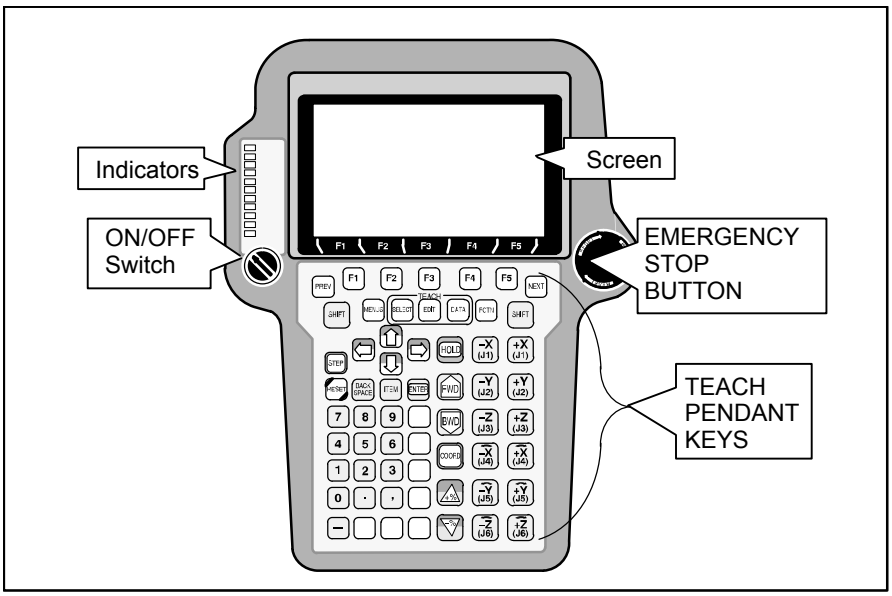
The monochrome teach pendant is also available as an Intrinsically Safe teach pendant (ISTP). The ISTP, used in most hazardous paint environments, plugs into the intrinsically safe barrier (ISB)

unit within the controller and consists of an additional eleven status indicators displayed on the teach pendant screen.

The monochrome teach pendant, shown in Figure 1–13 provides

- A 16 line x 40 character teach pendant screen
- Eleven status indicators

Figure 1–13. Monochrome Teach Pendant



Both the monochrome teach pendant and the *i*Pendant provide

- Teach pendant keys designed to make the FANUC Robotics software easy to use
- An ON/OFF switch, DEADMAN switches, and an EMERGENCY STOP button

In addition to these features the optional *i*Pendant, which is available in two styles shown in Figure 1–14 , provides:

- A color graphics interface
- Popup menus
- Multiple screens displayed simultaneously
- Internet/intranet access
- Integrated Help and Diagnostics

- User-customized help and web pages
- Customized displays
- An optional touch screen interface
- A USB port (available on some *i*Pendants)

Note Refer to Section 1.4.2 for information on additional *i*Pendant features.

Note The *i*Pendant is not available for use in hazardous paint environments and is not supported for use with PaintTool.

Figure 1–14. iPendant Overview



1.4.1.2 Motion Control Switches

The monochrome teach pendant and the *i*Pendant include keys that are used to display software menus, select options from the teach pendant menus, help you program, move the robot, and perform specific functions.

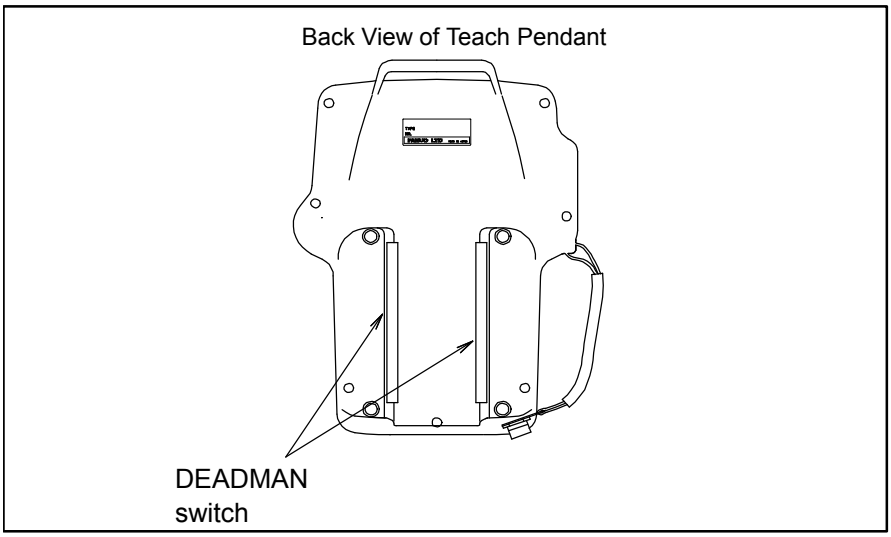
If you are using PalletTool, you do not have to perform special programming functions unless you want to add custom features to your application.

DEADMAN Switch

Two DEADMAN switches are located on the back of both teach pendant styles and are used as enabling devices. Each DEADMAN switch is a three position switch. When the teach pendant is enabled, this switch allows robot motion only while a DEADMAN switch is gripped in the middle position. If the DEADMAN switch is completely gripped, or is released, no motion is allowed. Press the DEADMAN switch in the middle position to run the robot.

See Figure 1–15 .

Figure 1–15. DEADMAN Switch

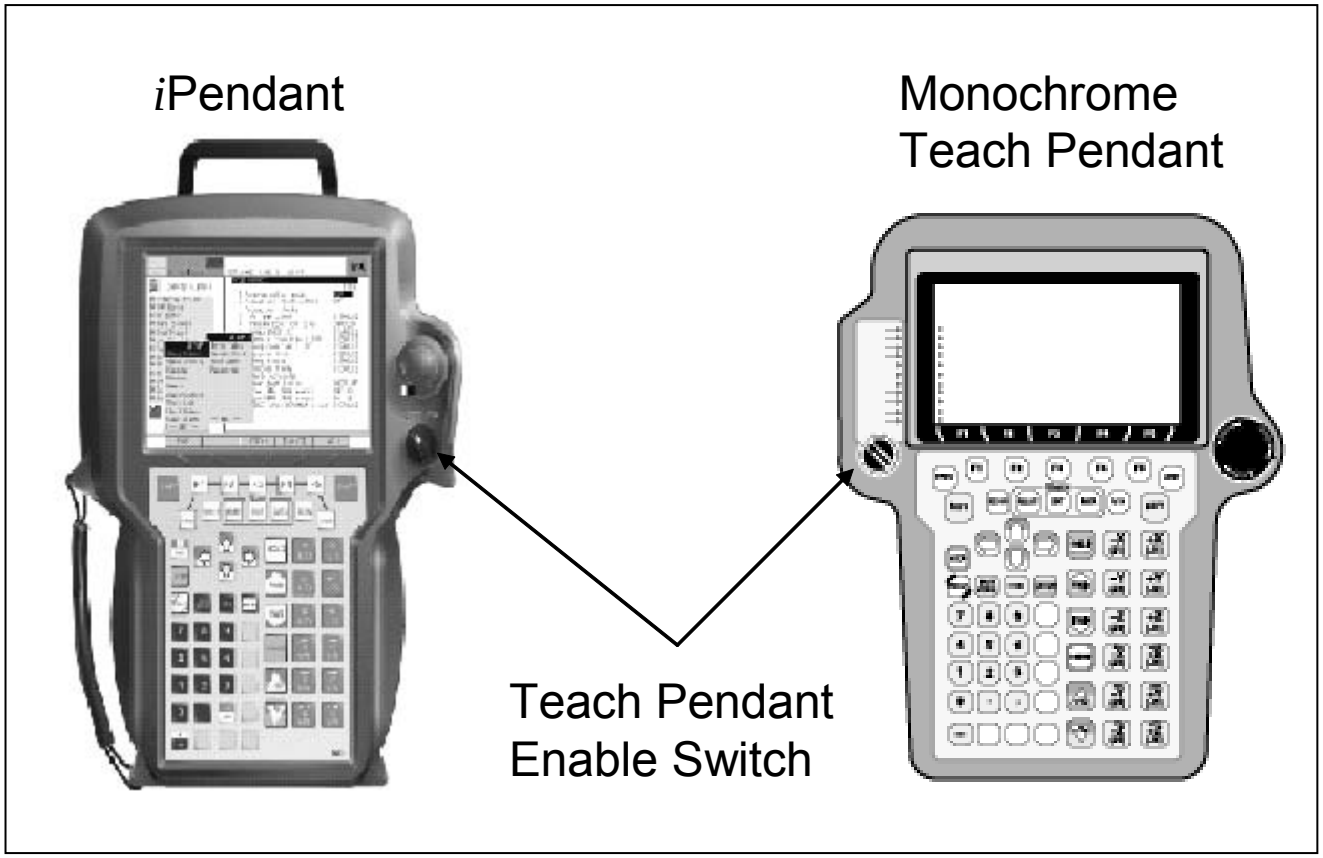


ON/OFF Switch

This switch turns on and off the teach pendant. When the teach pendant is off, you cannot jog the robot, create a program, or test/run a program.

See Figure 1–16 .

Figure 1–16. Teach Pendant Switches

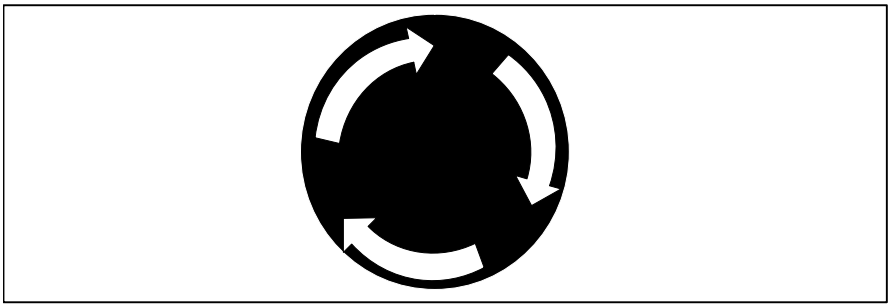


EMERGENCY STOP Button

The red EMERGENCY STOP button is located on the front of both teach pendant styles and can be pressed to stop the robot immediately in case of an accident or failure.

See Figure 1–17 .

Figure 1–17. EMERGENCY STOP Button



1.4.1.3 Status indicators

Teach pendant status indicators indicate the current system condition. Refer to the Status chapter of your application-specific Setup and Operations manual for more information. If you are using PalletTool, refer to the *HandlingTool Setup and Operations Manual* .

1.4.1.4 Teach Pendant Screen

The teach pendant screen displays the software menus. All robot functions can be performed by making selections from these menus. To display the list of menus, press MENUS.

You can display Full and Quick menus when you press the MENUS key. The full menus are a complete list of all available menus. The QUICK menus are a predefined partial list of specific menus. The menu choices available on the Quick menus are application-specific.

For more information about the screen display on the *i*Pendant, refer to Section 1.4.2 .

1.4.1.5 Teach Pendant Keys

The teach pendant has the following keys:

- Navigation and Data Entry Keys
- Robot Motion Keys
- Execution Keys
- Editing Keys
- Application-Specific Keys
- Help/Diagnostic Keys (iPendant only)

Table 1–3. Navigation and Data Entry Keys

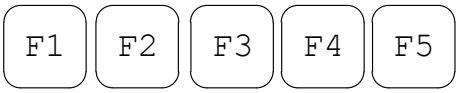
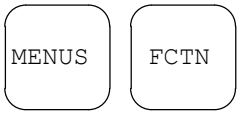
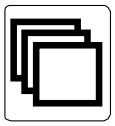
Monochrome Teach Pendant/*i*Pendant Key	Symbolic Teach Pendant Key (if available)	Function
		The F1 through F5 function keys are used to make choices based on the teach pendant display. Each function key has a unique label depending on the menu displayed on the teach pendant screen.
		The PREV key restores the most recent state. In some cases, the screen might not return to the immediately preceding status.
		The next page key is used to display the next set of function keys.
		The MENUS key is used to display the screen menu. The FCTN key is used to display the function menu.
	  	The SELECT key is used to display the program selection screen. The EDIT key is used to display the program edit screen. The DATA key is used to display the program data screen.

Table 1–3. Navigation and Data Entry Keys (Cont'd)

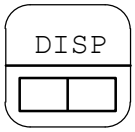
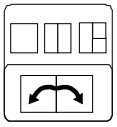
Monochrome Teach Pendant/*i*Pendant Key	Symbolic Teach Pendant Key (if available)	Function
		The DISP key is available only on the *i*Pendant and is used to activate the DISPLAY menu or change the focus. When you press SHIFT and the DISP key together, the DISPLAY menu appears. The DISPLAY menu allows you to change the number of windows displayed to be Single, Double, or Triple. The Status/Single choice displays status in addition to the single window. You can also use it to display help or diagnostics, set up user views and menu favorites, or to display a menu history.
		The FCTN key is used to display the supplementary menu.

Table 1–4. Robot Motion Keys

Monochrome Teach Pendant/*i*Pendant Key	Symbolic Teach Pendant Key (if available)	Function
		The GROUP key is used to switch groups. Hold down the GROUP key and press the numeric key to switch to a specified group. Press the GROUP key and the 0 key together to toggle the sub group.
		The SHIFT key is used to jog the robot, teach the position data, and start a program. The right and left SHIFT keys have the same function.

Table 1–4. Robot Motion Keys (Cont'd)

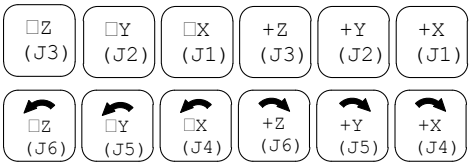
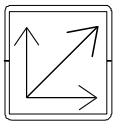
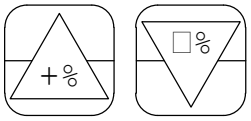
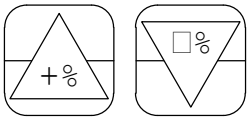
Monochrome Teach Pendant/*i*Pendant Key	Symbolic Teach Pendant Key (if available)	Function
		The jog keys are effective while a SHIFT key is pressed. They are used to jog the robot.
		The COORD key selects a jog coordinate system. Each time the COORD key is pressed, it selects the next jog type in the order: JOINT, JGFRM, World frame, TOOL, USER. When this key is pressed while a SHIFT key is pressed and held down, a jog menu for changing the coordinate system is displayed.
		The override key adjusts the feedrate override. Each time the override key is pressed, it selects the next override in the order: VFINE, FINE, 1%, 2% 3% 4% 5%, 10% 15% 20% and so forth in 5% increments to 100%. (Set $SHFTOV_ENB to alter the override settings.)

Table 1–5. Execution Keys

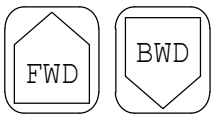
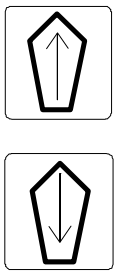
Monochrome Teach Pendant/*i*Pendant Key	Symbolic Teach Pendant Key (if available)	Function
		The FWD key or BWD key (+ SHIFT key) starts a program. When the SHIFT key is released during regeneration, the program halts.

Table 1–5. Execution Keys (Cont'd)

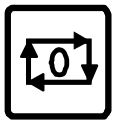
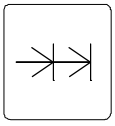
Monochrome Teach Pendant/*i*Pendant Key	Symbolic Teach Pendant Key (if available)	Function
		The HOLD key causes a program to halt.
		The STEP key selects step or continuous test operation.
		The RESET key is used to clear an alarm.

Table 1–6. Editing Keys

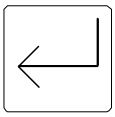
Monochrome Teach Pendant/*i*Pendant Key	Symbolic Teach Pendant Key (if available)	Function
		The ENTER key is used to process and activate the current information set.
		The BACK SPACE key deletes the character or numeral immediately before the cursor.

Table 1–6. Editing Keys (Cont'd)

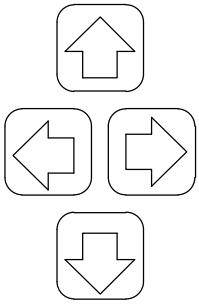
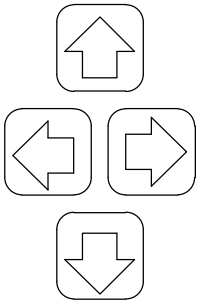
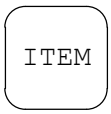
Monochrome Teach Pendant/*i*Pendant Key	Symbolic Teach Pendant Key (if available)	Function
		The arrow keys are used to highlight or select an item on the screen.
		The ITEM key moves the cursor to a line whose number is specified.

Table 1–7. HandlingTool-Specific Keys

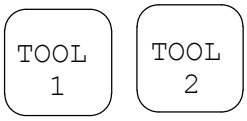
Monochrome Teach Pendant/*i*Pendant Key	Function
	The TOOL 1 or 2 key displays the tool 1 or 2 screen.
	The MOVE MENU key is not supported for HandlingTool.
	The SET UP key displays the SETUP screen.

Table 1–7. HandlingTool-Specific Keys (Cont'd)

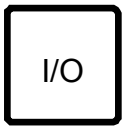
Monochrome Teach Pendant/*i*Pendant Key	Function
	The STATUS key displays the STATUS screen.
	The POSN key displays position data.
	The I/O key displays the I/O screen.

Table 1–8. Help and Diagnostic Key (*i*Pendant only)

Monochrome Teach Pendant/*i*Pendant Key	Symbolic Teach Pendant Key (if available)	Function
		The HELP key displays help files that are available that describe the functions available for the active window. The SHIFT and DIAG keys together display diagnostic information for the currently active error or the selected error in the Alarm menu. This key is only available on the *i*Pendant.

Table 1–9. FCTN Menu

Item	Function
ABORT ALL	This item aborts all running or paused programs.
Disable FWD/BWD	This item allows you to disable the ability to execute program instructions when the SHIFT and FWD keys or SHIFT and BWD keys are pressed. The ability to use SHIFT FWD and SHIFT BWD will be disabled until you press FCTN and select DISABLE FWD/BWD again.

Table 1–9. FCTN Menu (Cont'd)

Item	Function
RELEASE WAIT	During program execution, wait release allows you to override pauses in the program when the robot is waiting for I/O conditions to be satisfied. Release wait works only when a program is running.
QUICK/FULL MENU	This item toggles the menu display between a full or a shorter menu list.
SAVE	This item saves the current program.
PRINT SCREEN	This item saves the current teach pendant screen image to the default device as tpscrn.ls.
PRINT	This item saves the current program to the default device as *program_name*.ls.
UNSIM ALL I/O	This item unsimulates all simulated signals.
CYCLE POWER	This option is available on the FCTN menu after a Cold start. This item toggles the power relay via external I/O. The teach pendant must be enabled for this to work. If the teach pendant is enabled, a prompt box will be displayed asking if you are sure. This is not available on an external robot connection — only from the teach pendant.
ENABLE/DISABLE HMI MENUS	This item enables (or disables) the optional HMI Menus in both FULL and QUICK Menu mode.
REFRESH PANE	This item refreshes the page in the active window.

1.4.1.6 LEDs

The *i*Pendant with a USB Port has the following LEDs.

Table 1–10.

LED	Description
FAULT	This LED indicates that a fault condition has occurred.
POWER	This item indicates controller power is on.

1.4.2 Additional *i*Pendant Features

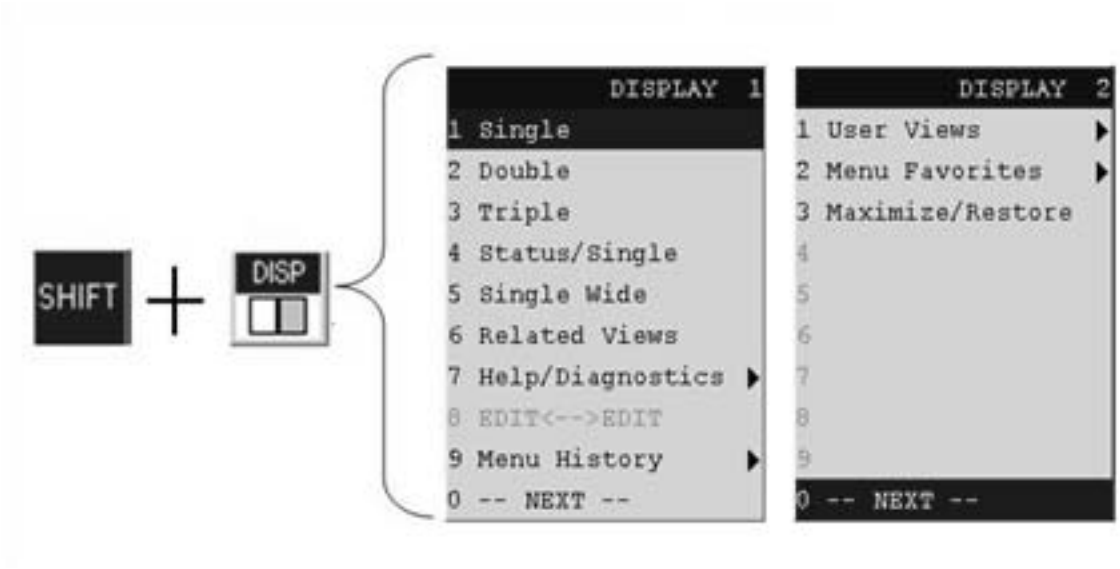
1.4.2.1 Windows

Information can be displayed in multiple windows. You can activate the Display Control menu to select the window modes by pressing the SHIFT key and the DISP key (the Display key) at the same time.

To select a window mode, you can either use the arrow keys to highlight a mode and then press ENTER, or you can type the number of the mode you want.

You will see a menu displayed as shown in Figure 1–18 .

Figure 1–18. Window Display Control Menu



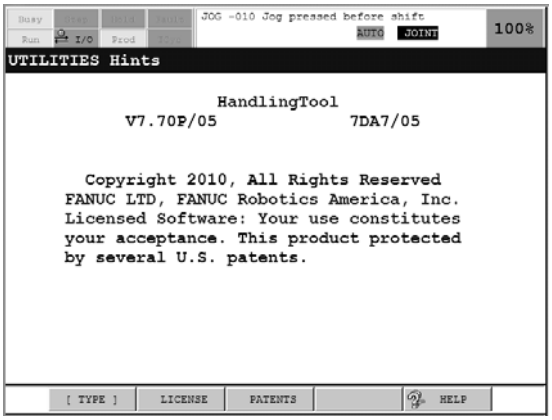
The following window modes are available.

- Single window
- Double window
- Triple window
- Status/Single window
- Single Wide window

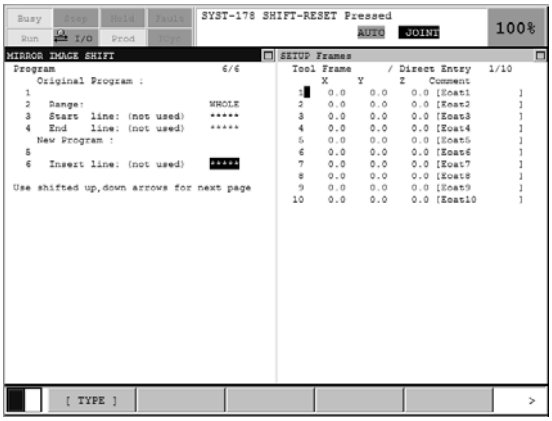
See Figure 1–19 .

Figure 1–19. Multiple Window Display

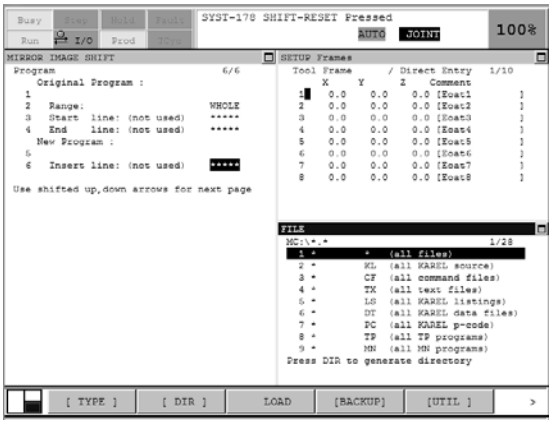
Status/Single Window



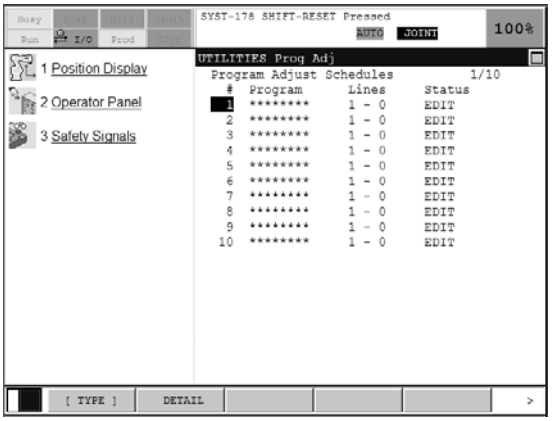
Double Window



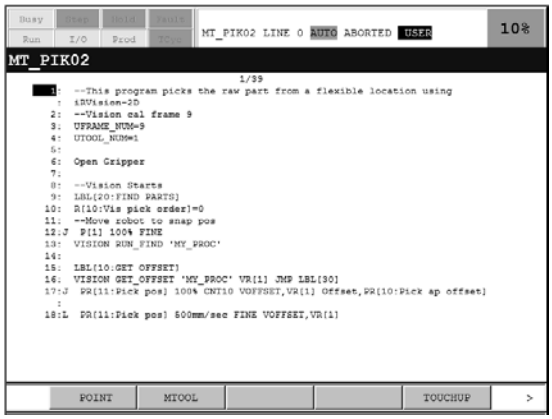
Triple Window



Status/Single Window



Single Wide Window



The following rules apply to multiple window display modes:

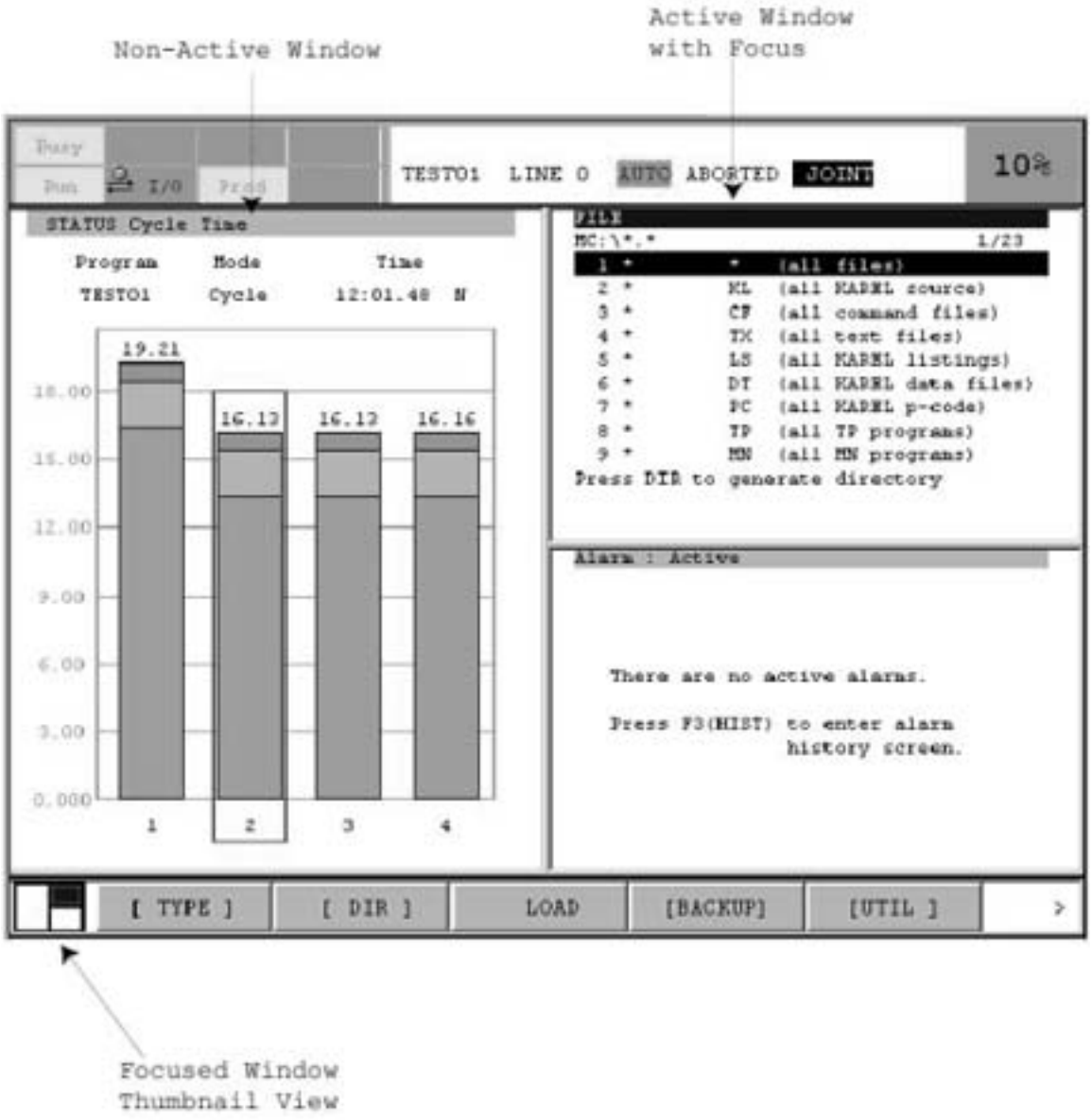
- Switching between Single window display mode and Status/Single window display mode does not affect the information in the right-hand window.
- When you switch from Double to Single window mode, the information in the left window is displayed in the Single window.
- Single and Status/Single window modes allow for the teach pendant editor to function exactly as it does on the monochrome teach pendant.
- In double and triple window modes, the editor is provided in the left-hand pane only.

1.4.2.2 Changing Focus

Focus identifies the currently active or primary window. The current focus can be one of the multiple windows. Only one window has focus at a time. The window with the current focus is identified with a red border and a blue title bar at the top of the window. All other displayed windows will have a grey title bar when they are not the current focus. The displayed function key labels reflect the currently focused window.

See Figure 1–20 .

Figure 1–20. Window and Focus Example



To change the focus automatically, press DISP without pressing the SHIFT key. After you have changed the focus, you can then perform the functions that are allowed for the currently active window.

Note Pressing the EDIT key in double or triple window modes will cause the editor to be displayed in the left hand window regardless of the focus. Also, if one of the windows in multi-window mode is a browser window, the Title bar is not changed when the focus changes.

1.4.2.3 Status/Single Window

The left hand side of the Status/Single window always displays a list of status information that you can select to display. It allows you to display one of the following status modes:

- Position Display
- Operator Panel
- Safety Signals

Note Additional status modes might be available based on your application.

Note The information displayed in the left hand side of the Status/Single Window is **always** Status.

While in any browser screen, the following *i*Pendant keys can be used to navigate the displayed web page.

Table 1–11. Navigating Web Pages

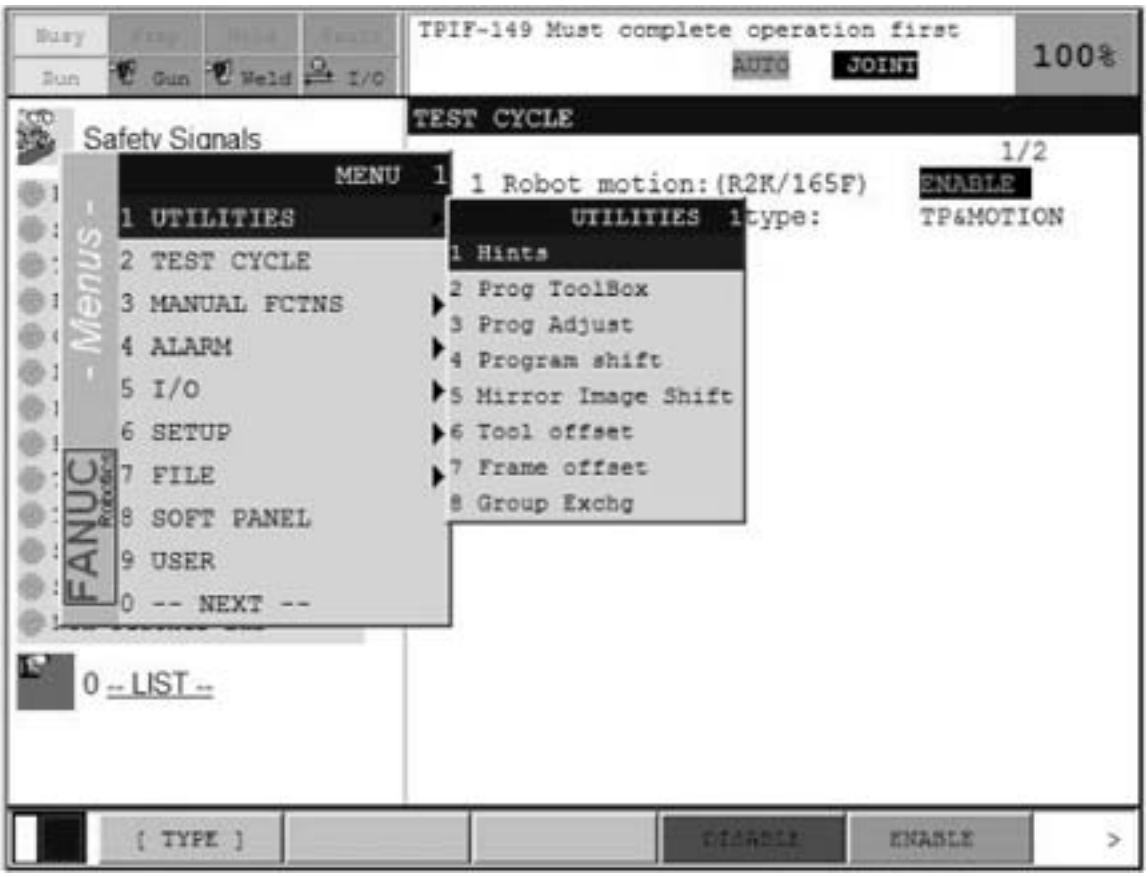
KEY	DESCRIPTION
Up, Down, Left and Right Arrow Key	Move to the next hyperlink in that direction on the page.
SHIFT + Up, SHIFT + Down, SHIFT + Left, SHIFT + Right	Scroll page in that direction. Scroll bars will be visible if more content is available in the horizontal or vertical direction.
ENTER	Select the link that is currently highlighted.

1.4.2.4 Popup Menus

To display the popup menus in the currently active window, press the MENUS key. This will display the first level popup menu.

See Figure 1–21 .

Figure 1–21. First Level and Fly Out Popup Menus



Use the up, down, left, and right arrow keys to select the desired menu item. When a second menu exists for the selected item, use the right arrow key to move to the fly out menu. Then use the up and down arrow keys to select an item on that menu. To display the menu item that you have selected, press ENTER.

Note You can also type the number of the item you have selected and press ENTER to display that menu item.

1.4.2.5 Multi-Pane Edit

A teach pendant program must be moved to the left-hand window for FWD/BWD to work. The DISPLAY menu will contain an EDIT<->EDIT entry. While in the left-hand window, the entry is only enabled if two windows are active. The entry is always enabled while in a right-hand window. The entry is available even if you are not in the EDIT screen. When pressed, the programs are swapped and the EDIT screens are automatically entered.

1.4.2.6 Status Bar Displays

The status bar displays dynamic controller and robot status information. and Table 1–12 describes each kind of status.

Figure 1–22 displays the following status information.

Figure 1–22. Status Bar Display

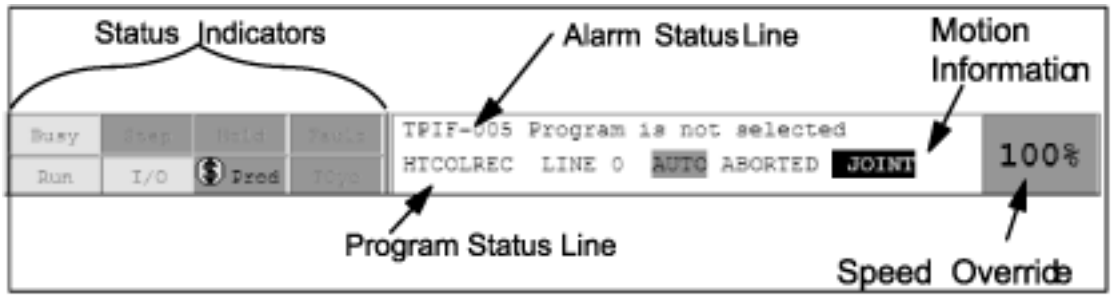


Table 1–12. Status Bar Display Items

STATUS ITEM	DESCRIPTION
Status Indicators	The teach pendant status indicators indicate the system condition when you are using the teach pendant to control the system. • Busy indicates that the controller is processing information. • Step indicates that the robot is in step mode. • HOLD indicates that the robot is in a hold condition. HOLD is not on continuously during a hold condition. • FAULT indicates that a fault condition has occurred. • Run indicates that a program is being executed. • The last three indicators are application-specific
Alarm Status Line	This item displays the current alarm.
Program Status Line	This item indicates the name of the currently executing program, and the current line number and status. It also indicates the mode AUTO, T1, or T2.

Table 1–12. Status Bar Display Items (Cont'd)

STATUS ITEM	DESCRIPTION
Motion Information	This item lists the following information: • Coordinate System • Group • Subgroup
Speed Override	The *speed override* is a percentage of the maximum speed at which the robot will move. A speed override of 100% indicates that the robot will move with the maximum possible speed.

1.4.2.7 Touch Screen Navigation

If you have the optional touch screen available on your *i*Pendant, you can select items by touching them on the screen. The item is activated as soon as you release the touch. If you press one of the following keys, you can select and activate a displayed item by touching and releasing that item on the screen:

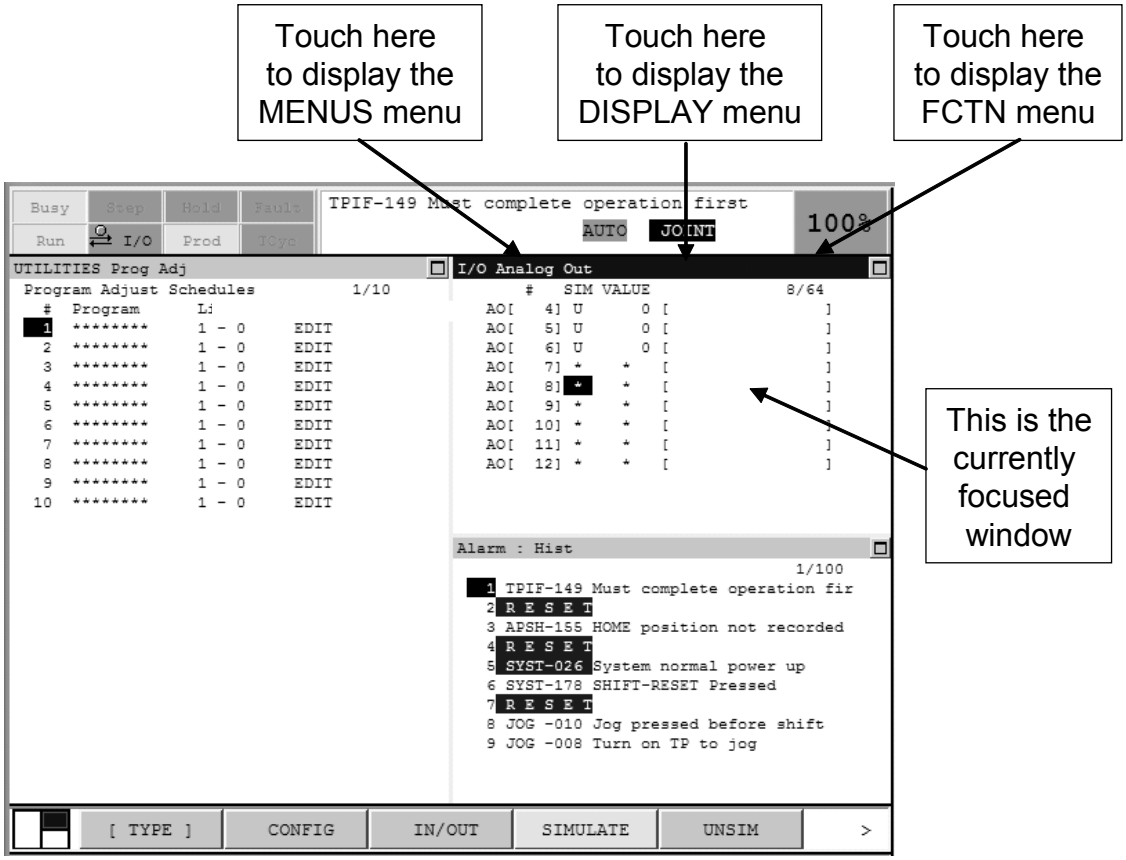
- MENUS
- DISPLAY
- FUNCTION
- [TYPE]
- [CHOICE]

Note Prompt box and Warn windows will not allow touch and release events. These events are typically shown in yellow on the *i*Pendant screen.

You can touch the screen anywhere to focus the window. The title bar of the focused window is divided into three parts and can be touched to activate a shortcut to the MENUS, DISPLAY, and FCTN menus.

See Figure 1–23 .

Figure 1–23. Touch Screen Navigation



The TP Editor will support touch events to select a line or an item in a line. Touching above the first line will scroll up and touching below the last line will scroll down. Most screens will support touch events.

1.4.2.8 Web Page Navigation

While in any browser screen, the following *i*Pendant keys can be used to navigate the displayed web page. Refer to Table 1–13 .

Table 1–13. Navigating Web Pages

KEY	DESCRIPTION
Up, Down, Left and Right Arrow Key	Move to the next hyperlink in that direction on the page.
SHIFT + Up, SHIFT + Down, SHIFT + Left, SHIFT + Right	Scroll page in that direction. Scroll bars will be visible if more content is available in the horizontal or vertical direction.
ENTER	Select the link that is currently highlighted.

1.4.2.9 Help and Diagnostics

You can use the HELP/DIAG key to display help and diagnostics for the data displayed in the current window on the *i*Pendant. You can also select the HELP/DIAGNOSTICS menu item on the DISPLAY popup menu.

To display help information for the currently active window, press the HELP/DIAG key. To display diagnostic information, press the SHIFT key and the HELP/DIAG key.

To use the HELP/DIAGNOSTICS menu item, refer to Table 1–14 . When help is displayed, you can perform the operations as shown in Table 1–15 .

Note If help is already displayed when you select another help function, the previous menu will be displayed. You must select help again to display the help menu.

Table 1–14. HELP/DIAGNOSTICS Menu Items

ITEM	DESCRIPTION
Help	This item displays help for the currently active window.
Context Help	This item displays context sensitive help, if it exists, for the currently active window.
Diagnostics/Diagnostics Home	These items display the Diagnostic Resource Center (DRC) Home page.
Menu Help	This item displays the menu tree and provides links to all the help files menus other than the current menu.

Table 1–15. Help and Diagnostics Operations

Key	Description
Up, Down, Left and Right Arrow Keys	Move to next hyperlink on the page in the direction you select.
SHIFT + Up, SHIFT + Down, SHIFT + Left, SHIFT + Right	Scroll page in that direction (Scroll bars will be visible if more content is available in the Horizontal or Vertical direction).
ENTER	Select the currently highlighted link.

Note The Help and Diagnostics information display location depends on the current window mode and the active window. Table 1–16 lists the display locations for each available window mode.

Table 1–16. Help/Diag Information Display Locations

Window Mode	Display Location
Single	The Help/Diag information replaces the current information in the window.
Status/Single	The Help/Diag information replaces the current information in window on the right.
Double	The Help/Diag information replaces the information in the inactive window (Window without focus) and makes HELP/DIAG window the active window.
Triple	The Help/Diag information replaces the current information in the window without focus and makes the HELP/DIAG window the active window. For example in triple mode, the left window is window 1, the upper right window is window 2, and the lower right window is window 3. Then, if the active window is window 2, the Help/Diag information replaces the contents of window 3. If the active window is window 3, the Help/Diag information replaces the contents of Window 1, and so forth.

Note To display the previous screen after Help has been displayed , press PREV.

To display Help information for any screen,

- Press the HELP key while in that screen. To return to the previous screen, press the HELP/DIAG key or press SHIFT and the HELP/DIAG key at the same time.
- Press SHIFT, and DISP to display the DISPLAY popup menu. Select Help/Diagnostics. Move the cursor to the right and select Help. Help for the current menu will be displayed.

To display context sensitive help if it exists for a screen ,

- Press the HELP key while in that screen. To return to the previous screen, press the HELP/DIAG key or press SHIFT and the HELP/DIAG key at the same time.

- Press SHIFT, and DISP to display the DISPLAY popup menu. Select Help/Diagnostics. Move the cursor to the right and select Context Help. Context sensitive help for the current menu will be displayed.

To display diagnostics information ,

- Press the SHIFT key and the HELP/DIAG key. A popup screen, similar to the one shown below, will be displayed. This screen contains the program status, the most recent alarm, and the cause code if one exists.



Cancel will return to the previous screen. Continue will display the diagnostic information for that specific alarm. If there is no active alarm, then the Diagnostic Resource Center will be displayed allowing you to go to the diagnostic information for any system error.

- Press SHIFT, and DISP to display the DISPLAY popup menu. Select Help/Diagnostics. Move the cursor to the right and select Diagnostics or Diagnostics Home.

In addition, if you are in an ALARM screen and press the SHIFT + HELP/DIAG key you will get the diagnostic information for the alarm that you have selected. To exit the diagnostic screen and return to your previous screen, press either the HELP/DIAG key or press SHIFT and the HELP/DIAG key at the same time. You can also press PREV to display the previous screen.

To display Menu Help ,

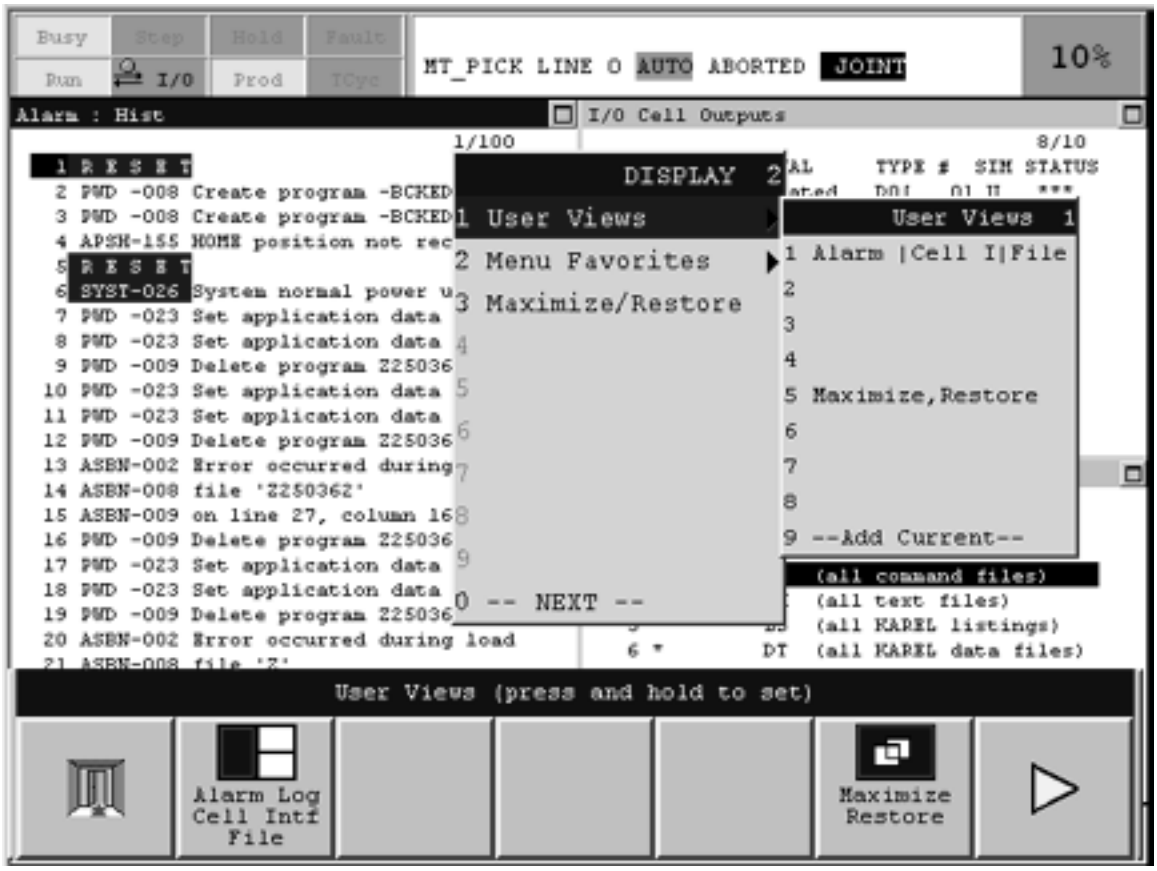
- Press the SHIFT key and the DISP key, select Help/Diagnostics and select Menu Help. This displays the menu tree and provides links to all the help files menus other than the current menu.

1.4.2.10 User Views

User views can be used to store up to ten commonly used user-defined single or multi-window displays. For example, if you commonly use the triple pane window and have the ALARM menu,

Cell I/O menu, and the FILE menu displayed, then you would want to define this set of menus as a user view. After you store a user view, it is named and listed on the User Views menu so that you can select it for redisplay later. The previous example user view would be listed on the User Views menu as Alarm|Cell I|File. In addition, the User Views are displayed on a dialog box at the bottom of the screen with a thumbnail view of the configuration. See Figure 1–24 .

Figure 1–24. User Views



Use Procedure 1-1 to add a menu set as a User View. Use Procedure 1-2 to modify the list of User Views.

Procedure 1-1 Adding a User View

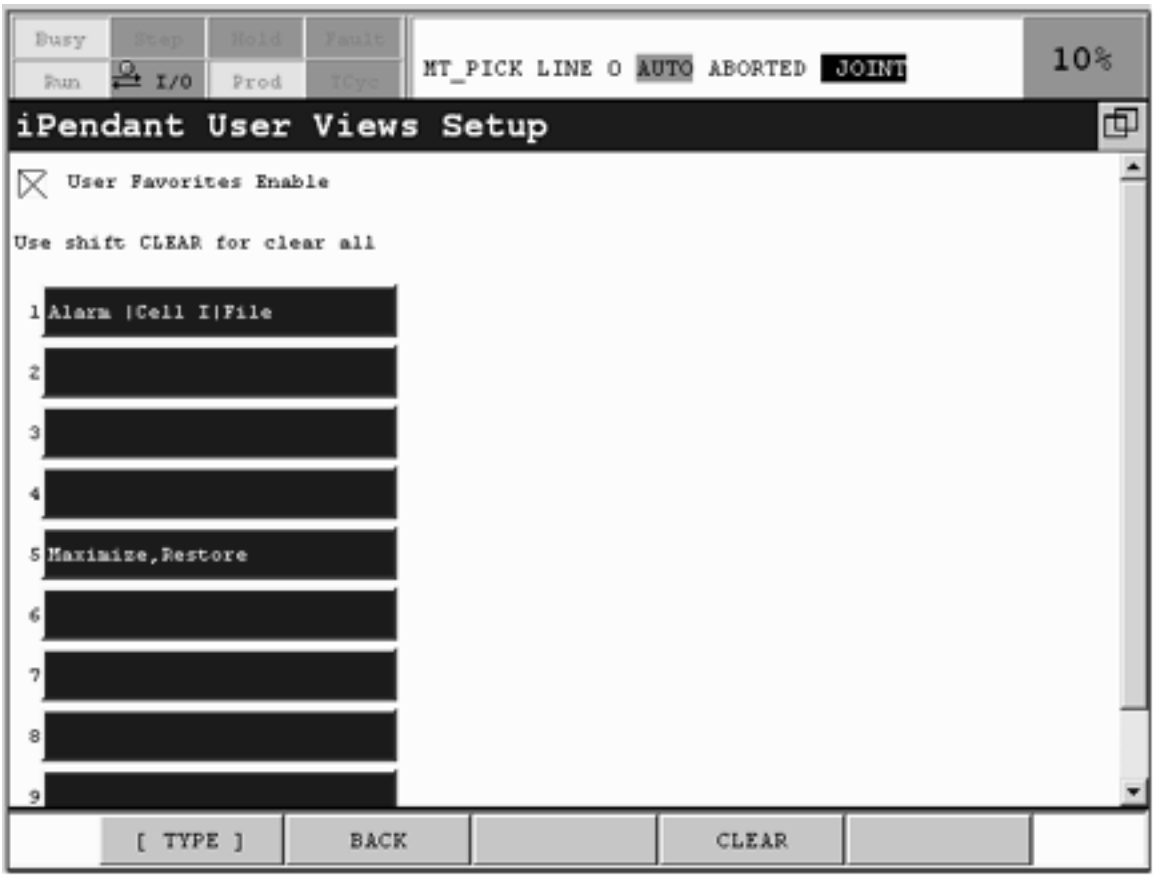
1. Display the set of menus on the *i*Pendant that you want to add as a view.
2. Press SHIFT and then press the DISP key.
3. If the User Views dialog box is shown at the bottom of the screen, you may press and hold the button or function key which you want to set for 3 seconds. The currently displayed menu set will be added to the button. You may overwrite existing buttons. In addition, you may press the right arrow button or next key to display and set the next 5 buttons. The left-most exit button or prev key will cancel the Display Menu.

4. If the User Views dialog box is not shown at the bottom of the screen, select User Views in the Display Menu. Move the arrow keys to Add current. The currently displayed menu set will be added to the list as Menu (if it is a Single Pane display), Menu|Menu (if it is a Double Pane display), and Menu|Menu|Menu (if it is a Triple Pane display).
5. Repeat Step 1 through Step 4 for each User View you want to add.

Procedure 1-2 Modifying User Views

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select iPendant Setup. You will see the iPendant General Setup Screen.
5. Select User Views Setup. You will see a screen similar to the following.

Figure 1–25. Modifying User Views Screen



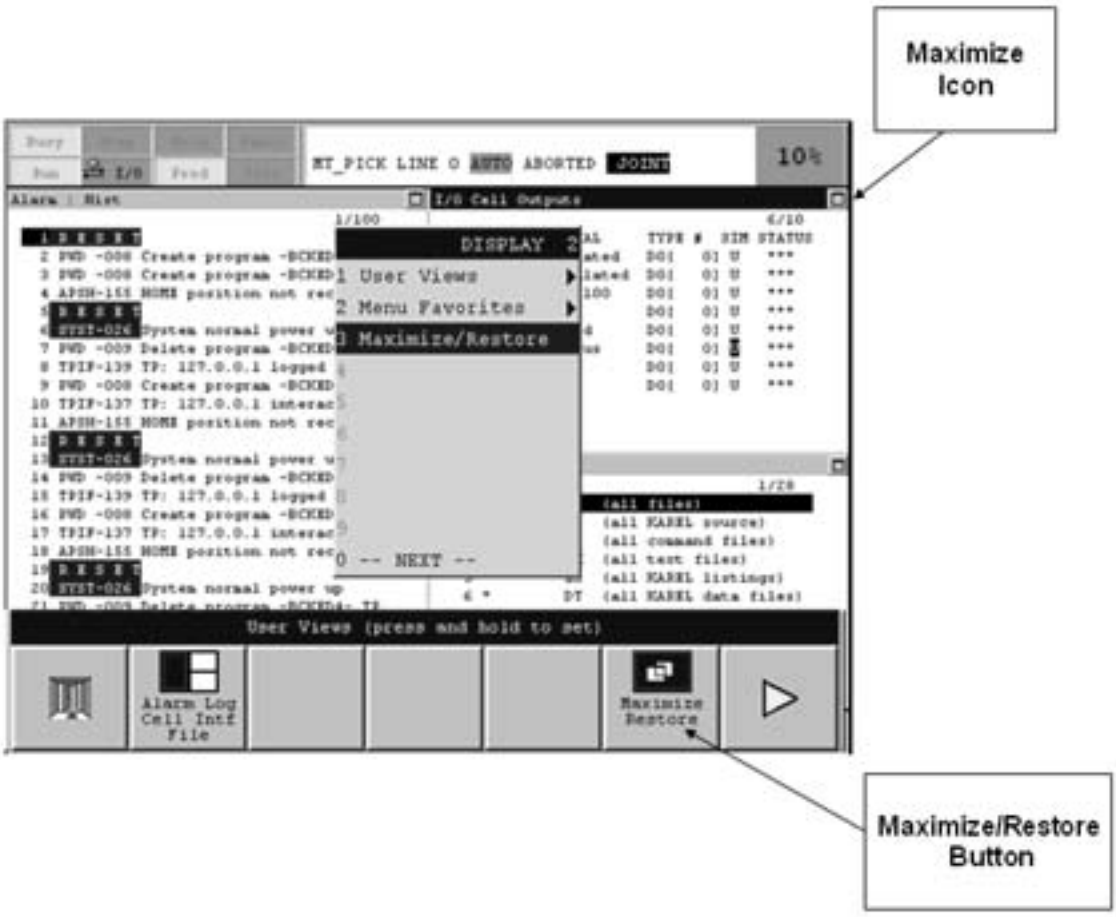
Note User Views, after they have been cleared, cannot be recovered.

6. **To clear one entry from the User View list** , move the cursor to the view you want to clear, and press F4, CLEAR. The User View list will be redisplayed automatically.
7. **To clear ALL entries from the User View list** , press SHIFT and F4, CLEAR. The User View list will be cleared automatically.
8. **To disable the User Views dialog box** which is displayed by the DISPLAY menu, clear the User Favorites checkbox.
9. **To enable the User Views dialog box** which is displayed by the DISPLAY menu, check the User Favorites checkbox.
10. **To display the main iPendant Setup menu** , press F2, BACK.

1.4.2.11 Maximize/Restore

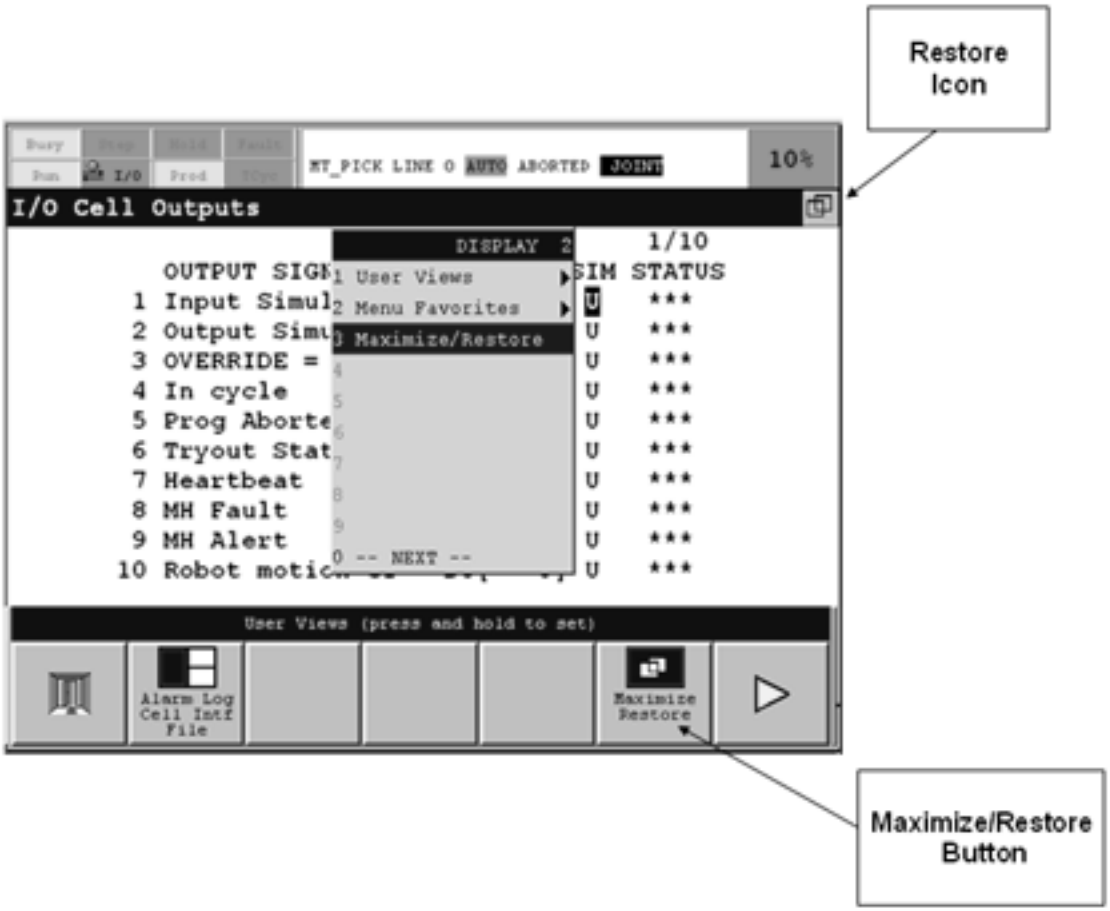
Maximize/Restore is the ability to ZOOM the current window. You either touch the Maximize icon or press the Maximize/Restore button or function key from User View dialog box or select the Maximize/Restore entry in the DISPLAY menu. The configuration changes to Single Window with the currently selected window. See Figure 1–26 .

Figure 1–26. Maximize Window



You can revert back to original display by touching the Restore icon or selecting Maximize/Restore. See Figure 1–27 . If you change configuration while the window is Maximized, then the Restore is cancelled.

Figure 1–27. Restore Window



1.4.2.12 Menu Favorites

Menu Favorites allows you to generate and store a list of menus that you display most often. There are 2 ways to add a menu favorite.

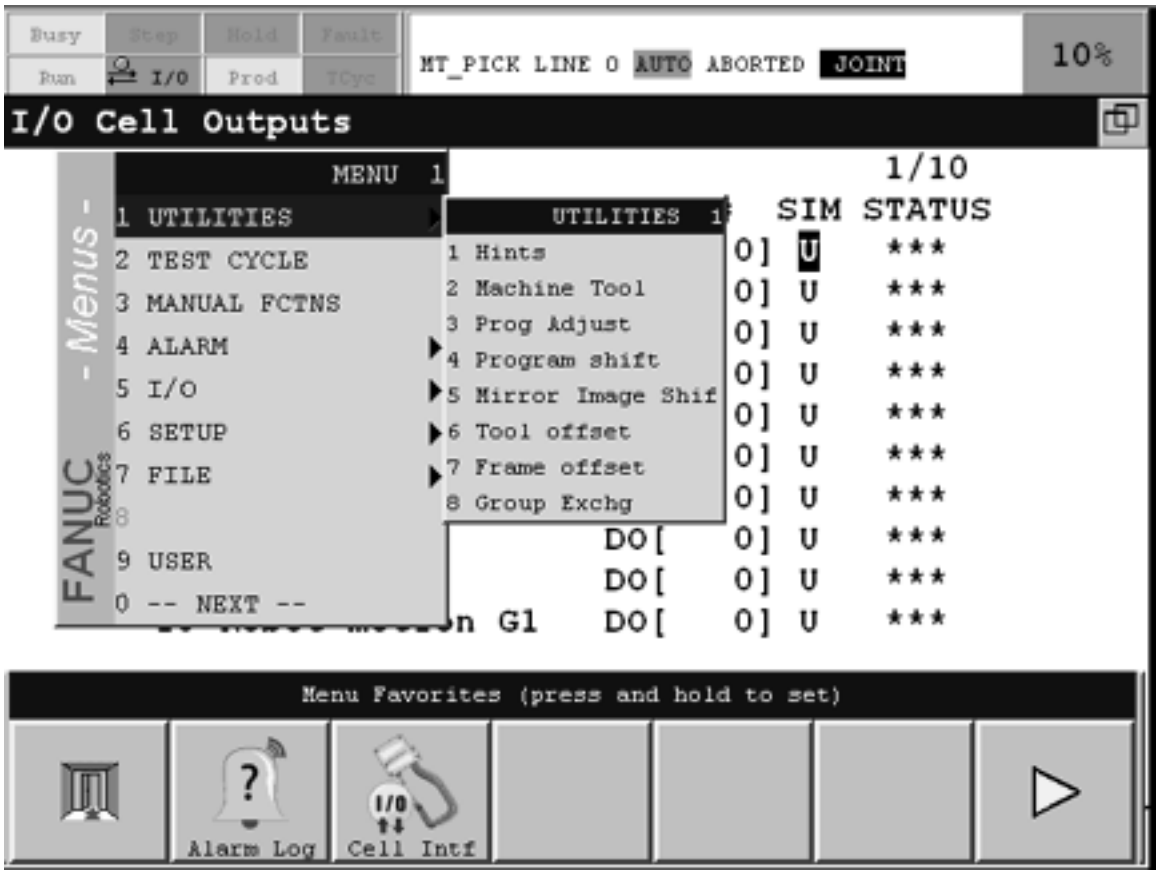
Use Procedure 1-3 to set up your Menu Favorites using the MENUS Dialog box. Use Procedure 1-4 to set up your Menu Favorites list using the DISPLAY Menu. Use Procedure 1-5 to modify the Menu Favorites list.

Procedure 1-3 Adding Menus to the Menu Favorites using MENUS Dialog Box

1. Display the menu that you want to add as a favorite menu.
2. Press MENUS key.
3. If the dialog box is shown at the bottom of the screen, you may press and hold the button or function key which you want to set for 3 seconds. The currently displayed menu will be added to

the button with an icon. You may overwrite existing buttons. In addition, you may press the right arrow button or next key to display and set the next 5 buttons. The left-most exit button or prev key will cancel the Dialog Box and the Menus popup. See Figure 1–28 for an example.

Figure 1–28. Menu Favorites



4. **To display a menu stored in the Menu Favorites** , press MENUS and choose the menu by pressing the button or function key.

Procedure 1-4 Adding Menus to the Menu Favorites List using DISPLAY Menu

1. Display the menu that you want to add as a favorite menu.
2. Press SHIFT and then press the DISP key.
3. Select Menu Favorites.
4. Move the arrow keys to Add current. The currently displayed menu will then be added to the list.

   See Figure 1–29 for an example.

Figure 1–29. Menu Favorites



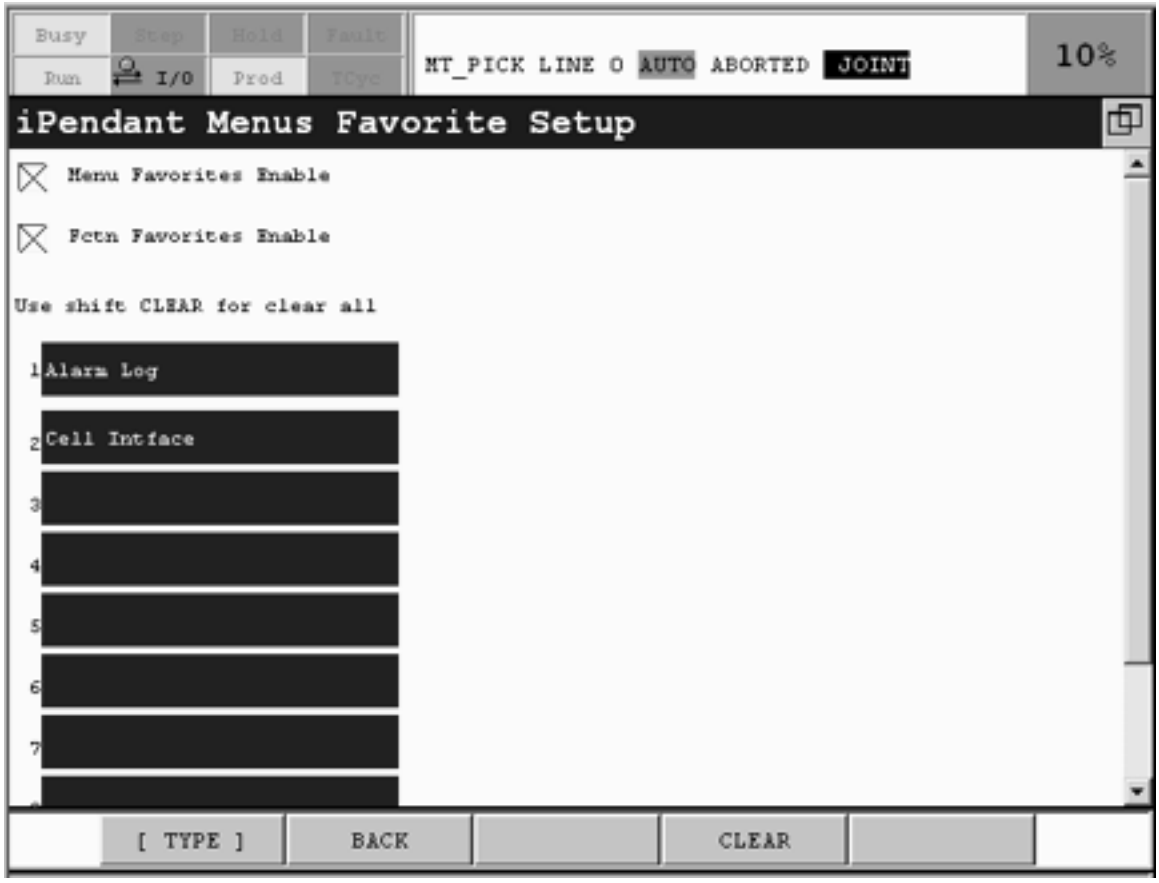
5. **To display a menu stored in the Menu Favorites list** , press SHIFT and then the DISP key. Move the cursor to Menu Favorites and choose the menu from the list.

Procedure 1-5 Modifying Menu Favorites

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select iPendant Setup. You will see the iPendant General Setup Screen.
5. Select Menu Favorites Setup.

   You will see a screen similar to the following.

Figure 1–30. Modifying the Menu Favorites Screen



Note Menu Favorites, after they have been cleared, cannot be recovered.

6. **To clear one entry from the Menu Favorites list**, move the cursor to the view you want to clear, and press F4, CLEAR. The Menu Favorites list will be redisplayed automatically.
7. **To clear ALL entries from the Menu Favorites list**, press SHIFT and F4, CLEAR. The Menu Favorites list will be cleared automatically.
8. **To disable the Menus Favorites dialog box** which is displayed by the MENUS popup, clear the Menu Favorites checkbox.
9. **To enable the Menus Favorites dialog box** which is displayed by the MENUS popup, check the Menu Favorites checkbox.
10. **To display the main iPendant Setup menu** , press F2, BACK.

1.4.2.13 Menu History

The Menu History keeps track of the last eight menus that have been displayed. This list is generated automatically and cannot be modified. You can, however, select a name on the list and that screen will be displayed.

See Figure 1–31 .

Figure 1–31. Menu History Menu List



1.4.2.14 Display Equip

This item is only used in applications that are controlling multiple equipments. Some menus in these applications are equipment related and change based on which equipment is chosen. The Display Equip item is used to select the equipment number being referenced by these menus. Display Equip applies to the menu in the pane that currently has focus. The equipment number selected is displayed in the focus bar for that pane.

If your system uses multiple equipment, refer to the multiple equipment section in this manual for more information.

1.4.2.15 Other *i*Pendant Guidelines

Editing Guidelines

- LOOK/MONITOR mode is available in any window.
- Each window can have a unique default program.
- Selecting a teach pendant program from the SELECT screen in the left-hand window will cause that program to be the default program for that window ($TP_DEFPROG).
- Selecting a teach pendant program from the SELECT screen in the right-hand window will cause that program to be the default program for that window ($UI_DEFPROG[1]).
- Selecting a teach pendant program from the SELECT screen in the lower right-hand window will cause that program to be the default program for that window ($UI_DEFPROG[2]).
- The Status line will always show the default program for the left-hand window even if it does not have focus.
- The Editor title line shows the program that is being edited.
- The program selected will be retained during cycle power. The current line number will only be retained during cycle power for the left-hand window during hot start. The right-hand windows do not retain the current line number.
- The same program can be edited in multiple windows. The cursor is independent. When a change is made in one window, the change is immediately reflected in the other window. Both windows can make changes to the same program. This is useful for copy and paste within the same program.
- The 3 windows will share the same copy and paste buffer. This provides the ability to copy and paste from one TP program to another.
- Each window has its own undo and redo buffers so multiple edit sessions will not affect each other.
- TP FWD/BWD is only supported in the left-hand window. If focus is in a right-hand window, then the warning “TPIF-166 FWD/BWD in left window only” is posted.
- The program must be moved to the left-hand window for TP FWD/BWD to work. The DISPLAY menu will contain EDIT->EDIT entry. While in the left-hand window, the entry is only enabled if 2 windows are active. The entry is always enabled while in a right-hand window. The entry is available even if you are not in the EDIT screen. When pressed, the programs are swapped and the EDIT screens are automatically entered.

Operational Guidelines

- The system variable $UI_CONFIG.$recovermenu is provided to display the menus in the same state they were in when the controller was turned off.

- After the controller is turned off and then on again, the editor configuration should not change if $UI_CONFIG.$recovermenu is set. Refer to Table 1–17 .

Table 1–17. $UI_CONFIG.$recovermenu

Setting	Description
0	At Cold start the system will display the "Hints" screen in single window mode.
1	At Cold start the system will display the last screen displayed before power down in single window mode.
2	At Cold start the system will display the "Hints" screen in all windows in whatever window mode was used at power down.
3	Both 1 and 2.
4	At Hot start the system will display the last screen displayed before power down in single window mode.
6	Both 4 and 2. This is the default setting.

1.4.2.16 Browser Guidelines

You can use the *i*Pendant to browse web pages that are accessible on the network from the robot.

Note In order to access any web page that is not resident on a controller to which the *i*Pendant is attached, the Internet Protocol Connectivity and Customization Option must be loaded and the Proxy Server properly configured. This includes accessing other robots on the network, the FANUC Robotics Diagnostics Resource Center (DRC), and any other intranet or internet site.

Without the Internet Protocol Connectivity and Customization Option, the *i*Pendant can only be used to access web pages resident on the controller to which the *i*Pendant is connected. This includes standard Help and Diagnostic (Cause/Remedy) information, the Home page of the robot and any custom developed .htm pages.



Caution

The browser only supports a limited subset of JavaScript and HTML tags. Browser components like FLASH, JAVA, ActiveX components, and so forth are not supported on the *i*Pendant. Therefore, not all web pages are guaranteed to display correctly.

To display the browser , press MENUS and select BROWSER. This will display the Browser Favorites menu. To access web pages, select Favorites from the Main menu. Table 1–18 lists the operations you can perform using the Favorites Screen function keys. Table 1–19 describes the following Favorites operations:

- Browse the Home Page of the robot to which the *i*Pendant is connected
- Add a hyperlink to another website or robot
- Modify or Delete a Hyperlink that you have created
- Select a Pre-Programmed Hyperlink (using the [TYPE] key)

Table 1–18. Favorites Screen Function Keys

Key	Description
[TYPE]	Use this key to access pre-programmed hyperlinks.
BACK	Use this key to move back to the previously displayed web page.
FORWARD	Use this key to move forward one web page in the history buffer.
REFRESH	Use this key to refresh the currently displayed web page.
HOME	Use this key to move to the Home page of the robot to which the iPendant is connected.
FRAME BACK	Use this key to move back to the previous web page frame.
FRAME FORW	Use this key to move forward to the next web page frame.
STOP	Use this key to stop the download of a web page.
FAVORITES	Use this key to display the Favorites Page or to add or select a link.

Table 1–19. Favorites Screen Operations

Operation	Description
Browse <my_robot> Home Page	This link displays the home page of the robot to which the *i*Pendant is connected. <my_robot> is the hostname of the robot to which the iPendant is connected. This is the robot name entered in the robot controller TCP/IP setup.

Table 1–19. Favorites Screen Operations (Cont'd)

Operation	Description
Add a Link	This allows you to create hyperlinks to any website available on the network with the robot or to link to .htm or .stm files resident on the robot. To access any file or link not resident on the controller the "Internet Protocol Connectivity and Customization Option" must be loaded and the Proxy and DNS services must be properly configured. Adding a link requires that you define two fields. The first is a " friendly name" for the link. This name is used to identify the site to which you are linking. The second field is the network address for the link. It can be either a static IP address, or a DNS name. (For example, http://192.168.1.125, http://robot11.frc.com, and so forth.) To link to a file resident on the robot, use a relative link without the http://. For example, /fr/pw_op1.stm refers to pw_op1.stm on the FR: device. Also, /md/memory.dg refers to the memory diagnostic page on the MD: device. When"Add a Link" is selected, the following screen is displayed.  Select the Enter a Name box using the cursor keys, and press ENTER. The popup keyboard will be displayed. Use the cursor keys to type the < name> (ex. FANUC, Yahoo, and so forth) you want associated with this link. When you are finished typing the name, select EXIT on the popup keyboard. Select the Enter an Address box and use the same procedure as above to type the link address (ex. http://192.168.1.1 or http://robot1.frc.com). **Note** The http:// will be added to the link address automatically.

Table 1–19. Favorites Screen Operations (Cont'd)

Operation	Description
Add a Link Continued	After the link name and address are entered, press Continue to add the entry to the Favorites Screen. See the following screen for an example.

Favorites
Browse <my_robot> Home Page
Add a Link

Browse Link	Modify Link Definition	Delete Link
Browse Robot1(http://192.168.1.1)	Modify Robot1	Delete Robot1
FANUC Robotics (http://www.fanucrobotics.com)	Modify FANUC Robotics	Delete FANUC Robotics
Yahoo (http://www.yahoo.com)	Modify Yahoo	Delete Yahoo

Modify an Entry	To modify an entry, go to the Modify <name> link and press ENTER. This will display a screen similar to the entry screen above. Modify the appropriate items and press CONTINUE to save the modifications.
Delete an Entry	To delete an entry, go to the Delete <name> link and press ENTER. This will immediately remove this link from the list.
Pre-Programmed Links	These links are those that are listed when you select the [TYPE] key while in the Browser. They also show up as a pop-up menu when you select BROWSER from the Main Menu. Up to 10 of these pre-programmed links can be created by setting the following system variables: **$tx_screen[n].$destination = "Your HTML page"** **$tx_screen[n].$screen_name = "Your screen name"** **Where n is 1 - 10** **Example: $tx_screen[1].$destination = "http://www.yahoo.com"** **$tx_screen[1].$screen_name = "yahoo"** In this case the popup [TYPE] menu will have "yahoo" as one of the entries. If it is selected the website, www.yahoo.com, will be displayed on the *i*Pendant.

Note Access to external links requires the Internet Protocol Connectivity and Customization Option.

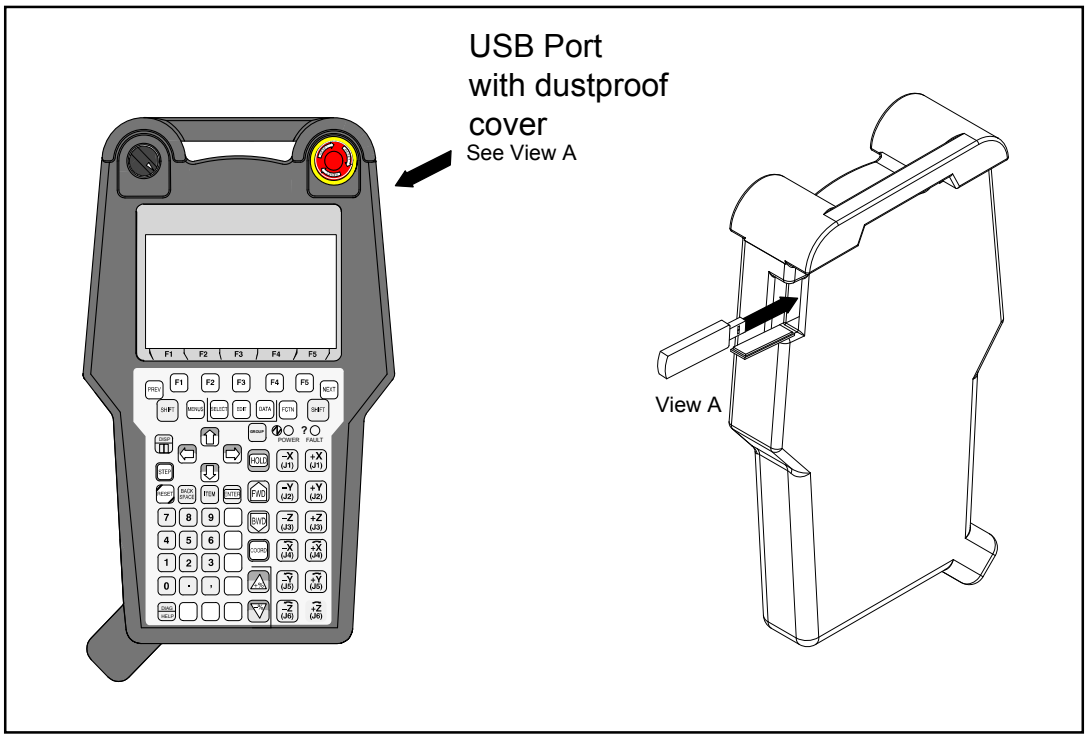
1.4.2.17 Screen Customizations

Refer to the "Advanced *i*Pendant Functions" Appendix in the application-specific *Setup and Operations Manual* for more information.

1.4.2.18 Universal Serial Bus (USB) Port

In addition to a USB port on the controller, a USB port also might exist on the *i*Pendant. You can connect a memory stick to this port or you can connect a camera for the vision sensor. It is also possible to connect it directly to a personal computer. See Figure 1–32

Figure 1–32. *i*Pendant with USB Port



1.5 FANUC ROBOTICS SOFTWARE

1.5.1 FANUC Robotics Software Overview

The FANUC Robotics software works in conjunction with the robot and the controller to allow you to:

- **Set up** information required for the application

- **Program** your application
- **Test** your program
- **Run production**
- **Display** and monitor process information

Other tools such as program and file management capabilities help you to maintain your system before, during, and after the production stage.

Note If you are using the DispenseTool Plug-in, the multiple application option must be installed and you must select the DispenseTool option at controlled start.

1.5.2 Set Up

The FANUC Robotics software provides the components necessary to set up all the information required for your application. It also provides the necessary commands for you to set up how you want your programs to run during production.

Your application software also allows you to work directly with external devices such as cell controllers. You will need to set up your equipment before you begin your application.

1.5.3 Program

An application program is a combination of instructions that, when executed in a sequence, will complete your task. Application programs can be called by several names.

Note If you are using PalletTool, you do not need to write any programs to set up and run PalletTool. The PalletTool menus guide you through everything you need to set up and run most palletizing applications. However, if the requirements of your application are very different from the standard capabilities provided by PalletTool, you can modify one or more PalletTool teach pendant programs.

Refer to the “Program Elements” chapter for more information.

1.5.4 Test

After you have set up the FANUC Robotics software and successfully created a program or modified a preexisting program, you must test your application to be sure it runs correctly. Refer to the “Testing a Program and Running Production” chapter for more information.

Testing the application is an important step in creating a successful application. Be sure to test your program thoroughly before running it in production.

1.5.5 Run Production

Running production is the final step in executing an application program.

It consists of

- Indicating to the controller which programs to run
- Performing recovery and restart procedures
- Adjusting program data
- Running the application program
- Displaying and monitoring process information

Refer to the “Testing a Program and Running Production” chapter for more information.

Chapter 2

TURNING ON AND JOGGING THE ROBOT

Contents

Chapter 2	TURNING ON AND JOGGING THE ROBOT	2–1
2.1	OVERVIEW	2–2
2.2	TURNING ON AND TURNING OFF THE ROBOT	2–2
2.3	JOGGING THE ROBOT	2–5
2.3.1	Overview	2–5
2.3.2	Jog Speed	2–6
2.3.3	Coordinate Systems	2–7
2.3.4	Wrist Jogging	2–10
2.3.5	Remote TCP Jogging	2–11
2.3.6	Motion Groups	2–12
2.3.7	Extended Axes and Sub-Groups	2–13
2.3.8	Jog Menu	2–18

2.1 OVERVIEW

Before you can create a program and run production, you must first know how to turn *on* and *jog* the robot. Turning on the robot provides power to the robot and controller. Jogging is moving the robot by pressing keys on the teach pendant. This chapter contains information and procedures to turn the robot on and off, and to jog the robot.

2.2 TURNING ON AND TURNING OFF THE ROBOT

Turning on the robot provides power to the robot and controller and performs the following actions:

- Initializes changes to system variables
- Initializes changes to I/O setup
- Displays the utilities hints screen (during Cold start only)
- Initializes changes to cell I/O

Note The action list above might vary according to your particular setup and installation.

For listings and descriptions of alternative startup methods, refer to Appendix C .



Warning

Lethal voltage is present in the controller WHENEVER IT IS CONNECTED to a power source. Be extremely careful to avoid electrical shock.

Turning the disconnect or circuit breaker to the OFF position removes power from the output side of the device only. High voltage is always present at the input side whenever the controller is connected to a power source.

Use Procedure 2-1 to turn on the robot. Use Procedure 2-2 to turn off the robot. Use Procedure C-5 to cycle controller power.



Caution

Your plant might require additional inspections before turning on power to the robot. To help ensure safe operation, you should familiarize yourself with the guidelines for your particular installation before you turn on the robot.

Procedure 2-1 Turning On the Robot

Conditions

- All personnel and unnecessary equipment are out of the workcell.

Steps

1. Visually inspect the robot, controller, workcell, and the surrounding area. During the inspection make sure all safeguards are in place and the work envelope is clear of personnel.
2. Turn the power disconnect circuit breaker on the operator panel to ON.
3. **If you are using a paint application** (PaintTool), complete the following steps. Otherwise, go to Step 4 .

   Note The controller will not turn on if the purge cycle has not been completed. If a purge fault exists, the Purge Complete light will be off and the Purge Fault light will be on.

   a. **If the Purge Fault light is on,** press and hold the Purge Enable button on the controller operator panel to initiate the purge cycle. Wait for the Purge Complete LED on the controller operator panel to light. This process takes five minutes from the time you press Purge Enable.

   b. Check the following conditions and correct them if they exist:

   - Any robot covers have been removed.
   - Plant air pressure has failed.
   - Robot air line has been removed.
   - Purge solenoids have failed.

   Warning

   DO NOT turn on the robot if you discover any problems or potential hazards. Report them immediately. Turning on a robot that does not pass inspection could result in serious injury.

4. When the controller has turned on, you will see a teach pendant screen similar to the following.

```
UTILITIES Hints              JOINT 100%
             ApplicationTool
        Vx.xxx            XXXX/XX
              Custom Vx.xxx
  Copyright xxxx, All Rights Reserved
FANUC LTD, FANUC Robotics America, Inc.
Licensed Software: Your use constitutes
your acceptance. This product protected
by several U.S. patents.
```

Table 2–1. Utilities Hints Screen Items

Line	Description
Line 1: Application Tool	This item lists the Application Tool software that is loaded
Line 2: Version Number	This item is the version number of the software. This first item shows the most important version number for Robot Controllers located in North America. The second item shows the most important version number for Robot Controllers located in Asia.
Line 3: Custom	The third line is either all blank or it shows the version number of a Customization. It will show all blank if there is no Customization installed or the if Customization does not set this Version.

Procedure 2-2 Turning Off the Robot

Steps

1. **If a program is running or if the robot is moving,** press the HOLD key on the teach pendant.
2. Perform any shutdown procedures specific to your installation.

   Note To cycle the controller off and then on again after a Cold start, press FCTN and select CYCLE POWER. This will turn the controller off and then back on

3. Turn the disconnect circuit breaker to OFF.

Warning

Lethal voltage is present in the controller WHENEVER IT IS CONNECTED to a power source. Be extremely careful to avoid electrical shock.

Turning the power disconnect circuit breaker to the OFF position removes power from the output side of the device only. High voltage is always present at the input side whenever the controller is connected to a power source.

Note If you are using a paint application when the power circuit breaker is turned off, a purge of the robot cavity will be required when the controller is turned back on.

Procedure 2-3 Cycling Controller Power

Conditions

- The teach pendant is enabled.
- You are not using an external robot connection. This is only available on the teach pendant.
- The controller is currently in a Cold start state.

1. Press FCTN.
2. Select CYCLE POWER.
3. Press ENTER. You will see a screen similar to the following.

```
This will cycle power.
Are you sure?
     [NO]      YES
```

4. Use the teach pendant arrow keys to select YES, and press ENTER.

2.3 JOGGING THE ROBOT

2.3.1 Overview

Jogging is moving the robot axes by pressing keys on the teach pendant. Before you add a motion instruction to a teach pendant program you must first jog the robot to the position you want.

The following items affect the way the robot jogs and the axes that move while jogging:

- **Jog speed** - How fast the robot moves when jogging
- **Coordinate system** - The way the robot moves when jogging
- **Minor axis wrist jogging** - How the wrist axes will jog
- **Remote TCP jogging** - Whether the tool is fixed in the workcell
- **PATH jogging** - Jog coordinate system corresponds to current path

The following items affect the axes that move while jogging:

- **Motion Groups** - Which motion group is selected
- **Extended axes and motion sub-groups** - Which extended axes or sub-group that is selected

Note Some items listed above might not be supported by your software application.

2.3.2 Jog Speed

The *jog speed* is a percentage of the maximum speed at which you can jog the robot. The current jog speed is displayed in the top right corner of every teach pendant screen.

A jog speed of 100% indicates that the robot will move with the maximum possible jog speed. The maximum possible jog speed varies depending on the robot model. The maximum possible jog speed is defined by the tool center point (TCP) moving at and below 250 millimeters per second. A jog speed of FINE or VFINE indicates that the robot will move in incremental steps. Table 2–2 lists all the possible values of the jog speed.

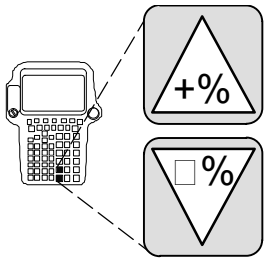
Note When you use FINE and VFINE speed values, the robot moves one step at a time. You must release the jog key and press it again to move the robot again.

Table 2–2. Jog Speed Values

Speed Values	Joint	Cartesian
100, 95, 90, 85, ... 20, 15, 10, 5, 4, 3, 2, 1	% of jog speed	% of jog speed
FINE (incremental steps)	Approximately 0.001 degrees	Approximately 0.023 mm
VFINE (incremental steps)	Approximately 0.0001 degrees	Approximately 0.002 mm

The jog speed keys on the teach pendant are used to increment or decrement the jog speed. The SHIFT key combined with a jog speed key causes the jog speed to be changed between 100, 50, 5, FINE, and VFINE. Figure 2–1 shows the jog speed keys.

Figure 2–1. Jog Speed Keys



Set the jog speed to a value that is appropriate for the conditions in the workcell, the kind of jogging the robot is doing, and your own experience in jogging a robot. Use a slow jog speed until you are familiar with the robot. The slower the jog speed, the more control you have over robot motion.

Note The jog speed increments only when the COORD/JOG SPEED screen on the teach pendant is displayed.

- Press the COORD or +% or -% jog speed key to display the COORD/JOG SPEED screen.
- Press the +% or -% jog speed key again to change the jog speed values.

2.3.3 Coordinate Systems

In jogging, a *coordinate system* defines how the robot will move. There are five coordinate systems:

- JOINT
- XYZ - includes WORLD, JGFRM, and USER
- TOOL
- PATH - (ArcTool only).
- LDR 2 - refer to the *Coordinated Motion Manual*

You change the coordinate system by pressing the COORD key on the teach pendant, shown in Figure 2–2 . The coordinate system you choose is displayed in the upper right hand corner of the teach pendant screen, and on the teach pendant indicators. Table 2–3 indicates the LED or LCD indicator that corresponds to the coordinate system you choose.

Figure 2–2. COORD Display

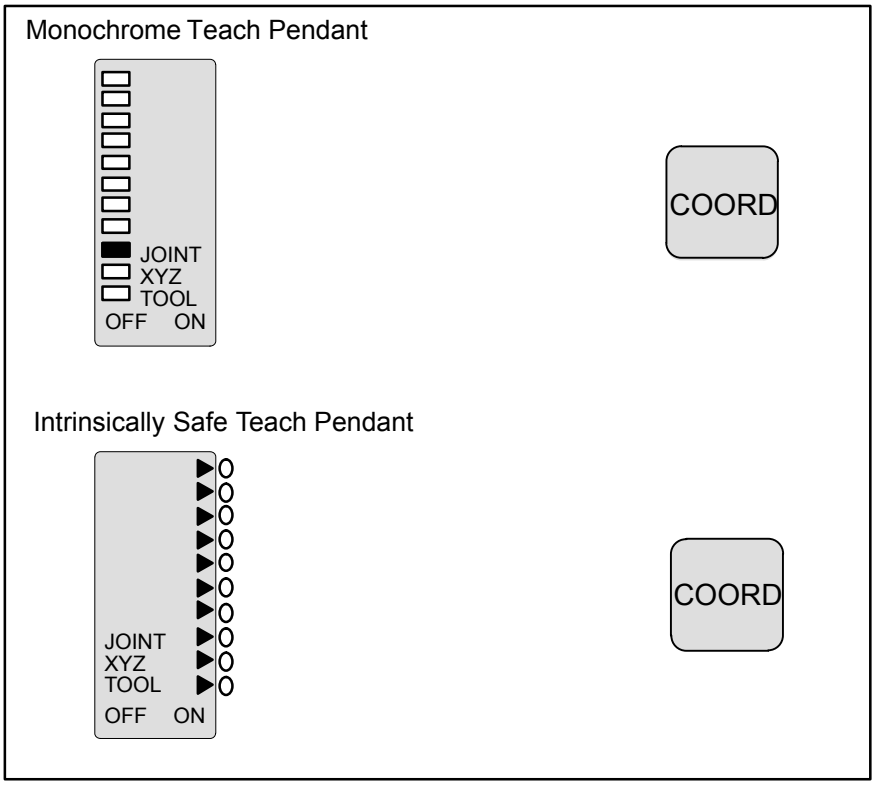


Figure 2–3. COORD Display for the iPendant

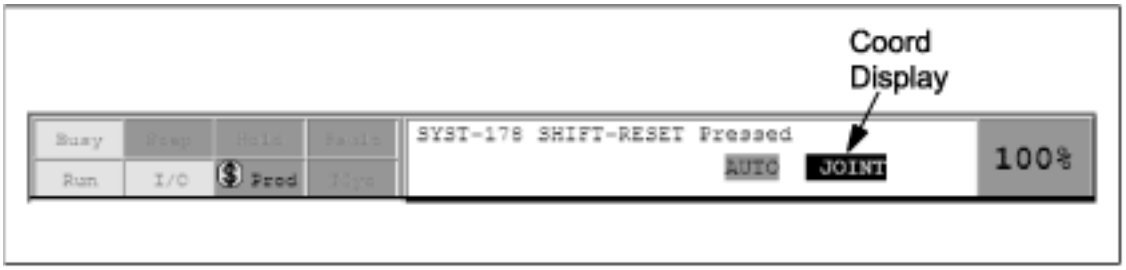


Table 2–3. LCD Indicators

LCD	Coordinate System
JOINT	JOINT
XYZ	WORLD, USER, JGFRAME
TOOL	TOOL
PATH	PATH (ArcTool only)

Refer to Table 2–4 for listings and descriptions of the JOINT, XYZ, and TOOL coordinate systems.

Table 2–4. Coordinate Systems

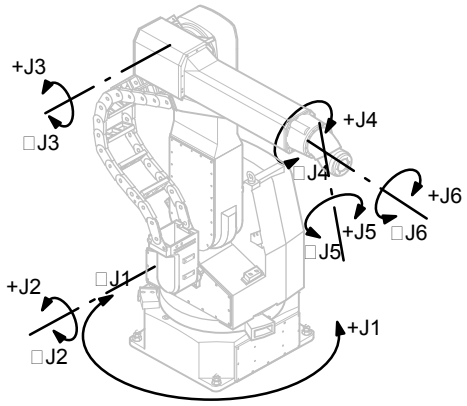
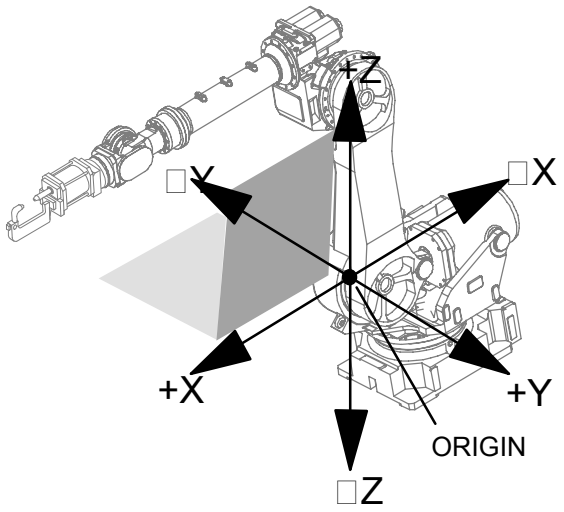
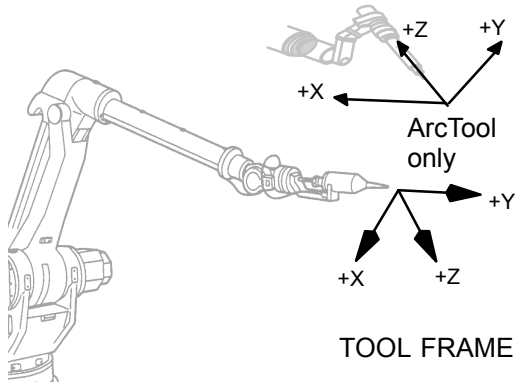
COORDINATE SYSTEM	DESCRIPTION	ILLUSTRATION
JOINT	Moves the individual axes of the robot. **Caution** Use of the JOINT motion mode with the F-200i series robots can cause damage to the robot. Jogging the F-200i series robot in JOINT is only permissible when single axis mastering.	

Table 2–4. Coordinate Systems (Cont'd)

COORDINATE SYSTEM	DESCRIPTION	ILLUSTRATION
XYZ	Moves the robot TCP in the x, y, or z directions and rotates about x (w), y (p), or z (r).	
TOOL	Moves the robot TCP in the x, y, or z direction and rotates x(w), y(p), and z(r) in the selected tool frame.	

2.3.4 Wrist Jogging

The wrist jog function allows you to control how the robot axes will jog when you are using a Cartesian coordinate system, such as WORLD or TOOL. In wrist jog, the wrist axes are fixed and do

not change when jogging in x, y, and z coordinates. This is helpful when you have to jog in the x, y, and z directions through a singularity position.

When you jog a wrist axis using wrist jog, the other wrist axes will remain stationary and the rest of the robot axes will move to accommodate the movement of the wrist axes to maintain a fixed TCP location.

Note If you are using a paint application, the gun orientation changes relative to the part and affects only the fan pattern.

You select wrist jog using the FCTN menu. When you select wrist jog, "W/" appears next to the coordinate system name displayed on the teach pendant screen as shown below.

```
PROGRAM NAME                          W/TOOL 10%
```

Use Procedure 2-4 to select wrist jog and jog the axes.

2.3.5 Remote TCP Jogging

The optional remote TCP jog function allows you to control how the robot axes will jog. This is useful for applications in which the tool is fixed in the workcell and the robot manipulates the workpiece around the tool. The frame used for jogging is a user frame (UFRAME) you set up and select.

Note Remote TCP jogging will not be available if the Coordinated Motion option is loaded.

When remote TCP jogging is turned on, if you move the robot in x, y, or z using the TOOL coordinate system, the robot moves as it normally would without remote TCP jogging. If you move the robot in w, p, and r (rotational moves), the rotational center will be the remote TCP position.

In remote TCP jogging, an invisible tool is established to connect the faceplate to the remote TCP position. For example, if you select the WORLD coordinate system and jog the robot in w, the invisible tool will rotate along the WORLD x-axis. If you select the TOOL coordinate system and jog the robot in w, the invisible tool will rotate along the TOOL x-axis.

You select remote TCP jogging using the FCTN menu. You also select the remote TCP frame using the FCTN menu. When you select remote TCP jogging, "Rn/" appears next to the coordinate system name displayed on the teach pendant screen, where "n" is the number of the user frame, which can be from 1 to 5 as shown below.

```
PROGRAM NAME                    S   R1/TOOL 10%
```

You can perform remote TCP jogging only if you have selected a Cartesian coordinate system such as WORLD, TOOL, JOGFRAME, or USER. You cannot perform remote TCP jogging if you have selected the JOINT coordinate system. Use Procedure 2-4 to select remote TCP jogging, select the remote TCP frame, and jog the axes.

2.3.6 Motion Groups

A motion group defines different groups of axes that can be used for independent pieces of equipment, positioning tables, and other devices. If your system contains more than nine axes, there is more than one group that controls motion. The robot is in Group 1.

The robot controller can operate up to four full kinematics devices (robot mechanical units) in eight motion groups, with a total of 40 axes; however,

- there is a maximum of nine axes in a single group.
  - — Process Axis devices (ServoTorch, Dispense ISDT, etc.) are considered an axis with respect to the total number of axes, but they do not constitute a motion group.
  - — Line tracking encoders are limited to 4 channels, but they are not considered axes with respect to the total number of axes.
- non-robot motion groups of one to four axes can be defined. The maximum number includes extended axes such as General Positioner, Basic Positioner, and Independent Axis devices.
  - — The Index axis device is one axis by definition and the Arc Positioner device is a two-axis motion group device. Neither of these can have additional axes installed.

Note The limitation of a maximum of two Index Devices has been removed

Up to 24 axes are supported on the Main CPU PCB. Up to 16 axes are supported on an Auxiliary Axis PCB.

Extended Axes

One to three extended axes can be added to a motion group. The axes cannot be used independently of the motion group.

Each extended axis adds a position data field (E1, E2, E3) to the motion group data.

Group Mask

When you create a program, you define the *group mask* which is the group of axes that the program will control. A single program can be defined to use all eight motion groups, but a maximum of two motion groups can perform Cartesian interpolated motion within a single program.

With multiple groups, the axes that jog depend on which group you have selected. You select groups using the FCTN menu or by pressing the SHIFT and COORD keys. Use Procedure 2-4 to select groups and jog the axes.

To change the group number, you can also use the jog menu. Refer to Section 2.3.8 .

2.3.7 Extended Axes and Sub-Groups

Extended axes are the available axes controlled by the controller beyond the standard number of robot axes. There is a limit of three extended axes per motion group.

- One to three extended axes can be added to a motion group. The axes cannot be used independently of the motion group.
- Each extended axis adds a position data field (E1, E2, E3) to the motion group data.
- The limitation of a maximum of two Index Devices has been removed
- Process Axis devices (ServoTorch, Dispense ISDT, etc.) are considered an axis with respect to the total number of axes, but they do not constitute a motion group.

Extended axes become a *sub-group* of the motion group to which they belong. Normally, the teach pendant keys control the first six robot axes. To jog the extended axes in a sub-group, you must first select the sub-group. You select sub-groups using the FCTN menu. The status line at the top of the screen displays whether a sub-group is being used, as shown below.

```
PROGRAM NAME                        S  JOINT 10%
```

For example, if the sub-group controls axes 7, 8 and 9, select the sub-group and then refer to Table 2–5 .

Table 2–5. SubGroup Example

For Axis Number	Use Jog Keys
7	+X, -X
8	+Y, -Y
9	+Z, -Z

To change the sub-group number, you can also use the jog menu. Refer to Section 2.3.8 for more information.

Use Procedure 2-4 to select sub-groups and jog the robot and other axes.

Procedure 2-4 Jogging the Robot and Other Axes

Condition

- All personnel and unnecessary equipment are out of the workcell.
- All EMERGENCY STOP faults have been cleared. Refer to Section 7.2 .
- All other faults have been cleared and the fault light is not illuminated.
- The MODE SELECT switch is in the T1 or T2 position.

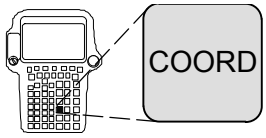
> **Warning**
>
> **Make certain that all safety requirements for your workplace have been followed; otherwise, you could injure personnel or damage equipment.**

Note If you are using ArcTool, the PATH coordinate system is available only when a program is paused while executing linear or circular motion instructions that do not use the wrist joint (Wjnt) motion option.

Step

> **Caution**
>
> Use of the JOINT motion mode with the F-200i series robots can cause damage to the robot. Jogging the F-200i series robot in JOINT is only permissible when single axis mastering.

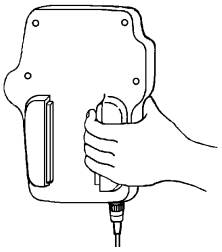
1. **Select a coordinate system** by pressing the COORD key on the teach pendant until the coordinate system you want is displayed in the upper right hand corner of the teach pendant screen, and on the teach pendant LEDs. You will see a screen similar to the following.




```
PROGRAM NAME                        S   TOOL 10%
```

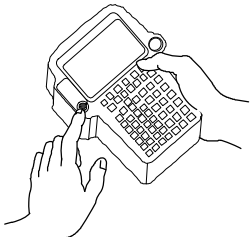
Note The jog speed value will automatically be set to 10%, when the teach pendant is turned on, or when the controller is first powered up.

2. Hold the teach pendant and continuously press the DEADMAN switch on the back of the teach pendant.

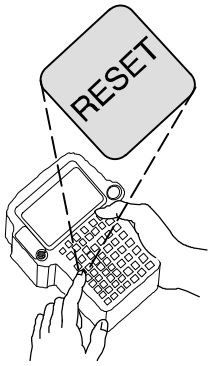


Note If you compress the DEADMAN switch fully, robot motion will not be allowed and an error occurs. This is the same as when the DEADMAN switch is released. **To clear the error,** press the DEADMAN switch in the center position and press RESET.

3. Turn the teach pendant ON/OFF switch to the ON position.



Note If you release the DEADMAN switch while the teach pendant is ON, an error will occur. **To clear the error,** continuously press the DEADMAN switch and then press the RESET key on the teach pendant.



4. **If your system is configured with multiple motion groups,** select the motion group you want to jog by doing the following:

a. Press and hold the SHIFT key and press COORD to display the JOG menu.

b. Move the cursor to GROUP and press the appropriate numeric key.

5. **If your system has extended axes,** select the sub-group that you want to jog by doing the following:

   a. Press FCTN.

   b. Move the cursor to TOGGLE SUB-GROUP and press ENTER You will see a screen similar to the following.

```
PROGRAM NAME                        S   JOINT 10%
```

   c. To de-select a sub-group press FCTN, move the cursor to TOGGLE SUB-GROUP, and press ENTER.

6. **If you want to use wrist jogging,**

   a. Press FCTN.

   b. Move the cursor to TOGGLE WRIST JOG and press ENTER. The status line indicator for wrist jog is displayed in the upper right hand corner of the teach pendant screen. See the following screen for an example.

```
PROGRAM NAME                      S    W/TOOL 10%
```

   c. To de-select wrist jogging press FCTN, move the cursor to TOGGLE WRIST JOG, and press ENTER.

   Note WRIST JOG is not available or applicable in "JOINT" mode.

7. **If you want to use optional remote TCP jogging,**

   a. Select the Cartesian coordinate system you want to use for remote TCP jogging.

   b. Press FCTN.

   c. Move the cursor to TOGGLE REMOTE TCP and press ENTER. The status line indicator for remote TCP jogging is displayed as "Rn/" in the upper right hand corner of the teach pendant screen next to the coordinate system, where "n" is the number of the remote TCP frame.

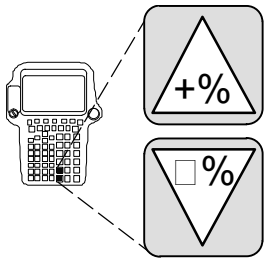
```
PROGRAM NAME                      S   R1/TOOL 25%
```

   d. Press FCTN.

e. Move the cursor to CHANGE RTCP FRAME and press ENTER. Each time you select CHANGE RTCP FRAME, the user frame selection is advanced: from 1 (R1) to 2 (R2) to 3 (R3) to 4 (R4) to 5 (R5) and then back to 1 (R1). Select the user frame (UFRAME) you want to use for remote TCP jogging.

f. To de-select remote TCP jogging, press FCTN and move the cursor to TOGGLE REMOTE TCP and press ENTER.

8. Select a jog speed by pressing and releasing the appropriate jog speed key until the jog speed you want is displayed in the upper right hand corner of the teach pendant screen, as shown below.




```
PROGRAM NAME                          S   TOOL 25%
```

Note Set the jog speed to a low percentage (%) value if you are inexperienced in jogging the robot, or if you are uncertain how the robot will move.

Warning

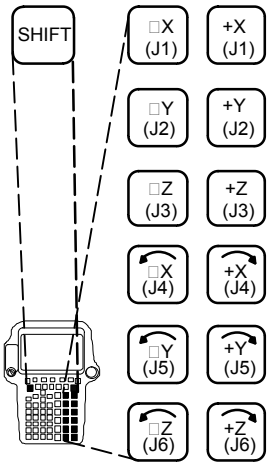
In the next step, the robot will move. To stop the robot immediately any time during jogging, release the DEADMAN switch or press the EMERGENCY STOP button.

9.

Caution

Use of the JOINT motion mode with the F-200i series robots can cause damage to the robot. Jogging the F-200i series robot in JOINT is only permissible when single axis mastering.

To jog, press and hold the SHIFT key and continuously press the jog key that corresponds to the direction in which you want to move the robot. **To stop jogging, release the jog key.**



Note If you have set the singularity stop system variable, $PARAM_GROUP[n].$T1T2_SNGSTP to TRUE, the robot will stop at singularity points while in T1 or T2 mode.

10. **When you are finished jogging,** turn the teach pendant ON/OFF switch to OFF, and release the DEADMAN switch.

2.3.8 Jog Menu

The *jog menu* provides a method to check and change the following jogging information:

- Currently selected frame number of each frame (TOOL, JOG, USER)
- Currently selected group number
- Currently selected sub-group type (ROBOT/EXT)

Jog Menu Modes of Operation

The jog menu has two modes of operation: AutoClose and Sticky.

- **In AutoClose mode** the jog menu operates as is has in the past. To change an item, press a single key on the TOOL, JOG, USER or Group lines. The item will change and the menu will close. AutoClose is the default if Extended User Frames is not loaded and fewer than nine UFrames or ten TOOL Frames are configured.

  The (.=10) on the Tool line indicates that AutoClose mode is active.

The dot (".") key is interpreted as entering 10. This allows you to set the TOOL Frame to 10 in AutoClose mode.

- **In Sticky mode** you can make multiple entries. You must close the menu when done.

In Sticky Mode, the < > characters indicate that a number is being entered but has not been applied yet. In the following example screen the jog menu is displayed on a controller with two motion groups.

```
Tool        14
Jog          3
User        55
Group        2
```

In Sticky Mode with Tool selected, when you press 2 you will see a display similar to the following:

```
Tool       <2>
Jog          3
User        55
Group        2
```

The < > symbols show that the number is being entered but has not yet been applied. When you press another number key, for example 3, the display will show <23>. You can accept this entry by

— pressing any one of ENTER, COORD, PREV, or a function Key. This will close the jog menu.

— pressing the up or down arrow. This will accept the entry, move the cursor to the item above or below, and keep the jog menu displayed.

You can edit a number before it has been accepted:

— pressing Back Space will remove the last number you entered.

— pressing Back Space again will remove the next to last number you entered.

In the example above, when <23> is displayed, after you press Back Space the <2> will remain. Pressing backspace again will remove the 2 and display the original number 14 without the < > characters.

Sticky mode is the default mode when the Extended user frames option is loaded and more than nine UFrames or ten TOOL Frames are configured. If fewer than nine UFrames or ten TOOL Frames are configured, you can enter Sticky mode by pressing ENTER. The (.=10) missing from the Tool line indicates that Sticky mode is active.

Use Procedure 2-5 to display and use the jog menu.

Procedure 2-5 Using the Jog Menu

Steps

1. To display the jog menu, press and hold the SHIFT key and press the COORD key.
2. Use the up and down arrow keys to move the cursor to the item you want to change.
3. **To change the number of each frame** , press the appropriate numeric key. The item on which the cursor is located is changed to the new value. Valid frame numbers are as follows:

   - USER: 0-9. With Extended User Frames, USER: 0-61
   - TOOL: 1-10. With Extended User Frames, TOOL: 0-29. To select 10, press the "." (period) key on the teach pendant.
   - JOG: 1-5

   Note The text (.=10) does not appear and the "." key is not accepted if there are fewer than ten tool frames.

   Note The maximum number of tool frames is set in the system variable $SCR.$MAXNUMUTOOL. Refer to the *Software Reference Manual* and Section 10.21 for more information.

4. To change to sub-group (available only for systems with extended axes), move the cursor to ROBOT/EXT and press the left and right arrow keys.
5. To change the group number (available only for multiple motion group systems), move the cursor to GROUP and press the appropriate numeric key. You can specify numbers only for existing motion groups.
6. **To close the jog menu without entering a number,**

   - Press SHIFT and COORD again.

   or

   - Press the PREV key.

7. **To close the jog menu after you enter a number,**

   - **in Auto Close mode:**
     If Extended User Frames is not loaded and fewer than nine UFrames or ten TOOL Frames are configured, the jog menu closes automatically after you enter a frame or group number.
   - **in Sticky Mode:**
     If Extended User Frames is loaded and more than nine UFrames or ten TOOL Frames are configured, you can close the jog menu by pressing any one of ENTER, COORD, PREV, or a function Key.

Chapter 3

GENERAL SETUP

Contents

Chapter 3	GENERAL SETUP	3–1
3.1	FRAMES SETUP	3–3
3.1.1	Frames Setup Overview	3–3
3.1.2	Tool Frame	3–6
3.1.3	User Frame	3–27
3.1.4	Remote TCP Frame	3–50
3.1.5	Jog Frame	3–61
3.1.6	Saving Frame Data	3–74
3.2	PRODUCTION OPERATION SETUP	3–76
3.2.1	Production Operation Setup Overview	3–76
3.2.2	Robot Service Request (RSR) Setup	3–82
3.2.3	Program Number Select (PNS) Setup	3–86
3.2.4	Style Name Setup	3–92
3.2.5	OTHER Program Select Mode	3–98
3.3	MACRO COMMANDS	3–99
3.3.1	Overview	3–99
3.3.2	Setting Up Macro Commands	3–99
3.3.3	Executing Macro Commands	3–113
3.3.4	Assigning I/O and Recording Positions in Macro Command Programs	3–117
3.3.5	HandlingTool Macro Commands	3–118
3.4	AXIS LIMITS SETUP	3–119
3.5	BRAKE TIMERS SETUP	3–122
3.6	SETUP GENERAL SETUP SCREEN	3–127
3.6.1	Overview	3–127
3.6.2	Brake on Hold Setup	3–127
3.6.3	Current Language Setup	3–129
3.6.4	Ignore Offset Setup	3–129
3.6.5	Ignore Tool Offset Setup	3–130

3.7 USER ALARM SETUP 3–131
3.8 OVERRIDE SELECT SETUP 3–134
3.9 PASSWORD SETUP 3–137
3.9.1 Password Operations Overview 3–137
3.9.2 Install User Password Operations 3–138
3.9.3 Disabling Passwords 3–142
3.9.4 Program and Set Up User Password Operations 3–143
3.9.5 Configuring Passwords 3–148
3.9.6 Password Log 3–162
3.9.7 Password Level Screen Permissions for HandlingTool 3–165
3.9.8 USB Password Function 3–169
3.9.9 Using KCL with Passwords Enabled 3–171
3.9.10 Password Auto Login Function 3–172
3.10 ERROR SEVERITY TABLE 3–174
3.10.1 Error Severity Table Overview 3–174
3.10.2 Modifying Error Severity 3–174
3.11 ERROR CODE OUTPUT SETUP (OPTION) 3–179
3.11.1 Error Code Output Setup Overview 3–179
3.11.2 Method 1: Output Errors Using 33 Digital Outputs 3–180
3.11.3 Method 2: Output Errors Using 3 Group Outputs 3–185
3.11.4 Output Error Parameters 3–186
3.11.5 Procedure 3–187
3.12 ROBOT PAYLOAD SETTING 3–188
3.12.1 Overview 3–188
3.12.2 Setting the Active Payload Schedule 3–188
3.12.3 Setup of Payload Schedules 3–190
3.12.4 Using Payload Ident. (Option 669) to Set Up Payload Schedules 3–195
3.12.5 Setting Up Arm Load Information 3–209
3.13 STROKE LIMIT SETUP 3–210
3.14 TEACHING KAREL VARIABLES (DispenseTool and HandlingTool) 3–212
3.14.1 Overview 3–212
3.14.2 KAREL Positions 3–212
3.14.3 KAREL Paths 3–215
3.14.4 KAREL Variables 3–224
3.15 SYSTEM CONFIGURATION SETUP 3–226

3.1 FRAMES SETUP

3.1.1 Frames Setup Overview

A *frame* is a set of three planes at right angles to each other. The point where all three planes intersect is the *origin* of the frame. This set of planes is called a *Cartesian coordinate system*. In the robot system, the intersecting edges of the planes are the x, y, and z axes of the frame.

Note If you are using PalletTool, your gripper might be set up differently than the one shown in the illustrations in this section. The orientation of the gripper depends on how the pneumatics of the gripper are set up. For example, if you are palletizing very large boxes, your gripper might be mounted perpendicularly to what is shown in the illustrations.



Caution

Do not set up or alter frames when you use PalletTool. Frames are automatically set up for you by PalletTool.

If you are using DispenseTool, refer to the *Line Tracking Setup and Operations Manual* for more information about setting up line tracking encoders and the line tracking frames for use with the Line Tracking option. However, in the Advanced Techniques chapter of the *Line Tracking Setup and Operations Manual*, do not use the section on Tracking Part Queues. Instead, refer to the section "Editing the Job Queue."

How Frames are Used

Frames are used to describe the *location* and *orientation* of a position. The location is the distance in the x, y, and z directions from the origin of the reference frame. The orientation is the rotation about the x, y, and z axes of the reference frame. When you record a position, its location and orientation are automatically recorded as x, y, z, w, p, and r relative to the origin of the frame it uses as a reference.

The location of a position is expressed as three dimensions, which are measured in millimeters from the origin in the x, y, and z directions. For example, 300,425,25 means the position is 300mm in the x direction, 425mm in the y direction, and 25mm in the z direction from the origin.

The orientation of a position is expressed as three dimensions, which are measured in degrees of rotation about the x, y, and z axes. For example, 0,-90,0 means that the position is rotated -90 degrees about the y axis and is not rotated about the x or z axes.

Kinds of Frames

The robot uses four kinds of frames. The different kinds of frames make it easier to perform certain tasks. The kinds of frames are

- World frame - the default frame of the robot

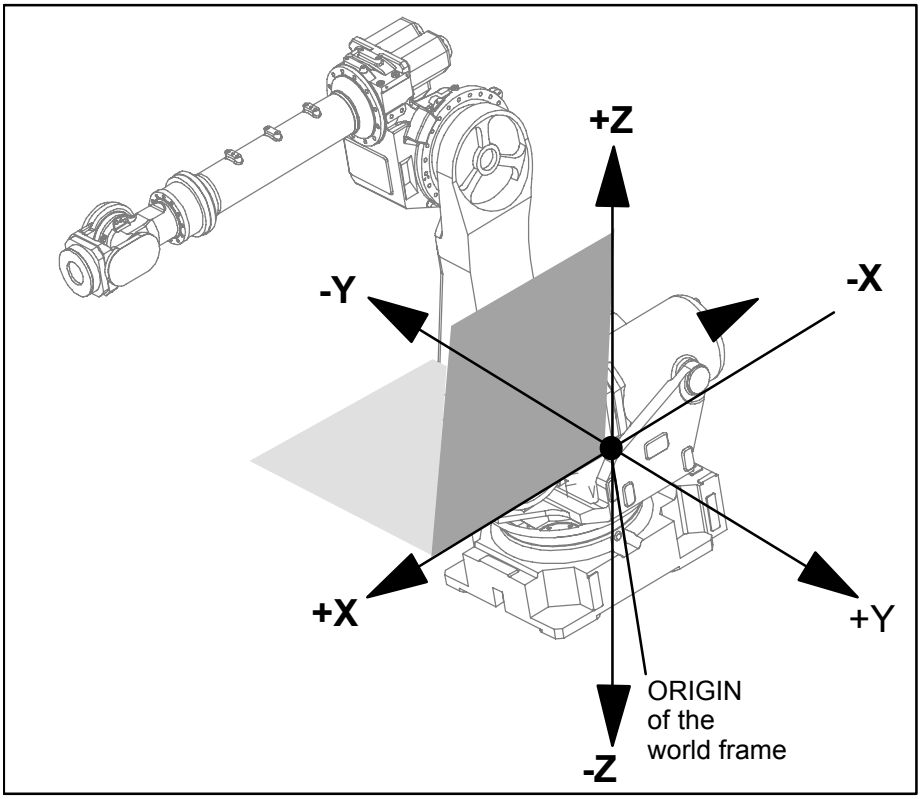
- Tool frame - a user-defined frame
- User frame - a user-defined frame
- Jog frame - a user-defined frame

World Frame

The world frame is a default frame that cannot be changed. The origin of the world frame (0,0,0,0,0,0) is the reference position for user frame and jog frame. The origin is located at a predefined position within the robot.

See Figure 3–1 for an example. Your robot might be different depending on your system.

Figure 3–1. World Frame



Tool Frame

The default tool frame is a Cartesian coordinate system that has the position of the tool center point (TCP) at its origin. When you set up a UTool, you move the default UTool from the robot faceplate to the point on the applicator, gun, torch, or tool at which the painting, welding, sealing, handling, or other application work is to be done.

User Frame

The user frame is the reference frame for all recorded positional data in a program. You can modify the user frame to offset the positions in the program easily. You can define this frame anywhere.



Caution

Every time you create a program, set the current user frame number to a value between 1 and 6. Do this even if you do not plan to use a user frame in the program, or if you want the user frame position to be zero (0, 0, 0, 0, 0, 0). Otherwise, if the current user frame number is zero, a user frame set in that program will not work.

Remote TCP (RTCP) Frame (HandlingTool, DispenseTool, and SpotTool+ only)

The remote TCP (RTCP) frame is a kind of user frame you must define in order to use remote TCP jogging and the remote TCP motion option. You define this frame using the location of the remote TCP as the origin of the frame.

Jog Frame

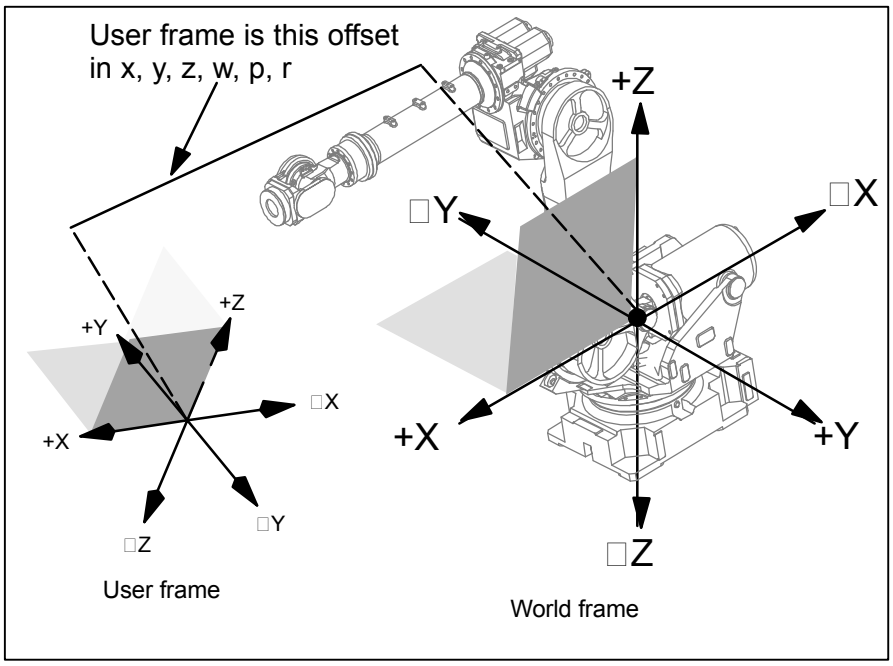
The jog frame is a frame in which to jog easily. It allows you to align the x, y, z coordinate system about a fixture or workpiece that is rotated with respect to the world frame of the robot.

Moving the Location and Orientation of a Frame

You can move the location and orientation of any frame except the world frame. When you move the location or orientation of a frame, all positions recorded with that frame also move. However, the location of those positions will stay the same within that frame.

See Figure 3–2 for the default location of the user frame.

Figure 3–2. Moving a Frame



Caution

If you change any**TOOL or USER** frame data after a program has been taught, you must reteach each program position or range. If you do not, **damage could occur** to the equipment.

3.1.2 Tool Frame

3.1.2.1 Setting Up Tool Frame

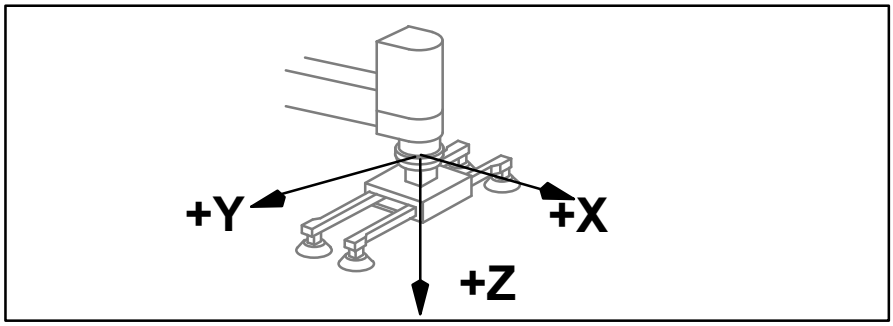
By default, the origin of the tool frame is on the faceplate of the robot. You must move the origin of the tool frame to the position, both location and orientation, where the work is to be done. This position is called the *tool center point (TCP)* .

For example, **in ArcTool,** the TCP is the tip of the wire; **in PaintTool,** the TCP is approximately 12 inches from the end of the applicator, but this can vary depending on your particular applicator; **in SpotTool+,** the TCP is where the tips of the gun meet when they are closed.

All measurements in tool frame are relative to the origin of the tool frame.

- Before you use tool frame, you must set up its location and orientation.
- For the KAREL robot commands MOVE NEAR and MOVE RELATIVE, the z-axis of the tool frame is the approach vector of the tool. That is, the z-axis defines the path the tool will take as it approaches the workpiece.
- You can set up as many as ten different tool frames for each robot. They will be stored in the system variable $MNUTOOL.
- You can select one tool frame to be active. The frame number will be stored in the system variable $MNUTOOLNUM.
- **In ArcTool,** the weave plane is relative (perpendicular) to tool z.
- You can jog the robot in tool frame.

Figure 3–3. HandlingTool Tool Frame



Warning

If you are using weaving, Thru-Arc Seam Tracking (TAST), coordinated motion, or TorchMate, you must define a six point tool frame by the six point method or the direct entry method. Failure to do so could injure personnel or damage equipment.

If a system uses different end-of-arm-tooling (EOAT), each tool will need a different tool frame. By setting up a different tool frame for each tool, the existing program points will be valid, regardless of the tool used.

Table 3–1 lists the tool frame setup items and function key operations.

Table 3–1. Tool Frame Setup Screen Items

ITEMS	DESCRIPTION
Tool Frame Setup/xxxx	This line shows the current selected method for setting each tool frame.
Frame number 1-10	These lines show the current position and comment for each available tool frame.

Table 3–1. Tool Frame Setup Screen Items (Cont'd)

X	This column shows the coordinate of each tool frame.
Y	This column show the coordinate of each tool frame.
Z	This column show the coordinate of each tool frame.
Comment	This column shows the comment for each tool frame.
Active TOOL $MNUTOOLNUM[1] = 1	This line indicates which tool frame is currently selected for use.
[TYPE]	Press this key to access various application-specific options.
DETAIL	Press this key to display detailed information for each tool frame and to set the definition and comment of each tool frame.
OTHER	Press this key to select one of the other available reference frames or to choose the motion group for the frame you are setting up (in systems with multiple motion groups).
CLEAR	To set the numerical values of any tool frame to zero, move the cursor to the frame number, press CLEAR and answer YES to the prompt.
SETIND	To select the tool frame to use, press F5, SETIND, type the number of the tool frame you want, and press ENTER. This sets the active tool frame ($MNUTOOLNUM[1]) to the number of the frame you want. If F5, SETIND, is not displayed, press PREV.

You can use the following methods to define the tool frame:

- Three point method
- Six point method
- Direct entry method

Note If you have a 4-axis robot, you can define a tool frame using only the direct entry method.

Three Point Method

Use the three point method to define the **location** of the tool frame when the values cannot be measured and directly entered. The three approach points must be taught with the tool touching a common point from three different approach directions. Do not use this method to set up the P-10 opener.

Note The three point method of defining the tool frame always places the +z direction of the frame outward from the faceplate. You cannot use this method to define the tool frame for weaving, Thru-Arc Seam Tracking, or touch sensing if you are using ArcTool.

Table 3–2. Tool Frame Setup Three Point DETAIL Screen Items

ITEMS	DESCRIPTION
Frame number	This line shows the number of the tool frame being defined.
X, Y, Z, W, P, R	These items show the current coordinates of the tool frame being defined.
Comment	This item is used to add a comment.
Approach point 1	This item is used to define the first approach point.
Approach point 2	This item is used to define the second approach point.
Approach point 3	This item is used to define the third approach point.
Active TOOL $MNUTOOLNUM[1] = 1	This line indicates which tool frame is currently selected for use.
[TYPE]	Press this key to access various application-specific options.
[METHOD]	Press this key to select a method of entry.
[FRAME]	Press this key to select a frame.
MOVE_TO	Press this key to move to a recorded position. Move the cursor to the desired point and press MOVE_TO along with the SHIFT key.
RECORD	Press this key along with the SHIFT key to record a position.

Six Point Method

Use the six point method to define the **location and orientation** of the tool frame when the values cannot be measured and directly entered. The six point method requires three points that define the direction vector for the tool, and three points that define the location of the tool center point.

Table 3–3. Tool Frame Setup Six Point DETAIL Screen Items

ITEM	DESCRIPTION
Frame number	This line shows the number of the tool frame being defined.
X, Y, Z, W, P, R	These items show the current coordinates of the tool frame being defined.
Comment	This item is used to add a comment.
Approach point 1	This item is used to define the first approach point.
Approach point 2	This item is used to define the second approach point.
Approach point 3	This item is used to define the third approach point.
Orient Origin Point	This item is used to define the reference point of the user frame.
X Direction Point	This item is used to define the +x direction point.
Z Direction Point	This item is used to define the +z direction point

Table 3–3. Tool Frame Setup Six Point DETAIL Screen Items (Cont'd)

Active TOOL $MNUTOOLNUM[1] = 1	This line indicates which tool frame is currently selected for use.
[TYPE]	Press this key to access various application-specific options.
[METHOD]	Press this key to select a method of entry.
[FRAME]	Press this key to select a frame.
MOVE_TO	Press this key to move to a recorded position. Move the cursor to the desired point and press MOVE_TO along with the SHIFT key.
RECORD	Press this key along with the SHIFT key to record a position.

Direct Entry Method

The direct entry method provides for direct recording and numerical entry of the frame position. For TCP dimensions, refer to the manufacturing specifications of the tool. Use Procedure 3-3 to set up the tool frame using the direct entry method.

Table 3–4. Tool Frame Setup Direct Entry DETAIL Screen Items

ITEM	DESCRIPTION
Frame number	This line shows the number of the tool frame being defined.
Comment	This item is used to add a comment.
X	This item is used to define the X component of the tool frame.
Y	This item is used to define the Y component of the tool frame.
Z	This item is used to define the Z component of the tool frame.
W	This item is used to define the W component of the tool frame.
P	This item is used to define the P component of the tool frame.
R	This item is used to define the R component of the tool frame.
Configuration	This line displays the current robot configuration.
Active TOOL $MNUTOOLNUM[1] = 1	This line indicates which tool frame is currently selected for use.
[TYPE]	Press this key to access various application-specific options.
[METHOD]	Press this key to select a method of entry.
[FRAME]	Press this key to select a frame.

3.1.2.2 HandlingTool Tool Frame Three Point Method

Use Procedure 3-4 to set up a tool frame using the three point method.

Procedure 3-1 Setting Up Tool Frame Using the Three Point Method

> **Warning**
>
> **If you set up a new frame, make sure that all frame data is zero or uninitialized before you record any positions. Press F4, CLEAR, to clear frame data.**
>
> **If you modify an existing frame, make sure that all frame data is set the way you want before you change it.**
>
> **Otherwise, you could injure personnel or damage equipment.**

> **Caution**
>
> Do not set up or alter frames when you use PalletTool. Frames are automatically set up for you by PalletTool.

Note If you have a 4-axis robot (such as an A-520i or M-410iHS/iHW), you can define a tool frame using only the direct entry method.

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Frames.
5. **To choose the motion group** for the frame you are setting up in systems with multiple motion groups, press F3, [OTHER], and select the group you want. The default motion group is Group 1.

   If you have the new *i*Pendant, you can press the GROUP key and press the numeric key to switch to a specified group. Press the GROUP key and the 0 key together to toggle the sub group.
6. **If tool frames are not displayed**, press F3, [OTHER], and select Tool Frame. If F3, [OTHER], is not displayed, press PREV.
7. **To display the settings for all frames**, press PREV repeatedly until you see a screen similar to the following.

```
SETUP Frames
Tool Frame Setup / Three Point
X       Y        Z     Comment
1:   0.0    0.0     0.0 *************
2:   0.0    0.0     0.0 *************
3:   0.0    0.0     0.0 *************
4:   0.0    0.0     0.0 *************
5:   0.0    0.0     0.0 *************
6:   0.0    0.0     0.0 *************
7:   0.0    0.0     0.0 *************
8:   0.0    0.0     0.0 *************
9:   0.0    0.0     0.0 *************
Active TOOL $MNUTOOLNUM[1]=1
```

Note The maximum number of tool frames is set in the system variable $SCR.$MAXNUMUTOOL. Refer to the *Software Reference Manual* for more information.

8. **To set the numerical values to zero,** move the cursor to the frame number, press F4, CLEAR, and then press F4, YES, to confirm.



Warning

Do not run a KAREL program that includes motion statements if more than one motion group is defined on your controller. If your controller is set up for more than one motion group, all motion must be initiated from a teach pendant program. Otherwise, the robot could move unexpectedly, personnel could be injured, and equipment could be damaged.

9. Press F2, DETAIL.
10. **To select a frame,**
    a. Press F3, FRAME.
    b. Type the desired frame number.
    c. Press ENTER.
11. Press F2, [METHOD].
12. Select Three Point. You will see a screen similar to the following.

```
SETUP Frames
Tool Frame Setup / Three Point
Frame Number: 1
X:  0.0     Y:  0.0     Z:  0.0
W:  0.0     P:  0.0     R:  0.0
Comment: ***************
Approach point 1:     UNINIT
Approach point 2:     UNINIT
Approach point 3:     UNINIT
Active TOOL $MNUTOOLNUM[1]=1
```

13. To add a comment:

a. Move the cursor to the comment line and press ENTER.

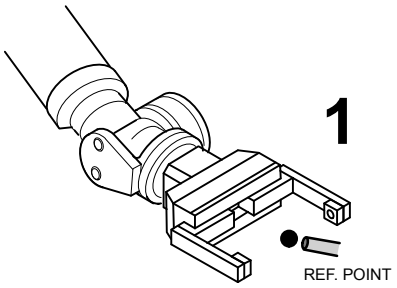
b. Select a method of naming the comment.

c. Press the appropriate function keys to enter the comment.

d. When you are finished, press ENTER.

Note Record the three approach points with the tool tip touching the same point from three different approach directions.

14. Record the first approach point **(Approach Point 1)**:

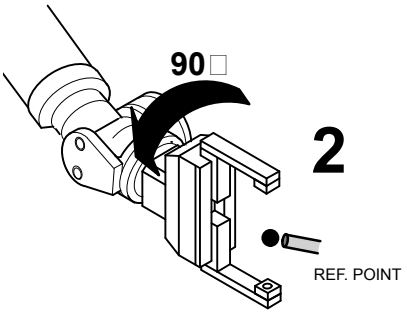


a. Move the cursor to Approach point 1.

b. Jog the robot so that the tool tip touches a reference point.

c. Press and hold the SHIFT key and press F5, RECORD.

15. Record the second approach point **(Approach Point 2)**:



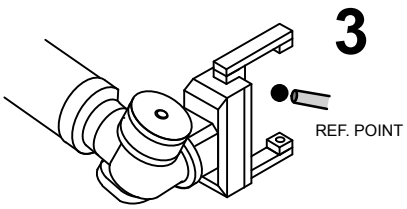
a. Move the cursor to Approach point 2.

b. Rotate the faceplate at least 90° (but no more than 360°) about the z axis of the tool coordinates.

c. Jog the robot so that the tool tip touches the reference point used in Step 14 .

d. Press and hold the SHIFT key and press F5, RECORD.

16. Record the third approach point **(Approach Point 3)**:



a. Move the cursor to Approach point 3.

b. Rotate the tool about either the x or y axis of the tool coordinates.

c. Jog the robot so that the tool tip touches the reference point used in Step 14 .

d. Press and hold the SHIFT key and press F5, RECORD.

17. **To select the tool frame to use,** press F5, SETIND, type the number of the tool frame you want, and press ENTER. If F5, SETIND, is not displayed, press PREV.

-OR-

Use the Jog Menu. Press and hold SHIFT and press COORD, move the cursor to Tool, and type the number of the frame you want to use. Refer to Section 2.3.8 for more information.

18. Jog the robot in the frame you just taught.

- If the TCP is correct, it will remain stationary during rotational moves. Go to Step 20 .

> **Caution**
>
> When you are finished setting the frame configuration, save the information to the default device so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

- If the TCP is not correct, it will not remain stationary during rotational moves. You need to review your recorded positions. If they are not correct, re-record them correctly. Go to Step 19 .

> **Warning**
>
> **When you use F4, MOVE_TO, to move the robot, unexpected motion can occur. This could injure personnel or damage equipment.**

19. **To move to a recorded position,** move the cursor to the desired position, press and hold the SHIFT key and press F4, MOVE_TO.

20. **To save the frames and related system variables to a file** on the default device,

a. Press MENUS.

b. Select FILE.

c. Press F1, [TYPE].

d. Select File.

e. Press F5, [UTIL].

f. Select Set Device.

g. Move the cursor to the device you want and press ENTER.

h. Display the tool frame screen.

i. Press FCTN.

j. Select SAVE. This will save the frame positions and comments for all frames to the file FRAMEVAR.VR, and the frame data to SYSFRAME.SV, on the default device.

Display the SYSTEM Variables menu,

k. Press MENUS.

l. Select SYSTEM.

m. Press F1, [TYPE].

n. Select Variables.

o. Press FCTN.

p. Select SAVE. The frame positions and system variables are saved in the SYSVAR.SV file, on the default device. This is optional since frame variables are saved in SYSFRAME.SV.

3.1.2.3 HandlingTool Tool Frame Six Point Method

Procedure 3-2 Setting Up Tool Frame Using the Six Point Method

> **Warning**
>
> **If you set up a new frame, make sure that all frame data is zero or uninitialized before you record any positions. Press F4, CLEAR, to clear frame data.**
>
> **If you modify an existing frame, make sure that all frame data is set the way you want before you change it.**
>
> **Otherwise, you could injure personnel or damage equipment.**

Note If you have a 4-axis robot (such as an A-520i or M-410iHS/iHW), you can define a tool frame using only the direct entry method.

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Frames.
5. **To choose the motion group** for the frame you are setting up in systems with multiple motion groups press F3, [OTHER], and select the group you want. The default motion group is Group 1.

   If you have the new *i*Pendant, you can press the GROUP key and press the numeric key to switch to a specified group. Press the GROUP key and the 0 key together to toggle the sub group.
6. **If tool frames are not displayed,** press F3, [OTHER], and select Tool Frame. If F3, [OTHER], is not displayed, press PREV.
7. **To display the settings for all the frames,** press PREV repeatedly until you see a screen similar to the following.

```
SETUP Frames
Tool Frame Setup / Six Point
X       Y      Z           Comment
1:     0.0     0.0    0.0       *************
2:     0.0     0.0    0.0       *************
3:     0.0     0.0    0.0       *************
4:     0.0     0.0    0.0       *************
5:     0.0     0.0    0.0       *************
6:     0.0     0.0    0.0       *************
7:     0.0     0.0    0.0       *************
8:     0.0     0.0    0.0       *************
9:     0.0     0.0    0.0       *************
Active TOOL $MNUTOOLNUM[1]=1
```

Note The maximum number of tool frames is set in the system variable $SCR.$MAXNUMUTOOL. Refer to the *Software Reference Manual* for more information.

8. **To set the numerical values to zero,** move the cursor to the frame number, press F4, CLEAR, and then press F4, YES, to confirm.



Warning

Do not run a KAREL program that includes motion statements if more than one motion group is defined on your controller. If your controller is set up for more than one motion group, all motion must be initiated from a teach pendant program. Otherwise, the robot could move unexpectedly, personnel could be injured, and equipment could be damaged.

9. Press F2, DETAIL.
10. **To select a frame,**
    a. Press F3, FRAME.
    b. Type the desired frame number.
    c. Press ENTER.
11. Press F2, [METHOD].
12. Select Six Point. You will see a screen similar to the following.

```
SETUP Frames
Tool Frame Setup/ Six Point
Frame Number: 1
X:    0.0     Y:  0.0     Z:  0.0
W:    0.0     P:  0.0     R:  0.0
Comment: ***************
Approach point 1: UNINIT
Approach point 2: UNINIT
Approach point 3: UNINIT
Orient Origin Point: UNINIT
X Direction Point: UNINIT
Z Direction Point: UNINIT
Active TOOL $MNUTOOLNUM[1]=1
```

13. To add a comment:

a. Move the cursor to the comment line and press ENTER.

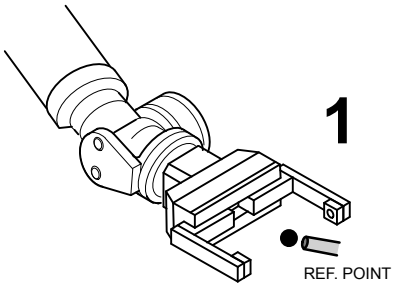
b. Select a method of naming the comment.

c. Press the appropriate function keys to enter the comment.

d. When you are finished, press ENTER.

Note Record the three approach points with the tool tip touching the same point from three different approach directions. The Tool frame will be inaccurate if the approach points face each other.

14. Record the first approach point **(Approach Point 1)**:

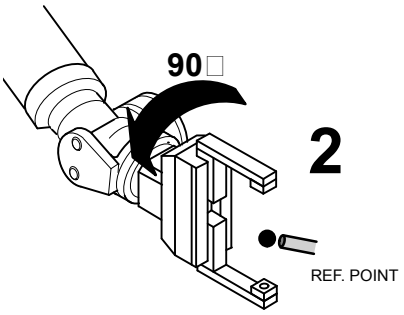


a. Move the cursor to Approach point 1.

b. Jog the robot so that the tool tip touches a reference point.

c. Press and hold the SHIFT key and press F5, RECORD.

15. Record the second approach point **(Approach Point 2)**:



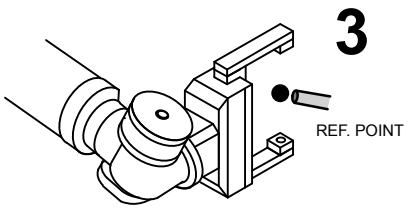
a. Move the cursor to Approach point 2.

b. Rotate the faceplate at least 90 (but no more than 180) about the z axis of the tool coordinates.

c. Jog the robot so that the tool tip touches the reference point used in Step 14 .

d. Press and hold the SHIFT key and press F5, RECORD.

16. Record the third approach point **(Approach Point 3)**:



a. Move the cursor to Approach point 3.

b. Rotate the tool about either the x or y axis of the tool coordinates.

c. Jog the robot so that the tool tip touches the reference point used in Step 14 .

d. Press and hold the SHIFT key and press F5, RECORD.

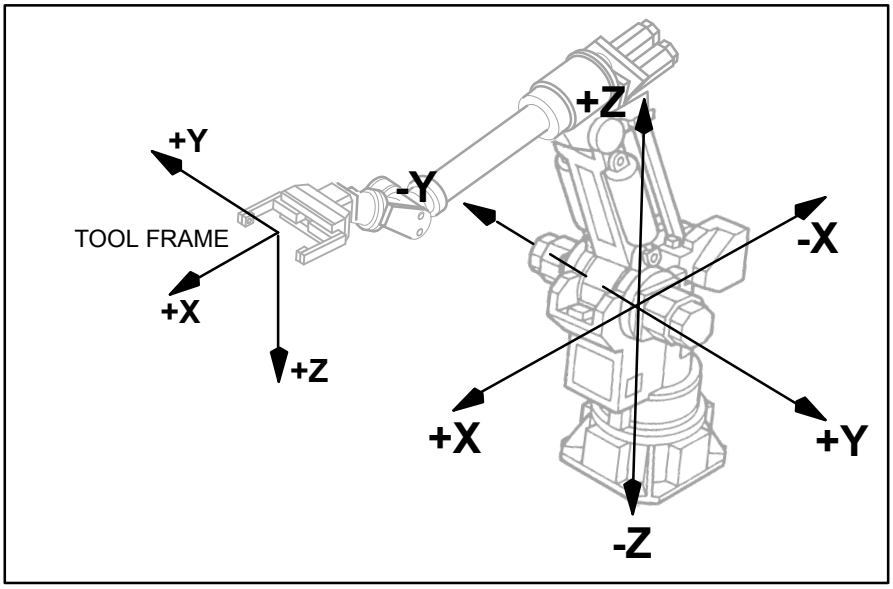
17. Define the orientation of the origin **(Orient Origin Point)**:

a. Move the cursor to Orient Origin Point.

b. Jog the robot so that the desired tool +z axis is parallel to the z axis of the world frame, pointing in the -z direction. Make sure that the x axis of the tool is parallel to the x axis of the world frame. See Figure 3–4 .

c. Press and hold the SHIFT key and press F5, RECORD.

Figure 3–4. Defining the Orientation of the Origin



18. Define the +x direction point **(X Direction Point)**:

a. Move the cursor to X Direction Point.

b. Change the jog coordinate system to WORLD.

c. Jog the robot so that the tool moves in the +x direction. For example, if the x axis of the tool is aligned with the world x axis, jog in the +x direction.

Note To assist you in moving the tool in the +x direction, move the tool at least 250mm or more.

d. Press and hold the SHIFT key and press F5, RECORD.

19. Define the +z direction point **(Z Direction Point)** :

a. Move the cursor to Orient Origin Point.

b. Press and hold the SHIFT key and press F4, MOVE_TO, to move the robot to the Orient Origin Point.

c. Move the cursor to Z Direction Point.

d. Jog the robot in the -z direction (of the world frame).

e. Press and hold the SHIFT key and press F5, RECORD.

20. **To select the tool frame to use,** press F5, SETIND, type the number of the tool frame you want, and press ENTER. If F5, SETIND, is not displayed, press PREV.

-OR-

You can also select the frame using the Jog Menu. Press and hold SHIFT and press COORD, move the cursor to Tool, and type the number of the frame you want to select. Refer to Section 2.3.8 for more information.

21. Jog the robot in the frame you just taught.

- If the TCP is correct, it will remain stationary during rotational moves. Go to Step 23 .

> **Caution**
>
> When you are finished setting the frame configuration, save the information to the default device so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

- If the TCP is not correct, it will not remain stationary during rotational moves. You need to review your recorded positions. If they are not correct, re-record them correctly. Go to Step 22 .

> **Warning**
>
> **When you use F4, MOVE_TO, to move the robot, unexpected motion can occur. This could injure personnel or damage equipment.**

22. **To move to a recorded position,** press and hold the SHIFT key and press F4, MOVE_TO.

23. **To save the frames and related system variables to a file** on the default device,

a. Press MENUS.

b. Select FILE.

c. Press F1, [TYPE].

d. Select File.

e. Press F5, [UTIL].

f. Select Set Device.

g. Move the cursor to the device you want and press ENTER.

h. Display the tool frame screen.

i. Press FCTN.

j. Select SAVE. This will save the frame positions and comments for all frames to the file FRAMEVAR.VR, and the frame data to SYSFRAME.SV, on the default device.

Display the SYSTEM Variables menu,

k. Press MENUS.

l. Select SYSTEM.

m. Press F1, [TYPE].

n. Select Variables.

o. Press FCTN.

p. Select SAVE. The frame positions and system variables are saved in the SYSVAR.SV file, on the default device. This is optional since frame variables are saved in SYSFRAME.SV.

3.1.2.4 Tool Frame Direct Entry Method

Procedure 3-3 Setting Up Tool Frame Using the Direct Entry Method

> **Warning**
>
> **If you set up a new frame, make sure that all frame data is zero or uninitialized before you record any positions. Press F4, CLEAR, to clear frame data.**
>
> **If you modify an existing frame, make sure that all frame data is set the way you want before you change it.**
>
> **Otherwise, you could injure personnel or damage equipment.**

> **Caution**
>
> Do not set up or alter frames when you use PalletTool. Frames are automatically set up for you by PalletTool.

Note If you have a 4-axis robot (such as an A-520i or M-410iHS/iHW), you can define a tool frame using only the direct entry method.

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Frames.
5. **To choose the motion group** for the frame you are setting up in systems with multiple motion groups press F3, [OTHER], and select the group you want. The default motion group is Group 1.

   If you have the new *i*Pendant, you can press the GROUP key and press the numeric key to switch to a specified group. Press the GROUP key and the 0 key together to toggle the sub group.

6. **If tool frames are not displayed** press F3, [OTHER], and select Tool Frame. If F3, [OTHER], is not displayed, press PREV.
7. **To display the settings for all the frames,** press PREV repeatedly until you see a screen similar to the following.

```
SETUP Frames
Tool Frame G1 / Direct Entry    1/29
X       Y        Z     Comment
1:   0.0    0.0     0.0 **************
2:   0.0    0.0     0.0 **************
3:   0.0    0.0     0.0 **************
4:   0.0    0.0     0.0 **************
5:   0.0    0.0     0.0 **************
6:   0.0    0.0     0.0 **************
7:   0.0    0.0     0.0 **************
8:   0.0    0.0     0.0 **************
9:   0.0    0.0     0.0 **************
Active TOOL $MNUTOOLNUM[1]=1
```

Note The maximum number of tool frames is set in the system variable $SCR.$MAXNUMUTOOL. Refer to Section 10.21 and to the *Software Reference Manual* for more information.

8. **To set the numerical values to zero,** move the cursor to the frame number, press F4, CLEAR, and then press F4, YES, to confirm.



Warning

Do not run a KAREL program that includes motion statements if more than one motion group is defined on your controller. If your controller is set up for more than one motion group, all motion must be initiated from a teach pendant program. Otherwise, the robot could move unexpectedly, personnel could be injured, and equipment could be damaged.

9. Press F2, DETAIL.
10. **To select a frame,**
    a. Press F3, FRAME.
    b. Type the desired frame number.
    c. Press ENTER.
11. Press F2, [METHOD].
12. Select Direct Entry. You will see a screen similar to the following.

```
SETUP Frames
Tool Frame       Direct Entry
Frame Number: 1
1  Comment:   ****************
2  X:                0.000
3  Y:                0.000
4  Z:                0.000
5  W:                0.000
6  P:                0.000
7  R:                0.000
Configuration:   N R D B, 0, 0, 0
```

13. To add a comment:

a. Move the cursor to the comment line and press ENTER.

b. Select a method of naming the comment.

c. Press the appropriate function keys to enter the comment.

d. When you are finished, press ENTER.

14. Set each position component:

a. Move the cursor to the component.

b. Enter the numeric value for the component.

c. Press the ENTER key to set the new value.

15. **To select the tool frame to use,** press F5, SETIND, type the number of the tool frame you want, and press ENTER. If F5, SETIND, is not displayed, press PREV.

-OR-

Use the Jog Menu. Press and hold SHIFT and press COORD, move the cursor to Tool, and type the number of the frame you want to select. Refer to Section 2.3.8 for more information.



Caution

When you are finished setting the frame configuration, save the information to the default device so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

16. **To save the frames and related system variables to a file** on the default device,

a. Press MENUS.

b. Select FILE.

c. Press F1, [TYPE].

d. Select File.

e. Press F5, [UTIL].

f. Select Set Device.

g. Move the cursor to the device you want and press ENTER.

h. Display the tool frame screen.

i. Press FCTN.

j. Select SAVE. This will save the frame positions and comments for all frames to the file FRAMEVAR.VR, and the frame data to SYSFRAME.SV

17. To Display the SYSTEM Variables menu (optional since data is saved in sysframe.sv),

a. Press MENUS.

b. Select SYSTEM.

c. Press F1, [TYPE].

d. Select Variables.

e. Press FCTN.

f. Select SAVE. The tool frame positions and system variables are saved in the SYSVAR.SV file, on the default device.

3.1.2.5 Selecting a Tool Frame

Procedure 3-4 Selecting a Tool Frame



Caution

Do not set up or alter frames when you use PalletTool. Frames are automatically set up for you by PalletTool.

Note You can also use the Jog Menu to select the number of the tool frame you want to use. Refer to Section 2.3.8 .

Conditions

- The tool frame you want to select has been set up.

Steps

1. Press MENUS.

2. Select SETUP.
3. Press F1, [TYPE].
4. Select Frames.
5. **To choose the motion group** for the frame you are setting up in systems with multiple motion groups press F3, [OTHER], and select the group you want. The default motion group is Group 1.

   If you have the new *i*Pendant, you can press the GROUP key and press the numeric key to switch to a specified group. Press the GROUP key and the 0 key together to toggle the sub group.

> **Warning**
>
> **Do not run a KAREL program that includes motion statements if more than one motion group is defined on your controller. If your controller is set up for more than one motion group, all motion must be initiated from a teach pendant program. Otherwise, the robot could move unexpectedly, personnel could be injured, and equipment could be damaged.**

6. **If tool frames are not displayed**, press F3, [OTHER], and select Tool Frame. If F3, [OTHER], is not displayed, press PREV. You will see a screen similar to the following.

```
SETUP Frames
Tool Frame Setup / Direct Entry     1/10
X       Y       Z      Comment
0.0     0.0     0.0 *********
****
2:    0.0     0.0      0.0 *************
3:    0.0     0.0      0.0 *************
4:    0.0     0.0      0.0 *************
5:    0.0     0.0      0.0 *************
6:    0.0     0.0      0.0 *************
7:    0.0     0.0      0.0 *************
8:    0.0     0.0      0.0 *************
9:    0.0     0.0      0.0 *************
Active TOOL $MNUTOOLNUM[1]=1
```

Note The maximum number of tool frames is set in the system variable $SCR.$MAXNUMUTOOL. Refer to the *Software Reference Manual* for more information.

7. **To select the tool frame to use,** press F5, SETIND, type the number of the tool frame you want, and press ENTER. If F5, SETIND, is not displayed, press PREV.

   -OR-

Use the Jog Menu. Press and hold SHIFT and press COORD, move the cursor to Tool, and type the number of the frame you want to select. Refer to Section 2.3.8 for more information.

8. When a position is recorded in the teach pendant program, the value of the position's tool frame will always equal the value of $MNUTOOLNUM[group_no] at the time the position was recorded.

When a teach pendant program is executed, you must make sure that the tool frame of the position equals the value of $MNUTOOLNUM [group_no], otherwise, an error will occur. Set the value of $MNUTOOLNUM using the UTOOL_NUM=n instruction in the teach pendant program before you record the position to guarantee that the tool frame numbers match during program execution.

Refer to Section 6.14 for more information on the UTOOL_NUM instruction.

9. To use a tool frame in a KAREL program, set $GROUP[group_no].$UTOOL=$MNUTOOL[group_no,$MNUTOOL NUM[group_no]] before executing any motion.

3.1.3 User Frame

3.1.3.1 Setting Up User Frame

User frame is a frame that you can set up in any location, with any orientation. User frames are used so that positions in a program can be recorded relative to the origin of the frame. All positions in a program are automatically recorded in user frame. If you do not set up the location and orientation of the user frame before you create a program, the user frame will be set by default to the world frame in the program.

Note If you are using the remote TCP option, you must define a user frame to be your remote TCP (RTCP) frame. Refer to the "Remote TCP Frame" section.



Caution

Recorded positions and position registers are affected by MNUFRAME, and MNUFRAME has an affect during playback. If you change MNUFRAME, any recorded positions and position registers will also change.

Enabling $USEUFRAME

The system variable $USEUFRAME defines whether the current value of $MNUFRAMENUM[group_no] will be assigned to the position's user frame when it is recorded or touched up.

Note Changing the user frame number in one group does not change the user frame number of other groups.

Caution

When $USEUFRAME=FALSE, the user frame number is equal to 0 when you initially record positions and touch them up, regardless of the value of $MNUFRAMENUM[group_no].

When $USEUFRAME=TRUE, the position's user frame number is equal to the user frame defined by $MNUFRAMENUM[group_no] when you initially record positions and touch them up. You must also touch up positions with the position's user frame equal to the user frame defined by $MNUFRAMENUM[group_no].

Be sure to set the system variable $USEUFRAME properly. Otherwise, your program will not operate properly.

After you set up the user frame, you can change its location and orientation. Before you teach positions in a frame that is to move, change $USEUFRAME to TRUE and record the positions. All positions in a program recorded relative to that frame change with it. All positions are taught relative to the default user frame. Use the UFRAME_NUM program instruction to change the user frame.

- You can set up as many as ten user frames for each robot. They will be stored in the system variable $MNUFRAME.
- You can select one user frame per robot group to be active at a time. The frame number will be stored in $MNUFRAMENUM.
- You can jog the robot in user frame.

Caution

Each time you create a program, set the current user frame number to a value between 1 and 9 (Procedure 3-8). Do this even if you do not plan to use a user frame in the program, or if you want the user frame position to be zero (0,0,0,0,0,0). Otherwise, if the current user frame number is zero, a user frame set in that program will not work.

See Figure 3–5 .

Table 3–5. User Frame Setup Screen Items

ITEM	DESCRIPTION
User frame setup/xxxx	This line shows the current selected method for setting each user frame.
Frame number 1-9	These lines show the current position and comment for each available user frame.

Table 3–5. User Frame Setup Screen Items (Cont'd)

X	This column shows the X coordinate of each user frame.
Y	This column shows the Y coordinate of each user frame.
Z	This column shows the Z coordinate of each user frame.
Comment	This column shows the comment for each user frame.
Active UFRAME $MNUFRAMENUM[1] = 0	This line indicates which user frame is currently selected for use.
[TYPE]	Press this key to access various application-specific options.
DETAIL	Press this key to display detailed information for each user frame and to set the definition and comment of each frame.
OTHER	Press this key to select one of the other available reference frames or to choose the motion group for the frame you are setting up (in systems with multiple motion groups).
CLEAR	To set the numerical values of any user frame to zero, move the cursor to the frame number, press CLEAR and answer YES to the prompt.
SETIND	To select the user frame to use, press F5, SETIND, type the number of the user frame you want, and press ENTER. This sets the active user frame ($MNUFRAMNUM[1]) to the number of the frame you want. If F5, SETIND, is not displayed, press PREV.
CLRIND	Press this key if you want to clear the number of the user frame currently selected for use.

Figure 3–5. World and User Frames

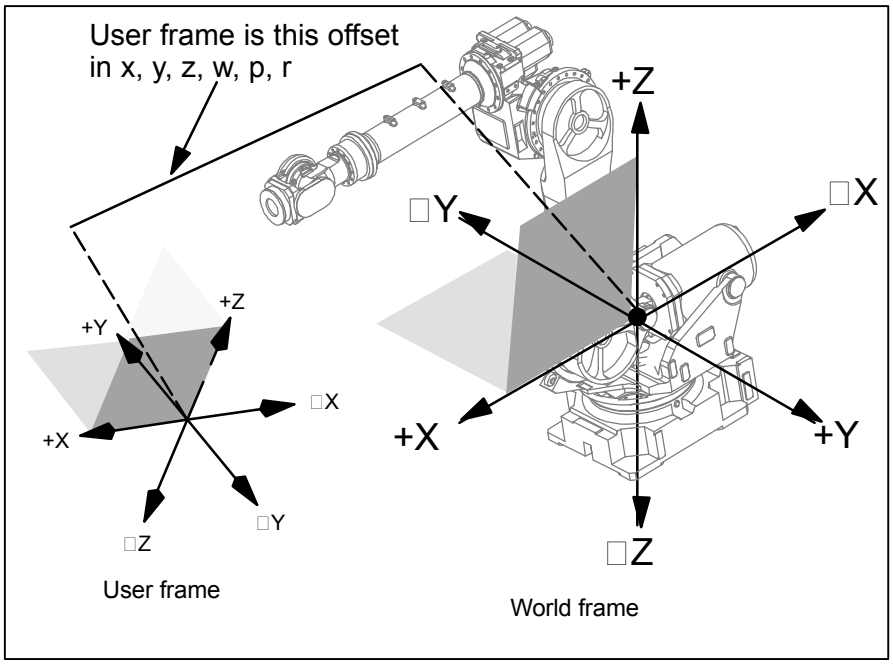
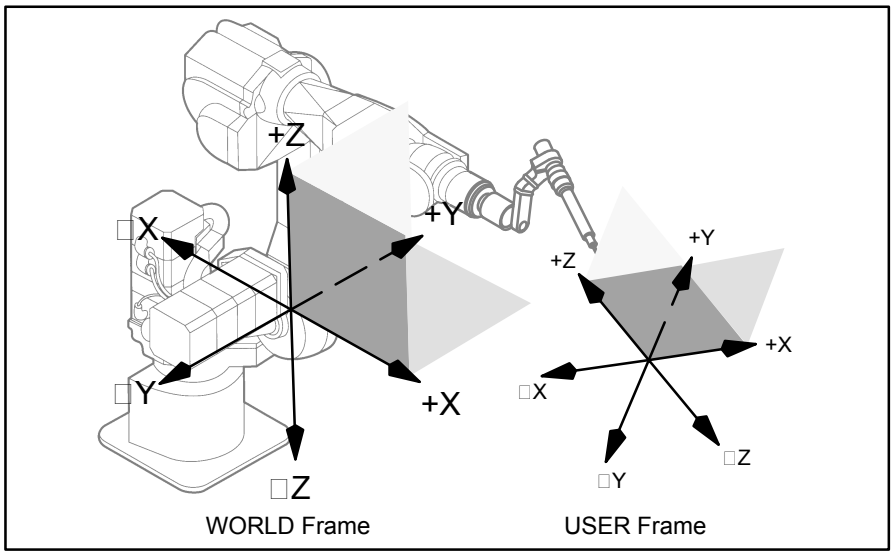


Figure 3–6. World and User Frames for ArcTool



You can use three methods to define the user frame:

- Three point method
- Four point method
- Direct entry method

Three Point Method

Recording three points defines the user frame. The three points are the origin, a position along the +x-axis of the user frame, and a position on the x-y plane of the user frame (defines the x-y plane and the y-z plane).

Table 3–6. User Frame Setup Three Point DETAIL Screen Items

ITEM	DESCRIPTION
Frame number	This line shows the number of the user frame being defined.
X, Y, Z, W, P, R	These items show the current coordinates of the user frame being defined.
Comment	This item is used to add a comment.
Orient Origin Point	This item is used to define the reference point of the user frame.
X Direction Point	This item is used to define the +x direction point.
Y Direction Point	This item is used to define a point on the X-Y plane.

Table 3–6. User Frame Setup Three Point DETAIL Screen Items (Cont'd)

Active UFRAME $MNUFRAMENUM[1] = 0	This line indicates which user frame is currently selected for use.
[TYPE]	Press this key to access various application-specific options.
[METHOD]	Press this key to select a method of entry.
[FRAME]	Press this key to select a frame.
MOVE_TO	Press this key to move to a recorded position. Move the cursor to the desired point and press MOVE_TO along with the SHIFT key.
RECORD	Press this key along with the SHIFT key to record a position.

Four Point Method

Use the four point method when you need to define a frame that has its origin at a position other than the reference of the frame. You can also use it to define multiple frames with parallel axes. The four points are the reference of the frame (called orient origin point), a point along the +x-axis of the frame (defines the x-z plane), a point on the x-y plane of the frame (defines the x-y plane and the y-z plane) and the origin of the frame (called system origin).

Table 3–7. User Frame Setup Four Point DETAIL Screen Items

ITEM	DESCRIPTION
Frame number	This line shows the number of the user frame being defined.
X, Y, Z, W, P, R	These items show the current coordinates of the user frame being defined.
Comment	This item is used to add a comment.
Orient Origin Point	This item is used to define the reference point of the user frame.
X Direction Point	This item is used to define the +x direction point.
Y Direction Point	This item is used to define a point on the X-Y plane.
System Origin	This item is used to teach the origin of the second user frame.
Active UFRAME $MUFRAMENUM[1] = 0	This line indicates which user frame is currently selected for use.
[TYPE]	Press this key to access various application-specific options.
[METHOD]	Press this key to select a method of entry.
[FRAME]	Press this key to select a frame.

Table 3–7. User Frame Setup Four Point DETAIL Screen Items (Cont'd)

MOVE_TO	Press this key to move to a recorded position. Move the cursor to the desired point and press MOVE_TO along with the SHIFT key.
RECORD	Press this key along with the SHIFT key to record a position.

Direct Entry Method

Use the direct entry method when you know the coordinates of the user frame. The direct entry method allows you to designate the origin with values for x, y, z, w, p, and r. Use Procedure 3-7 to define a user frame using the direct entry method.

Table 3–8. User Frame Setup Direct Entry DETAIL Screen Items

ITEM	DESCRIPTION
Frame number	This line shows the number of the user frame being defined.
Comment	This item is used to add a comment.
X	This item is used to define the component of the user frame.
Y	This item is used to define the component of the user frame.
Z	This item is used to define the component of the user frame.
W	This item is used to define the component of the user frame.
P	This item is used to define the component of the user frame.
R	This item is used to define the component of the user frame.
Configuration	This line displays the current robot configuration.
Active UFRAME $MNUFRAMENUM[1] =0	This line indicates which user frame is currently selected for use.
[TYPE]	Press this key to access various application-specific options.
[METHOD]	Press this key to select a method of entry.
[FRAME]	Press this key to select a frame.
MOVE_TO	Press this key along with the SHIFT key to move to a recorded position.
RECORD	Press this key along with the SHIFT key to record a position.

Use Procedure 3-8 to select a user frame.



Caution

Do not set up or alter frames when you use PalletTool. Frames are automatically set up for you by PalletTool.

3.1.3.2 HandlingTool User Frame Three Point Method

Procedure 3-5 Setting Up the User Frame Using the Three Point Method

> **Caution**
>
> Do not set up or alter frames when you use PalletTool. Frames are automatically set up for you by PalletTool.

Conditions

- The tool frame you want to select has been set up. (Procedure 3-1 , Procedure 3-2 , or Procedure 3-3)

> **Warning**
>
> **If you set up a new frame, make sure that all frame data is zero or uninitialized before you record any positions. Press F4, CLEAR, to clear frame data.**
>
> **If you modify an existing frame, make sure that all frame data is set the way you want before you change it.**
>
> **Otherwise, you could injure personnel or damage equipment.**

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Frames.
5. **To choose the motion group** for the frame you are setting up in systems with multiple motion groups press F3, [OTHER], and select the group you want. The default motion group is Group 1.

   If you have the new *i*Pendant, you can press the GROUP key and press the numeric key to switch to a specified group. Press the GROUP key and the 0 key together to toggle the sub group.

Warning

Do not run a KAREL program that includes motion statements if more than one motion group is defined on your controller. If your controller is set up for more than one motion group, all motion must be initiated from a teach pendant program. Otherwise, the robot could move unexpectedly, personnel could be injured, and equipment could be damaged.

6. **If user frames are not displayed**, press F3, [OTHER], and select User/RTCP. If F3, [OTHER], is not displayed, press PREV.
7. **To display the settings for all frames**, press PREV repeatedly until you see a screen similar to the following.

```
SETUP Frames
User/RTCP Setup / Three Point
X       Y       Z     Comment
1:    0.0     0.0     0.0 *************
2:    0.0     0.0     0.0 *************
3:    0.0     0.0     0.0 *************
4:    0.0     0.0     0.0 *************
5:    0.0     0.0     0.0 *************
6:    0.0     0.0     0.0 *************
7:    0.0     0.0     0.0 *************
8:    0.0     0.0     0.0 *************
9:    0.0     0.0     0.0 *************
Active UFRAME/RTCP $MNUFRAMNUM[1]=0
```

8. **To set the numerical values to zero**, move the cursor to the frame number, press F4, CLEAR, and then press F4, YES, to confirm.
9. Press F2, DETAIL.
10. **To select a frame**,
    a. Press F3, FRAME.
    b. Type the desired frame number.
    c. Press ENTER.
11. Press F2, [METHOD].
12. Select Three Point. You will see a screen similar to the following.

```
SETUP Frames
User/RTCP Setup/ Three Point
Frame Number: 2
X:  0.0     Y:  0.0     Z:  0.0
W:  0.0     P:  0.0     R:  0.0
Comment: ***************
Orient Origin Point:      UNINIT
X Direction Point:        UNINIT
Y Direction Point:        UNINIT
Active UFRAME/RTCP $MNUFRAMNUM[1]=1
```

13. To add a comment:

a. Move the cursor to the comment line and press ENTER.

b. Select a method of naming the comment.

c. Press the appropriate function keys to enter the comment.

d. When you are finished, press ENTER.

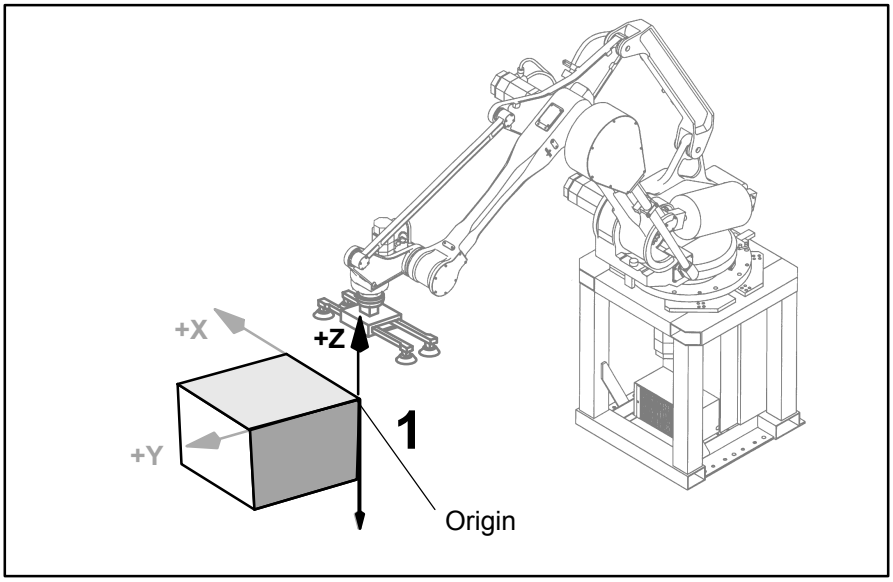
14. Define the origin point of the user frame: **(Orient Origin Point)** :

a. Move the cursor to Orient Origin Point.

b. Jog the robot TCP to the origin. In Figure 3–7 , the origin is labeled 1.

c. Press and hold the SHIFT key and press F5, RECORD.

Figure 3–7. Defining the Origin



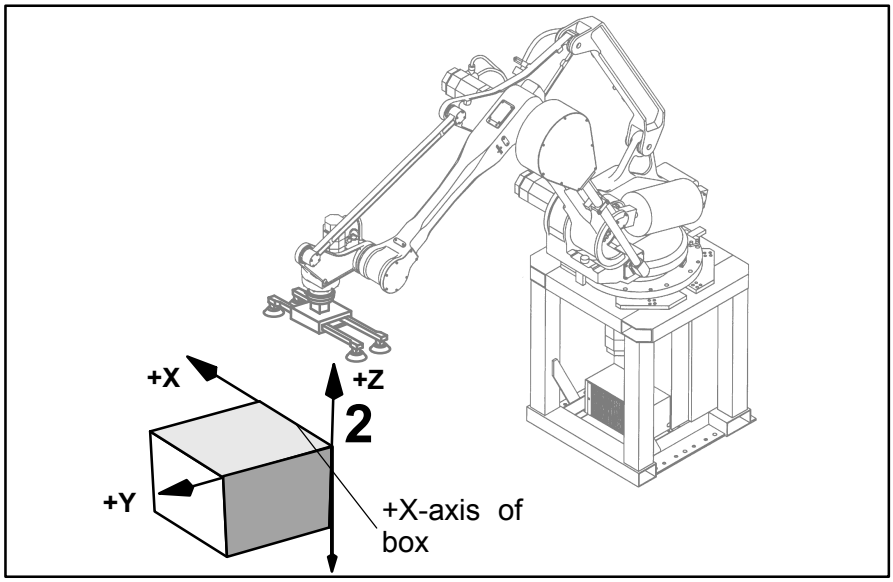
15. Define the +x direction point **(X Direction Point)** :

a. Move the cursor to X Direction Point.

b. Jog the robot tool tip to a point along the +x-axis. In Figure 3–8 , this point is labeled number 2.

c. Press and hold the SHIFT key and press F5, RECORD.

Figure 3–8. Defining the X Direction Point



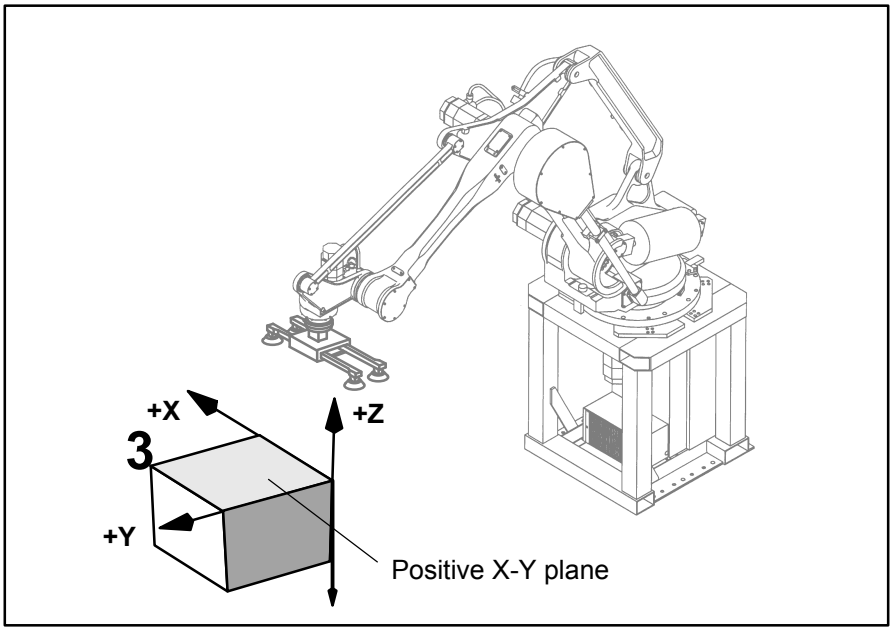
16. Define a point on the positive X-Y plane **(Y Direction Point)** :

a. Move the cursor to Y Direction Point.

b. Jog the robot to a location on the positive X-Y plane. In Figure 3–9 , this point is labeled number 3.

c. Press and hold the SHIFT key and press F5, RECORD.

Figure 3–9. Defining the X-Y Plane



17. **To select the user frame to use,** press F5, SETIND, type the number of the user frame you want, and press ENTER. This sets the active user frame ($MNUFRAMNUM[1]) to the number of the frame you want. If F5, SETIND, is not displayed, press PREV.

-OR-

Use the Jog Menu. Press and hold SHIFT and press COORD, move the cursor to User, and type the number of the frame you want to use. Refer to Section 2.3.8 for more information.

18. Jog the robot in the +x, +y, and +z directions. The robot should move in the correct directions according to the frame you defined. If the robot does not move in the correct directions, go to Step 19 . Otherwise, go to Step 20 .

> **Caution**
>
> When you are finished setting the frame configuration, save the information to the default device so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

19. **To move to a recorded position,** move the cursor to the desired position, press and hold the SHIFT key and press F4, MOVE_TO.

Warning

When you use F4, MOVE_TO, to jog the robot, unexpected motion can occur. This could injure personnel or damage equipment.

Note If you want to clear the current user frame selected, press NEXT, >, and then F2, CLRIND. This sets the active user frame ($MNUFRAMNUM[1]) to zero, which means that the default user frame is currently selected.

20. To save the frames and related system variables to a file on the default device,

a. Press MENUS.

b. Select FILE.

c. Press F1, [TYPE].

d. Select File.

e. Press F5, [UTIL].

f. Select Set Device.

g. Move the cursor to the device you want and press ENTER.

h. Display the user frame screen.

i. Press FCTN.

j. Select SAVE. This will save the frame positions and comments for all frames to the file FRAMEVAR.VR, and the frame data to SYSFRAME.SV, on the default device.

Display the SYSTEM Variables menu,

k. Press MENUS.

l. Select SYSTEM.

m. Press F1, [TYPE].

n. Select Variables.

o. Press FCTN.

p. Select SAVE. The frame positions and system variables are saved in the SYSVAR.SV file, on the default device. This is optional since frame variables are saved in SYSFRAME.SV.

3.1.3.3 HandlingTool User Frame Four Point Method

Procedure 3-6 Setting Up User Frame Using the Four Point Method

> **Caution**
>
> Do not set up or alter frames when you use PalletTool. Frames are automatically set up for you by PalletTool.

> **Warning**
>
> **If you set up a new frame, make sure that all frame data is zero or uninitialized before you record any positions. Press F4, CLEAR, to clear frame data.**
>
> **If you modify an existing frame, make sure that all frame data is set the way you want before you change it.**
>
> **Otherwise, you could injure personnel or damage equipment.**

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Frames.
5. **To choose the motion group** for the frame you are setting up in systems with multiple motion groups press F3, [OTHER], and select the group you want. The default motion group is Group 1.

   If you have the new *i*Pendant, you can press the GROUP key and press the numeric key to switch to a specified group. Press the GROUP key and the 0 key together to toggle the sub group.

   > **Warning**
   >
   > **Do not run a KAREL program that includes motion statements if more than one motion group is defined on your controller. If your controller is set up for more than one motion group, all motion must be initiated from a teach pendant program. Otherwise, the robot could move unexpectedly, personnel could be injured, and equipment could be damaged.**

6. **If user frames are not displayed,** press F3, [OTHER], and select User Frame. If F3, [OTHER], is not displayed, press PREV.

7. **To display the settings for all frames,** press PREV repeatedly until you see a screen similar to the following.

```
SETUP Frames
User/RTCP Setup / Four Point
X       Y       Z     Comment
1:    0.0     0.0     0.0 *************
2:    0.0     0.0     0.0 *************
3:    0.0     0.0     0.0 *************
4:    0.0     0.0     0.0 *************
5:    0.0     0.0     0.0 *************
6:    0.0     0.0     0.0 *************
7:    0.0     0.0     0.0 *************
8:    0.0     0.0     0.0 *************
9:    0.0     0.0     0.0 *************
Active UFRAME/RTCP $MNUFRAMNUM[1]=0
```

8. **To set the numerical values to zero**, move the cursor to the frame number, press F4, CLEAR, and then press F4, YES, to confirm.
9. Press F2, DETAIL.
10. **To select a frame**,
    a. Press F3, FRAME.
    b. Type the desired frame number.
    c. Press ENTER.
11. Press F2, [METHOD].
12. Select Four Point. You will see a screen similar to the following.

```
SETUP Frames
User/RTCP Setup/ Four Point
Frame Number: 2
X:  0.0      Y:  0.0      Z:  0.0
W:  0.0      P:  0.0      R:  0.0
Comment: ***************
Orient Origin Point:      UNINIT
X Direction Point:        UNINIT
Y Direction Point:        UNINIT
System Origin:            UNINIT
Active UFRAME/RTCP $MNUFRAMNUM[1]=1
```

13. To add a comment:

a. Move the cursor to the comment line and press ENTER.

b. Select a method of naming the comment.

c. Press the appropriate function keys to enter the comment.

d. When you are finished, press ENTER.

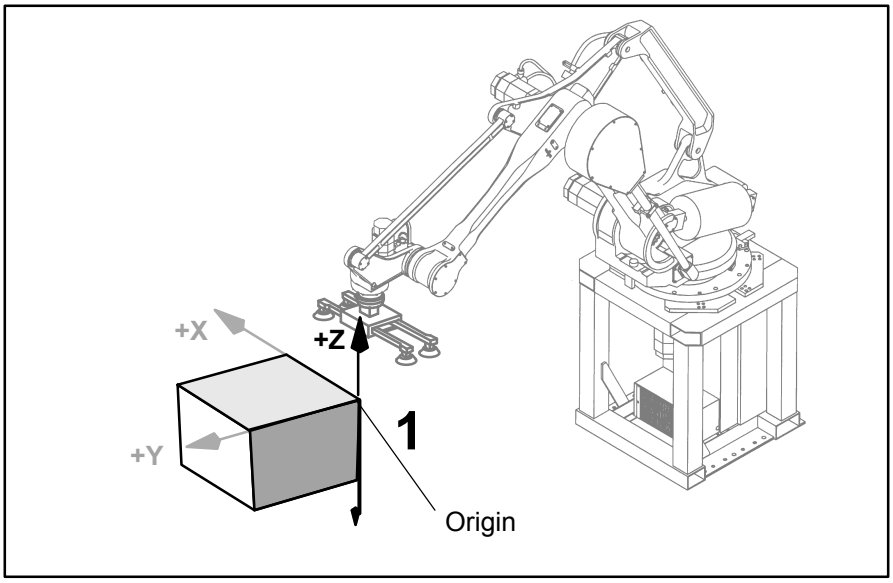
14. Define the reference point of the user frame **(Orient Origin Point)** :

a. Move the cursor to Orient Origin Point.

b. Jog the robot TCP to the origin. In Figure 3–10 , the origin is labeled 1.

c. Press and hold the SHIFT key and press F5, RECORD.

Figure 3–10. Defining the Origin



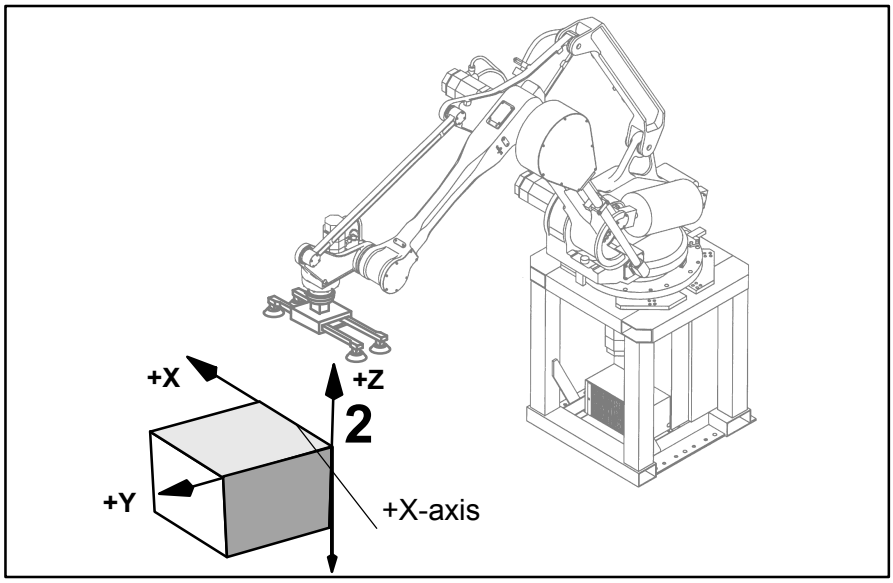
15. Define the +x direction point **(X Direction Point)** :

a. Move the cursor to X Direction Point.

b. Jog the robot TCP to a point along the +x-axis. In Figure 3–11 , this point is labeled number 2.

c. Press and hold the SHIFT key and press F5, RECORD, to record a position.

Figure 3–11. Defining the X Direction Point



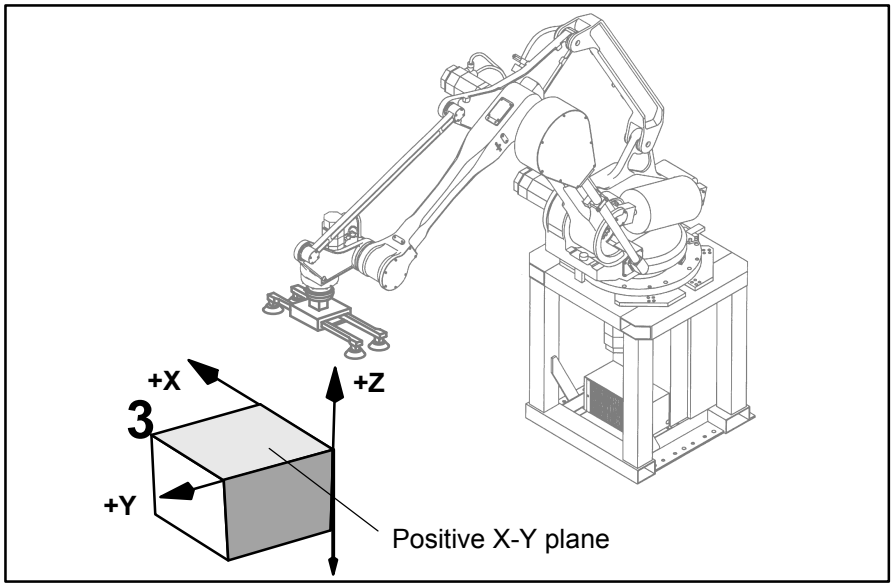
16. Define a point on the X-Y plane **(Y Direction Point)**:

a. Move the cursor to Y Direction Point.

b. Jog the robot to a location on the positive X-Y plane. In Figure 3–12 this point is labeled number 3.

c. Press and hold the SHIFT key and press F5, RECORD.

Figure 3–12. Defining the X-Y Plane



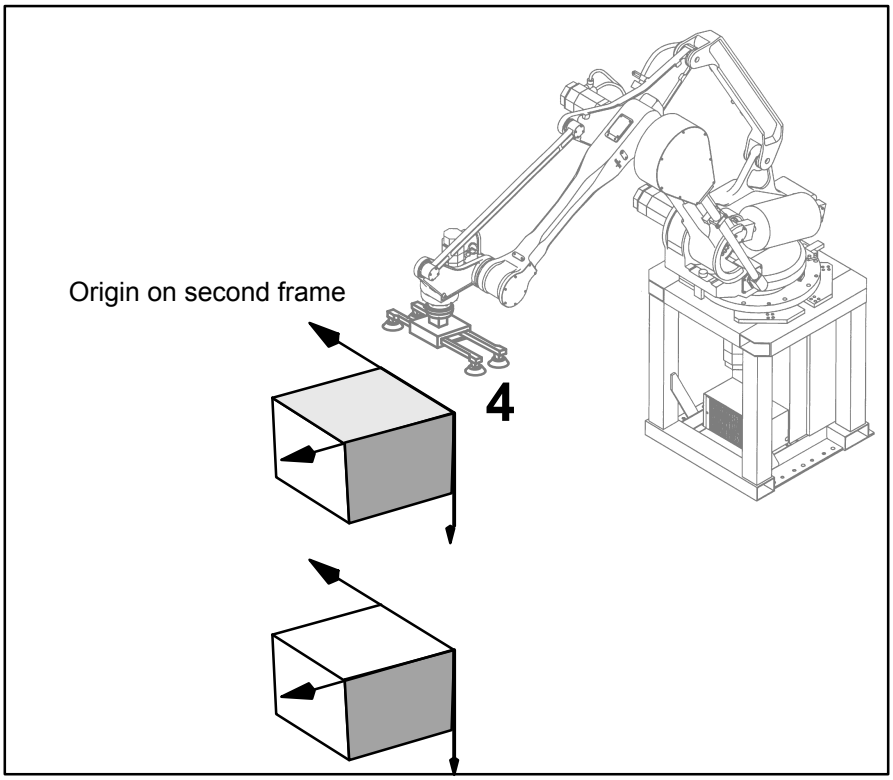
17. Teach the origin of the second user frame **(System Origin)**:

a. Move the cursor to System Origin.

b. Jog the robot TCP to the origin of the second user frame. In Figure 3–13 , the origin is labeled 4.

c. Press F5, RECORD, to record a position.

Figure 3–13. Defining the Second Origin



18. **To select the user frame to use,** press F5, SETIND, type the number of the user frame you want, and press ENTER. This sets the active user frame ($MNUFRAMNUM[1]) to the number of the frame you want. If F5, SETIND, is not displayed, press PREV.

-OR-

Use the Jog Menu. Press and hold SHIFT and press COORD, move the cursor to User, and type the number of the frame you want to use. Refer to Section 2.3.8 for more information.

19. Jog the robot in the +x, +y, and +z directions. The robot should move in the correct directions according to the frame you defined. If the robot does not move in the correct directions, go to Step 20 . Otherwise, go to Step 21 .

Caution

When you are finished setting the frame configuration, save the information to the default device so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

20. **To move to a recorded position**, move the cursor to the desired position, press and hold the SHIFT key and press F4, MOVE_TO.

Warning

When you use F4, MOVE_TO, to jog the robot, unexpected motion can occur. This could injure personnel or damage equipment.

Note If you want to clear the current frame to zero, move the cursor to the frame number and press NEXT, >, and then F2, CLRIND. This sets the active user frame ($MNUFRAMNUM[1]) to zero, which means that the default user frame is currently selected.

21. **To save the frames and related system variables to a file** on the default device,

a. Press MENUS.
b. Select FILE.
c. Press F1, [TYPE].
d. Select File.
e. Press F5, [UTIL].
f. Select Set Device.
g. Move the cursor to the device you want and press ENTER.
h. Display the user frame screen.
i. Press FCTN.
j. Select SAVE. This will save the frame positions and comments for all frames to the file FRAMEVAR.VR, and the frame data to SYSFRAME.SV, on the default device.

Display the SYSTEM Variables menu,

k. Press MENUS.
l. Select SYSTEM.
m. Press F1, [TYPE].
n. Select Variables.
o. Press FCTN.

p. Select SAVE. The frame positions and system variables are saved in the SYSVAR.SV file, on the default device. This is optional since frame variables are saved in SYSFRAME.SV.

3.1.3.4 User Frame Direct Entry Method

Procedure 3-7 Setting Up User Frame Using the Direct Entry Method

> **Caution**
>
> Do not set up or alter frames when you use PalletTool. Frames are automatically set up for you by PalletTool.

> **Warning**
>
> **If you set up a new frame, make sure that all frame data is zero or uninitialized before you record any positions. Press F4, CLEAR, to clear frame data.**
>
> **If you modify an existing frame, make sure that all frame data is set the way you want before you change it.**
>
> **Otherwise, you could injure personnel or damage equipment.**

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Frames.
5. **To choose the motion group** for the frame you are setting up in systems with multiple motion groups press F3, [OTHER], and select the group you want. The default motion group is Group 1.

   If you have the new *i*Pendant, you can press the GROUP key and press the numeric key to switch to a specified group. Press the GROUP key and the 0 key together to toggle the sub group.

Warning

Do not run a KAREL program that includes motion statements if more than one motion group is defined on your controller. If your controller is set up for more than one motion group, all motion must be initiated from a teach pendant program. Otherwise, the robot could move unexpectedly, personnel could be injured, and equipment could be damaged.

6. **If user frames are not displayed**, press F3, [OTHER], and select User Frame. If F3, [OTHER], is not displayed, press PREV.
7. **To display the settings for all the frames**, press PREV repeatedly until you see a screen similar to the following.

```
SETUP Frames
User/RTCP Setup / Direct Entry
X       Y       Z     Comment
1:    0.0     0.0     0.0 *************
2:    0.0     0.0     0.0 *************
3:    0.0     0.0     0.0 *************
4:    0.0     0.0     0.0 *************
5:    0.0     0.0     0.0 *************
6:    0.0     0.0     0.0 *************
7:    0.0     0.0     0.0 *************
8:    0.0     0.0     0.0 *************
9:    0.0     0.0     0.0 *************
Active UFRAME/RTCP $MNUFRAMNUM[1]=0
```

8. **To set the numerical values to zero**, move the cursor to the frame number, press F4, CLEAR, and then press F4, YES, to confirm.
9. Press F2, DETAIL.
10. **To select a frame**,
    a. Press F3, FRAME.
    b. Type the desired frame number.
    c. Press ENTER.
11. Press F2, [METHOD].
12. Select Direct Entry. You will see a screen similar to the following.

```
SETUP Frames
User/RTCP Setup/ Direct Entry
Frame Number: 1
1 Comment: ****************
2 X:                0.000
3 Y:                0.000
4 Z:                0.000
5 W:                0.000
6 P:                0.000
7 R:                0.000
Configuration:  N, 0, 0, 0
Active UFRAME/RTCP $MNUFRAMENUM[1]=0
```

13. To add a comment:

a. Move the cursor to the comment line and press ENTER.

b. Select a method of naming the comment.

c. Press the appropriate function keys to enter the comment.

d. When you are finished, press ENTER.

14. Set each position component:

a. Move the cursor to the component.

b. Enter the numeric value for the component.

c. Press the ENTER key to set the new value.

15. **To select the user frame to use**, press F5, SETIND, type the number of the user frame you want, and press ENTER. This sets the active user frame ($MNUFRAMNUM[1]) to the number of the frame you want. If F5, SETIND, is not displayed, press PREV.

-OR-

Use the Jog Menu. Press and hold SHIFT and press COORD, move the cursor to User, and type the number of the frame you want to use. Refer to Section 2.3.8 for more information.

Note If you want to clear the current frame to zero, move the cursor to the frame number and press NEXT, >, and then F2, CLRIND. This sets the active user frame ($MNUFRAMNUM[1]) to zero, which means that the default user frame is currently selected.

Caution

When you are finished setting the frame configuration, save the information to the default device so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

16. **To save the frames and related system variables to a file** on the default device,
    a. Press MENUS.
    b. Select FILE.
    c. Press F1, [TYPE].
    d. Select File.
    e. Press F5, [UTIL].
    f. Select Set Device.
    g. Move the cursor to the device you want and press ENTER.
    h. Display the user frame screen.
    i. Press FCTN.
    j. Select SAVE. This will save the frame positions and comments for all frames to the file FRAMEVAR.VR, and the frame data to SYSFRAME.SV, on the default device.
17. To save the SYSTEM Variables menu,
    a. Press MENUS.
    b. Select SYSTEM.
    c. Press F1, [TYPE].
    d. Select Variables.
    e. Press FCTN.
    f. Select SAVE. The frame positions and system variables are saved in the SYSVAR.SV file, on the default device. This is optional since frame variables are saved in SYSFRAME.SV.

3.1.3.5 Selecting a User Frame

Procedure 3-8 Selecting a User Frame

Caution

The system variable $USEUFRAME defines whether the current value of $MNUFRAMENUM[group_no] will be assigned to the position's user frame when it is being recorded or touched up.

When $USEUFRAME=FALSE, the initial recording of positions and the touching up of positions is done with the user frame number equal to 0, regardless of the value of $MNUFRAMENUM[group_no].

When $USEUFRAME=TRUE, the initial recording of positions is done with the position's user frame equal to the user frame defined by $MNUFRAMENUM[group_no]. The touching up of positions must also be done with the position's user frame equal to the user frame defined by $MNUFRAMENUM[group_no].

Be sure to set system variable $USEUFRAME correctly. Otherwise, your program will not operate properly.

Note You can also use the Jog Menu to select the number of the user frame you want to use. Refer to Section 2.3.8 .

Conditions

- The user frame you want to select has been set up.

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Frames.
5. If user frames are not displayed press F3, [OTHER], and select User Frame. If F3, [OTHER], is not displayed, press PREV. You will see a screen similar to the following.

```
SETUP Frames
User/RTCP Setup / Direct Entry
X       Y        Z      Comment
1:    0.0    0.0      0.0 *************
2:    0.0    0.0      0.0 *************
3:    0.0    0.0      0.0 *************
4:    0.0    0.0      0.0 *************
5:    0.0    0.0      0.0 *************
6:    0.0    0.0      0.0 *************
7:    0.0    0.0      0.0 *************
8:    0.0    0.0      0.0 *************
9:    0.0    0.0      0.0 *************
Active UFRAME/RTCP $MNUFRAMNUM[1]=0
```

6. **To select the user frame to use**, press F5, SETIND, type the number of the user frame you want, and press ENTER. This sets the active user frame ($MNUFRAMNUM[1]) to the number of the frame you want. If F5, SETIND, is not displayed, press PREV.

   -OR-

   Use the Jog Menu. Press and hold SHIFT and press COORD, move the cursor to User, and type the number of the frame you want to use. Refer to Section 2.3.8 for more information.

   Note When a teach pendant program is executed, you must make sure that the user frame of the position equals the value of $MNUFRAMENUM[group_no], otherwise, an error will occur. Set the value of $MNUFRAMENUM[1] using the UFRAME_NUM=n instruction in the teach pendant program and then execute that instruction before you record the position. This guarantees that the position corresponds to the correct user frame.

3.1.4 Remote TCP Frame

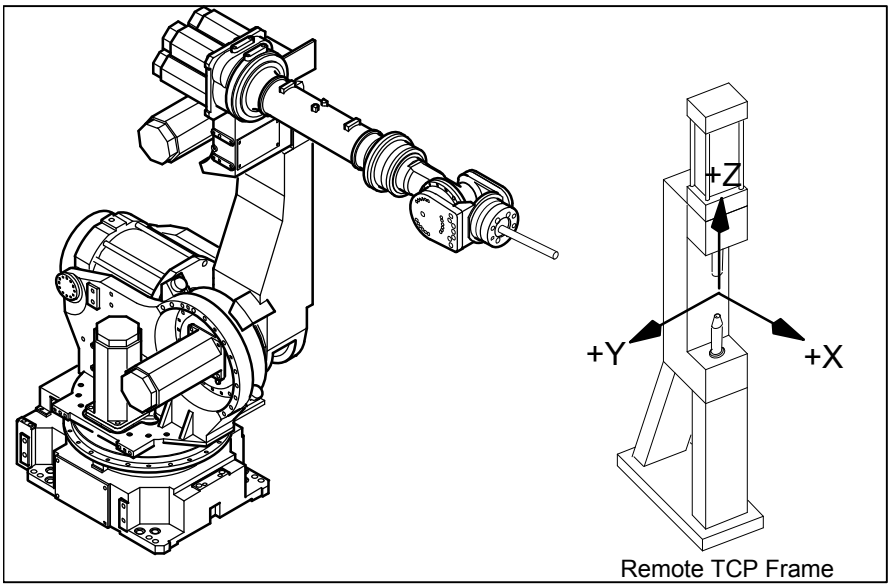
3.1.4.1 Setting Up a Remote TCP Frame

You must define a remote TCP (RTCP) frame before you can use the remote TCP option to jog or include remote TCP within a motion instruction. Use Procedure 3-10 to set up an RTCP frame. You define this frame using the location of the remote TCP as the origin of the frame.

Note Remote TCP frame will not be available if the Coordinated Motion option is loaded.

See Figure 3–14 for an example of a remote TCP frame.

Figure 3–14. Remote TCP Frame



You can use two methods to define the RTCP frame:

- Three point method
- Direct entry method

Three Point Method

Recording three points defines the RTCP frame. The three points are the origin, or TCP of the fixed tool, a position along the +x-axis of the RTCP frame, and a position on the y-axis of the RTCP frame. Use Procedure 3-9 to define an RTCP frame using the three point method.

Direct Entry Method

If you cannot use the three point method, use the direct entry method. In the direct entry method, you will be required to specify values for x, y, and z of the remote TCP. Use Procedure 3-10 to define an RTCP frame using the direct entry method.

Use Procedure 3-11 to select an RTCP frame.

Procedure 3-9 Setting Up a Remote TCP Frame Using the Three Point Method

Warning

If you are setting up a new frame, make sure that all frame data is zero or uninitialized before you record any positions. Press F4, CLEAR, to clear frame data.

If you are modifying an existing frame, make sure that all frame data is set the way you want before you change it.

Otherwise, you could injure personnel or damage equipment.

Conditions

- The end-of-arm tooling or a setup pointer is attached to the robot faceplate.
- You have set up the tool frame for the end-of-arm tooling or setup pointer. Refer to Section 3.1.2.1 .

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Frames.
5. **If user/RTCP frames are not displayed**, press F3, [OTHER], and select User/RTCP. If F3, [OTHER], is not displayed, press PREV.
6. **To display the settings for all frames**, press PREV repeatedly until you see a screen similar to the following.

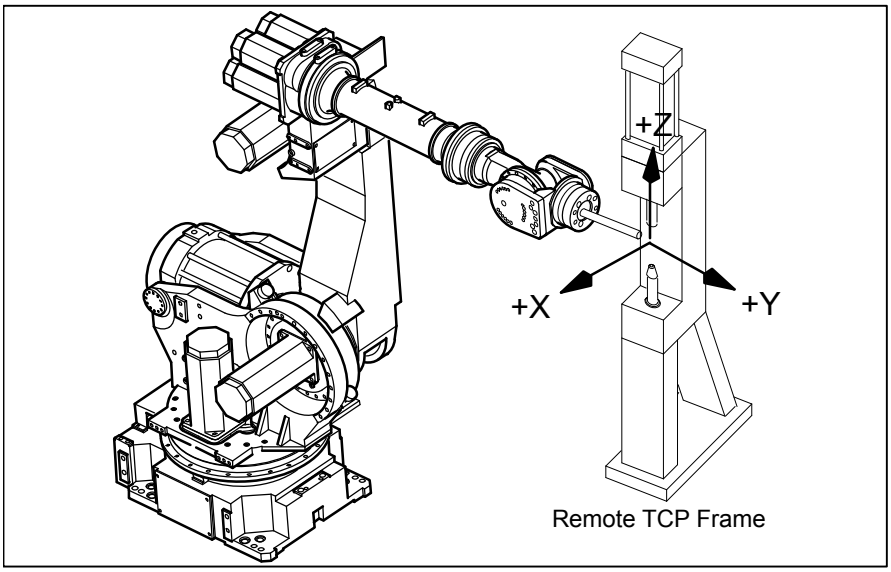
```
SETUP Frames
User/RTCP Setup / Three Point
X       Y       Z     Comment
1:    0.0    0.0     0.0 *************
2:    0.0    0.0     0.0 *************
3:    0.0    0.0     0.0 *************
4:    0.0    0.0     0.0 *************
5:    0.0    0.0     0.0 *************
6:    0.0    0.0     0.0 *************
Active UFRAME/RTCP $MNUFRAMNUM[1]=0
```

7. **To set the numerical values to zero**, move the cursor to the frame number, press F4, CLEAR, and then press F4, YES, to confirm.
8. Press F2, DETAIL.
9. **To select a frame**,
   a. Press F3, FRAME.
   b. Type the desired frame number.
   c. Press ENTER.
10. Press F2, [METHOD].
11. Select Three Point. You will see a screen similar to the following.

```
SETUP Frames
User/RTCP Setup/ Three Point
Frame Number: 2
X:  0.0     Y:  0.0     Z:  0.0
W:  0.0     P:  0.0     R:  0.0
Comment: ***************
Orient Origin Point:      UNINIT
X Direction Point:        UNINIT
Y Direction Point:        UNINIT
Active UFRAME/RTCP $MNUFRAMNUM[1]=0
```

12. To add a comment:
    a. Move the cursor to the comment line and press ENTER.
    b. Select a method of naming the comment.
    c. Press the appropriate function keys to enter the comment.
    d. When you are finished, press ENTER.
13. Define the origin point of the remote TCP frame **(Orient Origin Point)**:
    a. Move the cursor to Orient Origin Point.
    b. Jog the robot so that the TCP of the end-of-arm tooling or setup pointer touches the remote TCP of the fixed tool. See Figure 3–15 .
    c. Press and hold the SHIFT key and press F5, RECORD.

Figure 3–15. Touching the TCP of the Robot Tool to the Remote TCP



14. Define the +x direction point **(X Direction Point)**:

a. Move the cursor to X Direction Point.

b. Select an xyz coordinate system.

c. Jog the robot in the desired +x direction.

d. Press and hold the SHIFT key and press F5, RECORD.

15. Define a point on the positive X-Y plane **(Y Direction Point)**:

a. Move the cursor to Orient Origin Point.

b. Select an xyz coordinate system.

c. Press and hold the SHIFT key and press F4, MOVE_TO.

d. Move the cursor to Y Direction Point.

e. Jog the robot in the desired +y direction.

f. Press and hold the SHIFT key and press F5, RECORD.

16. **To select the RTCP frame to use** press F5, SETIND, type the number of the user frame you want, and press ENTER. This sets the active user frame ($MNUFRAMNUM[1]) to the number of the frame you want. If F5, SETIND, is not displayed, press PREV.

-OR-

Use the Jog Menu. Press and hold SHIFT and press COORD, move the cursor to RTCP, and type the number of the frame you want to use. Refer to Section 2.3.8 for more information.

17. Jog the robot in the +x, +y, and +z directions. The robot should move in the correct directions according to the frame you defined. If the robot does not move in the correct directions, go to Step 18 . Otherwise, go to Step 19 .

> **Caution**
>
> When you are finished setting the frame configuration, save the information to the default device so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

18. **To move to a recorded position**, move the cursor to the desired position, press and hold the SHIFT key and press F4, MOVE_TO.

> **Warning**
>
> **When you use F4, MOVE_TO, to jog the robot, unexpected motion can occur. This could injure personnel or damage equipment.**

Note If you want to clear the current RTCP frame selected, press NEXT, >, and then F2, CLRIND. This sets the active user frame ($MNUFRAMNUM[1]) to zero, which means that the default user frame is currently selected.

19. **To save the frames and related system variables to a file** on the default device,

a. Press MENUS.

b. Select FILE.

c. Press F1, [TYPE].

d. Select File.

e. Press F5, [UTIL].

f. Select Set Device.

g. Move the cursor to the device you want and press ENTER.

h. Display the RTCP frame screen.

i. Press FCTN.

j. Select SAVE. This will save the frame positions and comments for all frames to the file FRAMEVAR.VR, and the frame data to SYSFRAME.SV, on the default device.

20. To display the SYSTEM Variables menu,

a. Press MENUS.

b. Select SYSTEM.

c. Press F1, [TYPE].

d. Select Variables.

e. Press FCTN.

f. Select SAVE. The frame positions and system variables are saved in the SYSVAR.SV file, on the default device.

Procedure 3-10 Setting Up a Remote TCP Frame Using the Direct Entry Method

Warning

If you are setting up a new frame, make sure that all frame data is zero or uninitialized before you record any positions. Press F4, CLEAR, to clear frame data.

If you are modifying an existing frame, make sure that all frame data is set the way you want before you change it.

Otherwise, you could injure personnel or damage equipment.

Conditions

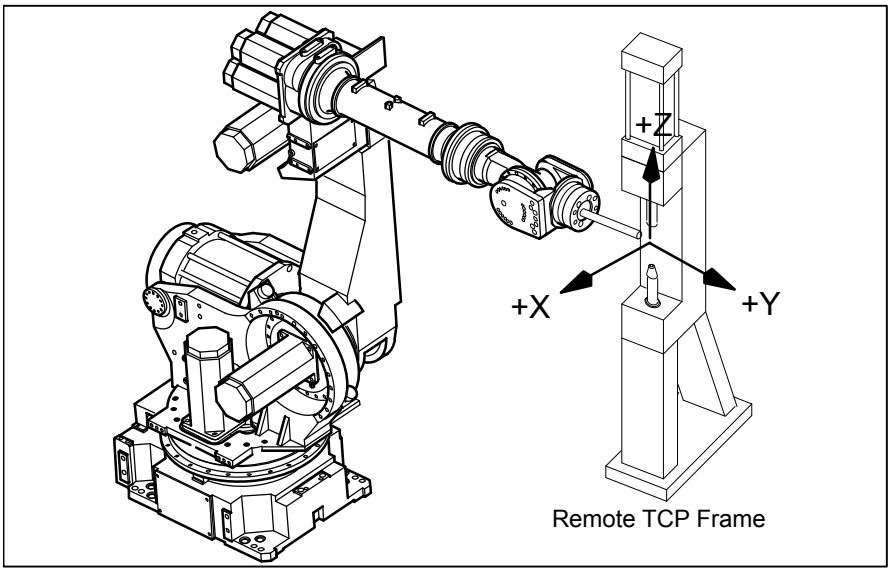
- The end-of-arm tooling or a setup pointer is attached to the robot faceplate.
- You have set up the tool frame for the end-of-arm tooling or setup pointer. Refer to Section 3.1.2.1 .

Steps

1. **Determine the Remote TCP of the fixed tool:**

   a. Jog the robot so that the TCP of the end-of-arm tooling or setup pointer touches the remote TCP of the fixed tool.

Figure 3–16. Touching the TCP of the Robot Tool to the Remote TCP



b. Press POSN.

c. Press F4, WORLD.

d. Write down the x, y, and z values of the current position in the WORLD coordinate system. This is the remote TCP of the fixed tool.

2. Use the Direct Entry Method to set up the Remote TCP frame:

a. Press MENUS.

b. Select SETUP.

c. Press F1, [TYPE].

d. Select Frames.

e. **If user/RTCP frames are not displayed**, press F3, [OTHER], and select User/RTCP. If F3, [OTHER], is not displayed, press PREV.

f. **To display the settings for all the frames**, press PREV repeatedly until you see a screen similar to the following.

```
SETUP Frames
User/RTCP Setup/ Direct Entry
X       Y      Z     Comment
1:   0.0     0.0    0.0 *************
2:   0.0     0.0    0.0 *************
3:   0.0     0.0    0.0 *************
4:   0.0     0.0    0.0 *************
5:   0.0     0.0    0.0 *************
6:   0.0     0.0    0.0 *************
Active UFRAME/RTCP $MNUFRAMNUM[1]=0
```

g. **To set the numerical values to zero**, move the cursor to the frame number, press F4, CLEAR, and then press F4, YES, to confirm.

h. Press F2, DETAIL.

i. **To select a frame**, press F3, FRAME, type the desired frame number, and press ENTER.

j. Press F2, [METHOD].

k. Select Direct Entry. You will see a screen similar to the following.

```
SETUP Frames
User/RTCP Setup/ Direct Entry
Frame Number: 1
1 Comment: ****************
2 X:                0.000
3 Y:                0.000
4 Z:                0.000
5 W:                0.000
6 P:                0.000
7 R:                0.000
Configuration:  N R D B, 0, 0, 0
Active UFRAME/RTCP $MNUFRAMENUM[1]=0
```

l. **To add a comment**, move the cursor to the comment line, press ENTER, select a method of naming the comment, and press the appropriate function keys to enter the comment. When you are finished, press ENTER.

m. Set each position component:

- Move the cursor to the x, y, and z components and enter the values you recorded in Step 1d .

- Move the cursor to the w, p, and r components and enter 0 for each component.

3. **To select the RTCP frame to use,** press F5, SETIND, type the number of the RTCP frame you want, and press ENTER. This sets the active user frame ($MNUFRAMNUM[1]) to the number of the frame you want. If F5, SETIND, is not displayed, press PREV.

   -OR-

   Use the Jog Menu. Press and hold SHIFT and press COORD, move the cursor to RTCP, and type the number of the frame you want to use. Refer to Section 2.3.8 for more information.

   Note If you want to clear the current frame to zero, move the cursor to the frame number and press NEXT, >, and then F2, CLRIND. This sets the active user frame ($MNUFRAMNUM[1]) to zero, which means that the default user frame is currently selected.

   Caution

   When you are finished setting the frame configuration, save the information to the default device so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

4. **To save the frames and related system variables to a file** on the default device,
   a. Press MENUS.
   b. Select FILE.
   c. Press F1, [TYPE].
   d. Select File.
   e. Press F5, [UTIL].
   f. Select Set Device.
   g. Move the cursor to the device you want and press ENTER.
   h. Display the RTCP frame screen.
   i. Press FCTN.
   j. Select SAVE. This will save the frame positions and comments for all frames to the file FRAMEVAR.VR, and the frame data to SYSFRAME.SV, on the default device.
5. To display the SYSTEM Variables menu,
   a. Press MENUS.
   b. Select SYSTEM.
   c. Press F1, [TYPE].

d. Select Variables.

e. Press FCTN.

f. Select SAVE. The frame positions and system variables are saved in the SYSVAR.SV file, on the default device. This is optional since frame variables are saved in SYSFRAME.SV.

Procedure 3-11 Selecting an RTCP Frame

Note To select the number of the RTCP frame you want to use, you can also use the jog menu. Refer to Section 2.3.8 for more information.

Conditions

- The user frame you want to select has been set up.

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Frames.
5. If user frames are not displayed press F3, [OTHER], and select User Frame. If F3, [OTHER], is not displayed, press PREV. You will see a screen similar to the following.

```
SETUP Frames
User/RTCP Setup/Direct Entry
X       Y      Z     Comment
1:    0.0     0.0    0.0 *************
2:    0.0     0.0    0.0 *************
3:    0.0     0.0    0.0 *************
4:    0.0     0.0    0.0 *************
5:    0.0     0.0    0.0 *************
6:    0.0     0.0    0.0 *************
Active UFRAME/RTCP $MNUFRAMNUM[1]=0
```

6. **To select the user frame to use** press F5, SETIND, type the number of the user frame you want, and press ENTER. This sets the active user frame ($MNUFRAMNUM[1]) to the number of the frame you want. If F5, SETIND, is not displayed, press PREV.

   -OR-

   Use the Jog Menu. Press and hold SHIFT and press COORD, move the cursor to RTCP, and type the number of the frame you want to use. Refer to Section 2.3.8 for more information.

7. The system variable $USEUFRAME defines whether the current value of $MNUFRAMENUM[group_no] will be assigned to the position's user frame when it is being recorded or touched up.

When $UFRAMENUM=FALSE, the initial recording of positions and the touching up of positions is done with the user frame number equal to 0, regardless of the value of $MNUFRAMENUM[group_no].

When $UFRAMENUM=TRUE, the initial recording of positions is done with the position's user frame equal to the user frame defined by $MNUFRAMENUM[group_no]. The touching up of positions must also be done with the position's user frame equal to the user frame defined by $MNUFRAMENUM[group_no].

Note When a teach pendant program is executed, you must make sure that the user frame of the position equals the value of $MNUFRAMENUM[group_no]; otherwise, an error will occur. Set the value of $MNUFRAMENUM[1] using the UFRAME_NUM=n instruction in the teach pendant program before you record the position to guarantee that the user frame numbers match during program execution.

3.1.5 Jog Frame

3.1.5.1 Setting Up Jog Frame

Jog frame is a frame that you can set up in any location, with any orientation. Jog frame provides a convenient way to move along a part when the part is oriented differently from the world frame. See Figure 3–17 .

You can set up jog frame so that the coordinates of jog frame correspond to the coordinates of the part. You can then jog along x, y, and z to teach the positions on the part.

- Before you use jog frame, you must set up its location and orientation.
- You can set up as many as five different jog frames for each robot.
- You can select one jog frame to be active at a time per robot group.
- You can jog the robot in jog frame.

Figure 3–17. Jog Frame Defined Parallel to Part

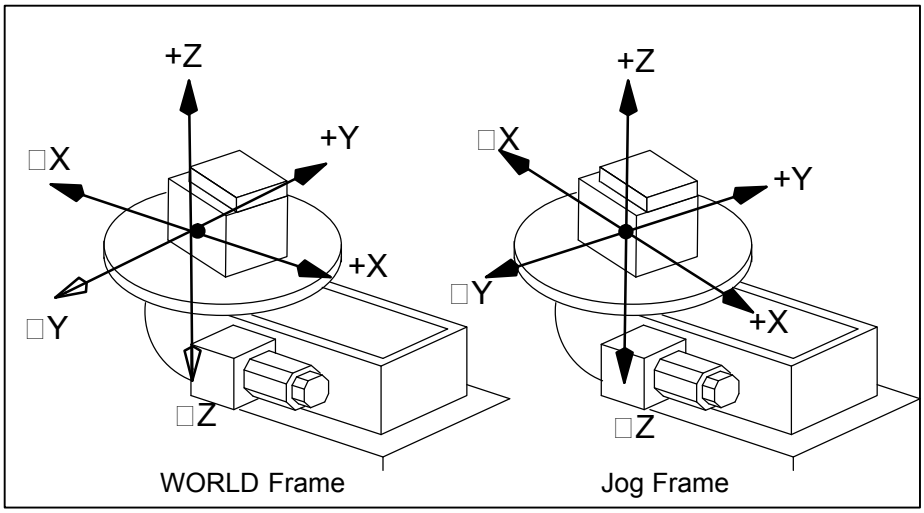


Table 3–9. Jog Frame Setup Screen Items

ITEM	DESCRIPTION
Jog frame setup/xxxx	This line shows the current selected method for setting each jog frame.
Frame number 1-5	These lines show the current position and comment for each available jog frame.
X	This column shows the X coordinate of each jog frame.
Y	This column shows the Y coordinate of each jog frame.
Z	This column shows the Z coordinate of each jog frame.
Comment	This column shows the comment for each jog frame.
Active JOG FRAME[1] = 0	This line indicates which jog frame is currently selected for use.
[TYPE]	Press this key to access various application-specific options.
DETAIL	Press this key to display detailed information for each jog frame and to set the definition and comment of each jog frame.
OTHER	Press this key to select one of the other available reference frames or to choose the motion group for the frame you are setting up (in systems with multiple motion groups).
CLEAR	To set the numerical values of any jog frame to zero, move the cursor to the frame number, press CLEAR and answer YES to the prompt.
SETIND	To select the jog frame to use, press F5, SETIND, type the number of the jog frame you want, and press ENTER. This sets the active jog frame to the number of the frame you want. If F5, SETIND, is not displayed, press PREV.

You can use two methods to define the jog frame.

- Three point method
- Direct entry method

Three Point Method

The three point method allows you to define a jog frame by recording three points: the origin, a point along the +x-axis of the user frame, and a point on the x-y plane of the user frame (defines the x-y plane and the y-z plane).

Table 3–10. Jog Frame Setup Three Point DETAIL Screen Items

ITEM	DESCRIPTION
Frame number	This line shows the number of the jog frame being defined.
X, Y, Z, W, P, R	These items show the current coordinates of the jog frame being defined.
Comment	This item is used to add a comment.
Orient Origin Point	This item is used to define the reference point of the jog frame.
X Direction Point	This item is used to define the +x direction point.
Y Direction Point	This item is used to define a point on the X-Y plane.
Active JOG FRAME[1] = 0	This item indicates which jog frame is currently selected for use
[TYPE]	Press this key to access various application-specific options.
[METHOD]	Press this key to select a method of entry.
[FRAME]	Press this key to select a frame.
MOVE_TO	Press this key to move to a recorded position. Move the cursor to the desired point and press MOVE_TO along with the SHIFT key.
RECORD	Press this key along with the SHIFT key to record a position.

Direct Entry Method

The direct entry method allows you to designate the origin with values for x, y, z, w, p, and r. This method provides direct recording and numerical entry of the frame position.

Table 3–11. Jog Frame Setup Direct Entry DETAIL Screen Items

ITEM	DESCRIPTION
Frame number	This line shows the number of the jog frame being defined.
Comment	This item is used to add a comment.
X	This item is used to define the component of the jog frame.
Y	This item is used to define the component of the jog frame.
Z	This item is used to define the component of the jog frame.

Table 3–11. Jog Frame Setup Direct Entry DETAIL Screen Items (Cont'd)

W	This item is used to define the component of the jog frame.
P	This item is used to define the component of the jog frame.
R	This item is used to define the component of the jog frame.
Configuration	This line displays the current robot configuration.
Active JOG FRAME[1] = 0	This line indicates which jog frame is currently selected for use.
[TYPE]	Press this key to access various application-specific options.
[METHOD]	Press this key to select a method of entry.
[FRAME]	Press this key to select a frame.
MOVE_TO	Press this key along with the SHIFT key to move to a recorded position.
RECORD	Press this key along with the SHIFT key to record a position.

Use Procedure 3-14 to select a jog frame.

3.1.5.2 HandlingTool Jog Frame Three Point Method

Procedure 3-12 Setting Up the Jog Frame Using the Three Point Method

Caution

Do not set up or alter frames when you use PalletTool. Frames are automatically set up for you by PalletTool.

Warning

If you set up a new frame, make sure that all frame data is zero or uninitialized before you record any positions. Press F4, CLEAR, to clear frame data.

If you modify an existing frame, make sure that all frame data is set the way you want before you change it.

Otherwise, you could injure personnel or damage equipment.

Conditions

- You have a cardboard box.

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Frames.
5. **To choose the motion group** for the frame you are setting up in systems with multiple motion groups press F3, [OTHER], and select the group you want. The default motion group is Group 1.

   If you have the new *i*Pendant, you can press the GROUP key and press the numeric key to switch to a specified group. Press the GROUP key and the 0 key together to toggle the sub group.

   Warning

   Do not run a KAREL program that includes motion statements if more than one motion group is defined on your controller. If your controller is set up for more than one motion group, all motion must be initiated from a teach pendant program. Otherwise, the robot could move unexpectedly, personnel could be injured, and equipment could be damaged.

6. **If jog frames are not displayed**, press F3, [OTHER], and select Jog Frame. If F3, [OTHER], is not displayed, press PREV.
7. **To display the settings for all frames**, press PREV repeatedly until you see a screen similar to the following.

```
SETUP Frames
JOG Frame Setup / Three Point
X       Y       Z    Comment
1:    0.0     0.0     0.0 *************
2:    0.0     0.0     0.0 *************
3:    0.0     0.0     0.0 *************
4:    0.0     0.0     0.0 *************
5:    0.0     0.0     0.0 *************
Active JOG FRAME[1] = 0
```

8. **To set the numerical values to zero,**, move the cursor to the frame number, press F4, CLEAR, and then press F4, YES, to confirm.
9. Press F2, DETAIL.
10. **To select a frame**,

a. Press F3, FRAME.

b. Type the desired frame number.

c. Press ENTER.

11. Press F2, [METHOD].

12. Select Three Point. You will see a screen similar to the following.

```
SETUP Frames
Jog Frame Setup / Three Point
Frame Number: 2
X  0.0      Y  0.0      Z  0.0
W  0.0      P  0.0      R  0.0
Comment: ***************
Orient Origin Point:  UNINIT
X Direction Point:    UNINIT
Y Direction Point:    UNINIT
Active JOG FRAME[1] = 0
```

13. To add a comment:

a. Move the cursor to the comment line and press ENTER.

b. Select a method of naming the comment.

c. Press the appropriate function keys to enter the comment.

d. When you are finished, press ENTER.

14. Mount a box within the workcell so that the orientation of the box matches the orientation of the desired jog frame. Make sure that the corner of the box used to record the origin is at the proper location.

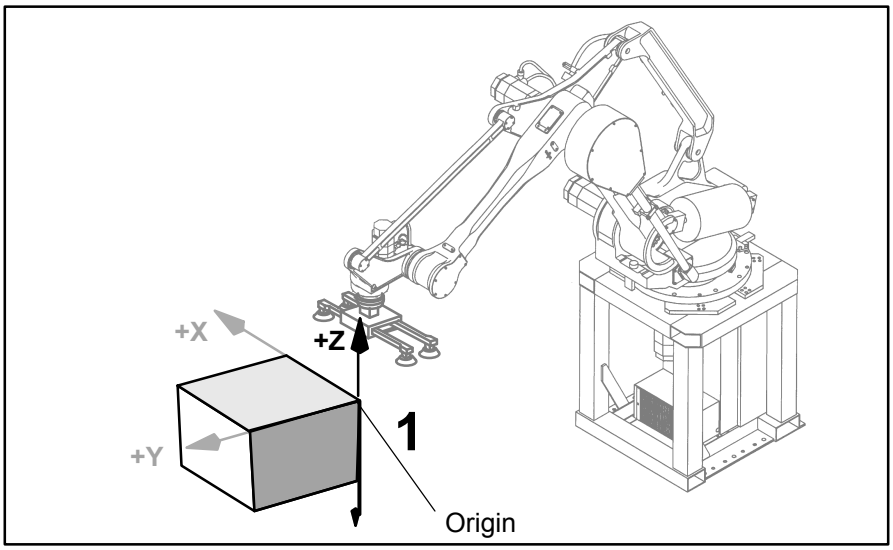
15. Define the origin of the jog frame **(System Origin Point)**:

a. Move the cursor to System Origin Point.

b. Jog the robot TCP to the origin. In Figure 3–18 the origin is labeled number 1.

c. Press and hold the SHIFT key and press F5, RECORD.

Figure 3–18. Defining the Origin



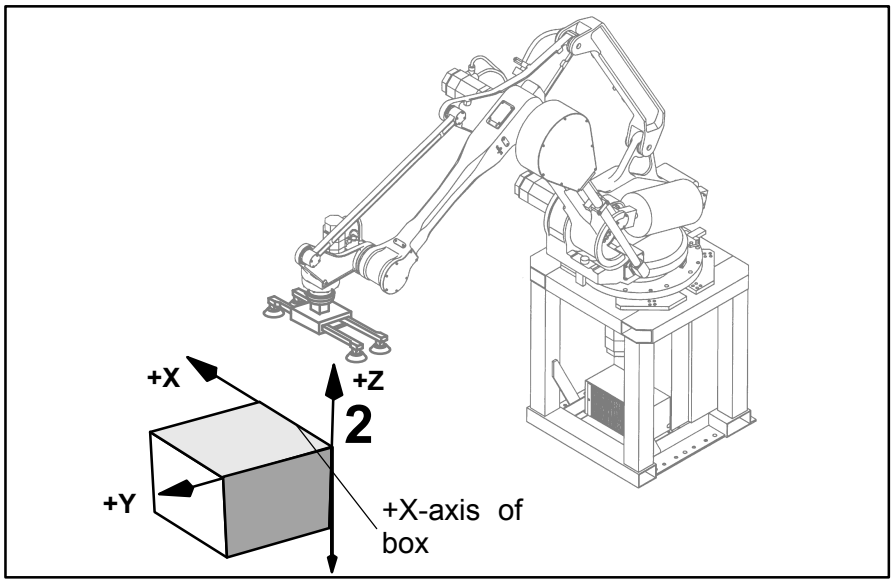
16. Define the +x direction point **(X Direction Point)**:

a. Move the cursor to X Direction Point.

b. Jog the robot TCP to a point along the +x-axis of the box. In Figure 3–19 , this point is labeled number 2.

c. Press and hold the SHIFT key and press F5, RECORD.

Figure 3–19. Defining the X Direction Point



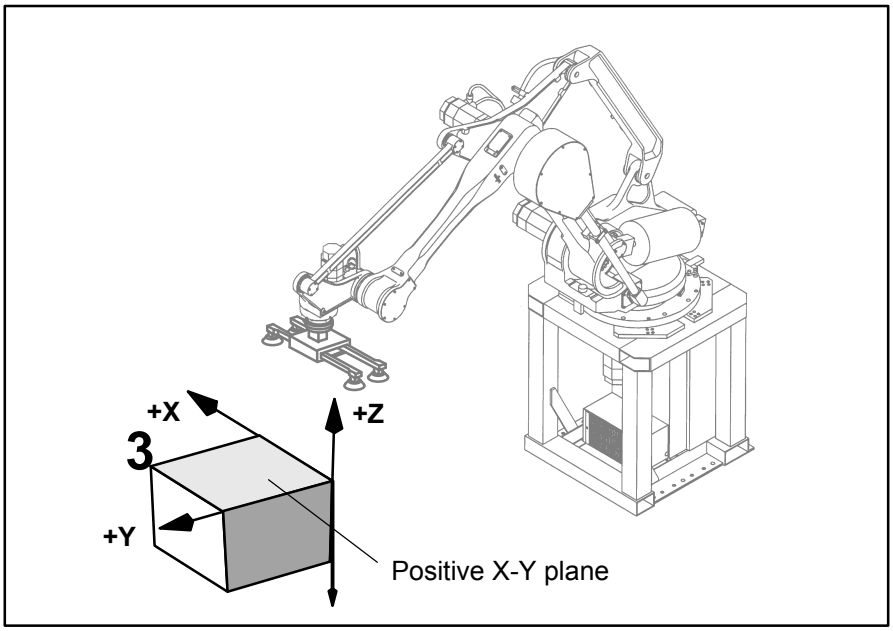
17. Define a point on the positive X-Y plane **(Y Direction Point)**:

a. Move the cursor to Y Direction Point.

b. Jog the robot to a location on the positive X-Y plane. In Figure 3–20 , this point is labeled number 3.

c. Press and hold the SHIFT key and press F5, RECORD.

Figure 3–20. Defining the X-Y Plane



18. **To select the jog frame to use,** press F5, SETIND, type the number of the jog frame you want, and press ENTER. If F5, SETIND, is not displayed, press PREV.

-OR-

Use the Jog Menu. Press and hold SHIFT and press COORD, move the cursor to Jog, and type the number of the frame you want to use. Refer to Section 2.3.8 for more information.

19. Jog the robot in the +x, +y, and +z directions. The robot should move in the correct directions according to the frame you defined. If the robot does not move in the correct directions, go to Step 20 . Otherwise, go to Step 21 .

Caution

When you are finished setting the frame configuration, save the information to the default device so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

Warning

When you use F4, MOVE_TO, to jog the robot, unexpected motion can occur. This could injure personnel or damage equipment.

20. **To move to a recorded position**, move the cursor to the desired position, press and hold the SHIFT key and press F4, MOVE_TO.
21. **To save the frames and related system variables to a file** on the default device,
    a. Press MENUS.
    b. Select FILE.
    c. Press F1, [TYPE].
    d. Select File.
    e. Press F5, [UTIL].
    f. Select Set Device.
    g. Move the cursor to the device you want and press ENTER.
    h. Display the jog frame screen.
    i. Press FCTN.
    j. Select SAVE. This will save the frame positions and comments for all frames to the file FRAMEVAR.VR, and the frame data to SYSFRAME.SV, on the default device.

    Display the SYSTEM Variables menu,

    k. Press MENUS.
    l. Select SYSTEM.
    m. Press F1, [TYPE].
    n. Select Variables.
    o. Press FCTN.
    p. Select SAVE. The frame positions and system variables are saved in the SYSVAR.SV file, on the default device.

3.1.5.3 Jog Frame Direct Entry Method

Procedure 3-13 Setting Up the Jog Frame Using the Direct Entry Method

Caution

Do not set up or alter frames when you use PalletTool. Frames are automatically set up for you by PalletTool.

Warning

If you set up a new frame, make sure that all frame data is zero or uninitialized before you record any positions. Press F4, CLEAR, to clear frame data.

If you modify an existing frame, make sure that all frame data is set the way you want before you change it.

Otherwise, you could injure personnel or damage equipment.

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. **To choose the motion group** for the frame you are setting up in systems with multiple motion groups press F3, [OTHER], and select the group you want. The default motion group is Group 1.

   If you have the new *i*Pendant, you can press the GROUP key and press the numeric key to switch to a specified group. Press the GROUP key and the 0 key together to toggle the sub group.

   Warning

   Do not run a KAREL program that includes motion statements if more than one motion group is defined on your controller. If your controller is set up for more than one motion group, all motion must be initiated from a teach pendant program. Otherwise, the robot could move unexpectedly, personnel could be injured, and equipment could be damaged.

5. Select Frames.
6. **If jog frames are not displayed**, press F3, [OTHER] and select Jog Frame. If F3, [OTHER], is not displayed, press PREV.
7. **To display the settings for all frames**, press PREV repeatedly until you see a screen similar to the following.

```
SETUP Frames
JOG Frame Setup / Three Point
X      Y       Z    Comment
1:    0.0    0.0     0.0 *************
2:    0.0    0.0     0.0 *************
3:    0.0    0.0     0.0 *************
4:    0.0    0.0     0.0 *************
5:    0.0    0.0     0.0 *************
Active JOG FRAME[1] = 0
```

8. **To set the numerical values to zero**, move the cursor to the frame number, press F4, CLEAR, and then press F4, YES, to confirm.
9. Press F2, DETAIL.
10. **To select a frame**,
    a. Press F3, FRAME.
    b. Type the desired frame number.
    c. Press ENTER.
11. Press F2, [METHOD].
12. Select Direct Entry. You will see a screen similar to the following.

```
SETUP Frames
Jog Frame Setup / Direct Entry
Frame Number: 1
1 Comment: ****************
2 X:                      0.000
3 Y:                      0.000
4 Z:                      0.000
5 W:                      0.000
6 P:                      0.000
7 R:                      0.000
Configuration:      N R D B, 0, 0, 0
Active JOG FRAME[1] = 0
```

13. To add a comment:
    a. Move the cursor to the comment line and press ENTER.
    b. Select a method of naming the comment.
    c. Press the appropriate function keys to enter the comment.

d. When you are finished, press ENTER.

14. Set each position component:

a. Move the cursor to the component.

b. Enter the numeric value for the component.

c. Press ENTER to set the new value.

15. **To select the jog frame to use**, press F5, SETIND, type the number of the jog frame you want, and press ENTER. If F5, SETIND, is not displayed, press PREV.

-OR-

Use the Jog Menu. Press and hold SHIFT and press COORD, move the cursor to Jog, and type the number of the frame you want to use. Refer to Section 2.3.8 for more information.

> **Caution**
>
> When you are finished setting the frame configuration, save the information to the default device so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

16. **To save the frames and related system variables to a file** on the default device,

a. Press MENUS.

b. Select FILE.

c. Press F1, [TYPE].

d. Select File.

e. Press F5, [UTIL].

f. Select Set Device.

g. Move the cursor to the device you want and press ENTER.

h. Display the jog frame screen.

i. Press FCTN.

j. Select SAVE. This will save the frame positions and comments for all frames to the file FRAMEVAR.VR, and the frame data to SYSFRAME.SV on the default device.

Display the SYSTEM Variables menu,

17. **To save the SYSTEM Variables (optional since data is in SYSFRAME.SV)**

a. Press MENUS.

b. Select SYSTEM.

c. Press F1, [TYPE].

d. Select Variables.

e. Press FCTN.

f. Select SAVE. The frame positions and system variables are saved in the SYSVAR.SV file, on the default device.

3.1.5.4 Selecting a Jog Frame

Procedure 3-14 Selecting a Jog Frame



Caution

Do not set up or alter frames when you use PalletTool. Frames are automatically set up for you by PalletTool.

Note You can also use the Jog Menu to select the number of the jog frame you want to use. Refer to Section 2.3.8 for more information.

Conditions

- The jog frame you want to select has been set up.

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Frames.
5. If jog frames are not displayed press F3, [OTHER], and select Jog Frame. If F3, [OTHER], is not displayed, press PREV. You will see a screen similar to the following.

```
SETUP Frames
Jog Frame Setup / Direct Entry
X     Y     Z   Comment
1: 0.0   0.0   0.0 *************
2: 0.0   0.0   0.0 *************
3: 0.0   0.0   0.0 *************
4: 0.0   0.0   0.0 *************
5: 0.0   0.0   0.0 *************
Active JOG FRAME[1] = 0
```

6. **To select the jog frame to use,** press F5, SETIND, type the number of the jog frame you want, and press ENTER. This copies the selected jog frame to $JOG_GROUP[group_no].$JOGFRAME. If F5, SETIND, is not displayed, press PREV.

   -OR-

   Use the Jog Menu. Press and hold SHIFT and press COORD, move the cursor to Jog, and type the number of the frame you want to use. Refer to Section 2.3.8 for more information.

3.1.6 Saving Frame Data

Saving frame data saves the frame positions and comments. Use Procedure 3-15 to save frame data to a file.

> **Caution**
>
> Do not set up or alter frames when you use PalletTool. Frames are automatically set up for you by PalletTool.

Procedure 3-15 Saving Frame Data to a File

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Frames.
5. Press F2, DETAIL.
6. **To select a frame,**
   a. Press F3, FRAME.
   b. Type the desired frame number.
   c. Press ENTER.
7. Press F2, [METHOD].
8. Select a frame method. You will see a screen similar to the following.

```
SETUP Frames
Tool Frame Setup / Three Point
X       Y        Z         Comment
1:   0.0    0.0     0.0  *************
2:   0.0    0.0     0.0  *************
3:   0.0    0.0     0.0  *************
4:   0.0    0.0     0.0  *************
5:   0.0    0.0     0.0  *************
6:   0.0    0.0     0.0  *************
ACTIVE TOOL $MNUTOOLNUM[1]=1
```



Caution

When you are finished setting the frame configuration, save the information to the default device so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

9. To save the frames and related system variables to a file on the default device,

a. Press MENUS.

b. Select FILE.

c. Press F1, [TYPE].

d. Select File.

e. Press F5, [UTIL].

f. Select Set Device.

g. Move the cursor to the device you want and press ENTER.

h. Display the frame screen.

i. Press FCTN.

j. Select SAVE. This will save the frame positions and comments for all frames to the file FRAMEVAR.VR, and the frame data to SYSFRAME.SV, on the default device.

10. To save the SYSTEM Variables,

a. Press MENUS.

b. Select SYSTEM.

c. Press F1, [TYPE].

d. Select Variables.

e. Press FCTN.

f. Select SAVE. The frame positions and system variables are saved in the SYSVAR.SV file, on the default device. This is optional since frame variables are saved in SYSFRAME.SV

3.2 PRODUCTION OPERATION SETUP

3.2.1 Production Operation Setup Overview

Production operation setup allows you to set up a program so that it is run automatically during production.

To run production you can use

- Program Select method (Remote/Local must be set to Remote)
  - — Program Number Select (PNS)
  - — Robot Service Request (RSR)
  - — Style Program Select (optional) *
  - — OTHER Program Select Mode *
- Production Start method (Remote/Local must be set to Remote)
  - — UOP PRODUCTION START input
  - — DIN[ ] only available for Style Program Select
  - — OTHER
- Default program on SELECT menu method (Remote/Local must be set to Local)
  - — SOP CYCLE START input

This section includes information on how to set up RSR and PNS programs and Style Program Select. UOP PRODUCTION START and SOP CYCLE START inputs do not require any software setup for RSR and PNS. Refer to Chapter 7 *TESTING A PROGRAM AND RUNNING PRODUCTION* for more information.

Before you can run production, you need to provide the information for the items on the Production Setup screen shown in Figure 3–21 . Table 3–12 describes the items shown on the Production SETUP screen.

Figure 3–21. Production SETUP Screen

```
Prog Select
1 Program select mode:              STYLE
2 Production start method           DIN[]
Production checks:
3   At home check:                 ENABLED
4   Resume position toler.:       DISABLED
5   Simulated I/O:                DISABLED
6   General override < 100%:      DISABLED
7   Prog override < 100%:         DISABLED
8   Machine lock:                 DISABLED
9   Single step:                   ENABLED
10  Process ready:                DISABLED
General controls:
11  Heartbeat timing:              1000 MS
12  Low TEMP DRAM memory:          100 KB
13  Low PERM CMOS memory:          50  KB
14  RESET when DEADMAN pressed:    ENABLED
```

Table 3–12. Production Setup Screen

Production Setup ITEM	DESCRIPTION
Program Select Mode	This item specifies which method will be used to select the program to run: • **RSR** uses the RSR method • **PNS** uses the PNS method • **STYLE** uses the Style Name method **Note** STYLE can be used to start a single program. MASH allows you to treat multiple independent programs as a single program. • **OTHER** uses the program selected by setting the system variable $SHELL_WRK.$cust_name to the name of the program to run

Table 3–12. Production Setup Screen (Cont'd)

Production Setup ITEM	DESCRIPTION
Production Start Method	This item specifies which signal is used to start or continue a program: • When you specify **UOP**, RSR and PNS are used as defined originally using UOP signals. • The **STYLE** program select mode can use the **OTHER** or UOP production start method. • When you specify**OTHER**, the program is selected using STYLE or OTHER, and the program is initiated by setting the system variable $SHELL_WRK.$cust_start = 1.

Table 3–12. Production Setup Screen (Cont'd)

Production Setup ITEM	DESCRIPTION

Production Checks - The following items are checked before running production. When you move the cursor to one of the production check items and press F3, DETAIL, the following DETAIL screen is displayed.

```
Prog Select DETAIL
Check :   At home check
1 Check when run:          ENABLED
2 Check when resume:       ENABLED
3 Prompt if failure:       ENABLED
4 Post error if failure:   ENABLED
5 Post warning if forced:  ENABLED
6 Force condition:         ENABLED
```

- **Check when run** indicates whether the system will perform the specified production check when a program is executed.
- **Check when resume** indicates whether the systems will perform the specified production check when a program is resumed.

If the specified Production check fails when the program is executed or resumed, the following actions will be taken. If "Check when run" or "Check when resume" are disabled, these actions are ignored. The "Force condition" action takes priority over the other actions. If "Force condition" and "Prompt if failure" are both enabled, only the "Force condition" action is performed.

- **Prompt if failure** allows you to specify that a prompt will be displayed on the teach pendant screen if the specified check caused the program start or program resume to fail. For example, if the Production check General override <100 is enabled, a prompt will be displayed on the teach pendant screen when the general override is less than 100. The user can then determine how to process: continue, force, or abort.
- **Post error if failure** is always enabled and displays an error message on the alarm log to indicate what caused the specified check for the program start or resume to fail. In the previous example, the message"SYST-085 Gen override not 100" would be displayed.
- **Post warning if forced** is used only when the Force condition check is enabled. It displays an error message on the alarm log.
- **Force condition** is not available for "At home check," "Resume position toler.", and "Machine lock" Production checks. This check overrides the Prompt if failure check. If the "Post warning if forced" check is enabled, a message is displayed on the alarm log indicating that the condition has been forced.

Production Check: At Home Check	When this item is set to ENABLED, production will be run if the robot is at the home position. When you press F3, DETAIL, you can specify in more detail the behavior of the at home checking. SO[6] - User LED#2 - will turn ON when At Home Check is enabled and the robot is in home position. If you do not want to turn this output on, set $SHELL_CFG.$DO_HOME_SOP = FALSE.

Table 3–12. Production Setup Screen (Cont'd)

Production Setup ITEM	DESCRIPTION
Production Check: Resume Position Toler	When this item is set to ENABLED, production will be run if the robot is at or near the position in which the program was paused. This check is ENABLED by default. If you move the cursor to this item and press F3, DETAIL, you will not be able to set "Check when resume" to DISABLED. If you want to disable "Check when resume," go to the SETUP RESUME TOL screen and disable resume tolerance for each motion group.
Production Check: Simulated I/O	When this item is set to ENABLED, production will be run if I/O signals are not simulated.
Production Check: General Override < 100%	When this item is set to ENABLED, production will be run only if the general override is set to 100%.
Production Check: Prog Override < 100%	When this item is set to ENABLED, production will be run only if the program override is set to 100%.
Production Check: Machine Lock	When this item is set to ENABLED, production will be run if Robot Lock is set to OFF on the SETUP TEST CYCLE screen.
Production Check: Single Step	When this item is set to ENABLED, production will be run if the robot is**not** in single step mode.

Table 3–12. Production Setup Screen (Cont'd)

Production Setup ITEM	DESCRIPTION

Production Check: Process Ready	This item is application-specific. When this item is set to ENABLED in a spot welding application, the following items will be checked for faults during production initiation.
	• Welder.
	• Water saver.
	• Water flow.
	• X-former overtemp.
	If a fault is detected, you will see the following prompt box.
	`Application process error`
	`Please check process`
	`Peripheral equipment`
	`CONTINUE RECHECK [STOP]`
	• **CONTINUE** will ignore the error and proceed to the next setup item.
	• **RECHECK** will immediately recheck the faulted I/O point. If it is still faulted, the prompt box will be displayed again.
	• **STOP** will cancel the production start and require another production start when ready.

Heartbeat timing: default: 1000 ms min: 0 ms max: 10,000 ms	The heartbeat is an output signal that is pulsed (turns on and off) at the interval you specify in this item. The heartbeat is used by the PLC to verify that the robot is still "alive." The heartbeat digital output is assigned using the I/O cell output menu. The digital output is pulsed at the heartbeat interval. If the timing is 0 or the output is not assigned, then the heartbeat is disabled.
Low TEMP DRAM memory: default: 100 kb min: 0 kb max: 9999 kb	This item allows you to define the lowest amount of TEMP DRAM memory below which an error message with WARN severity will be displayed. This item is checked every time a production program is run.

Table 3–12. Production Setup Screen (Cont'd)

Production Setup ITEM	DESCRIPTION
Low PERM CMOS memory: default: 50 kb min: 0 kb max: 9999 kb	This item allows you to define the lowest amount of CMOS memory below which an error message with WARN severity will be displayed. This item is checked every time a production program is run.
RESET When DEADMAN Pressed	This item allows you to specify how the system recovers from a DEADMAN release when the teach pendant is turned on: • ENABLED - If you release the DEADMAN and then press it again, the system will automatically perform a fault reset. • DISABLED - If you release the DEADMAN and then press it again, you must then press a RESET key to reset the fault.

3.2.2 Robot Service Request (RSR) Setup

A*Robot Service Request* (RSR) is a request for service from an external device. That request comes from a digital input signal on a preassigned RSR input line. You can use up to eight robot service request signals: RSR1, RSR2, RSR3, RSR4, RSR5, RSR6, RSR7, and RSR8.

When the robot controller receives a service request signal, the controller determines whether the signal is acceptable. If acceptable, the controller determines which program to execute.

Caution

Any program that you want to execute by using RSRs must be named RSR[nnnn], where [nnnn] represents a four digit number from 0001 to 9999; otherwise, the program will not be executed.

If no other program is currently running, the program assigned to the RSR input line starts. If a program is currently running, the robot stores the signal and runs the program when the other program is finished.

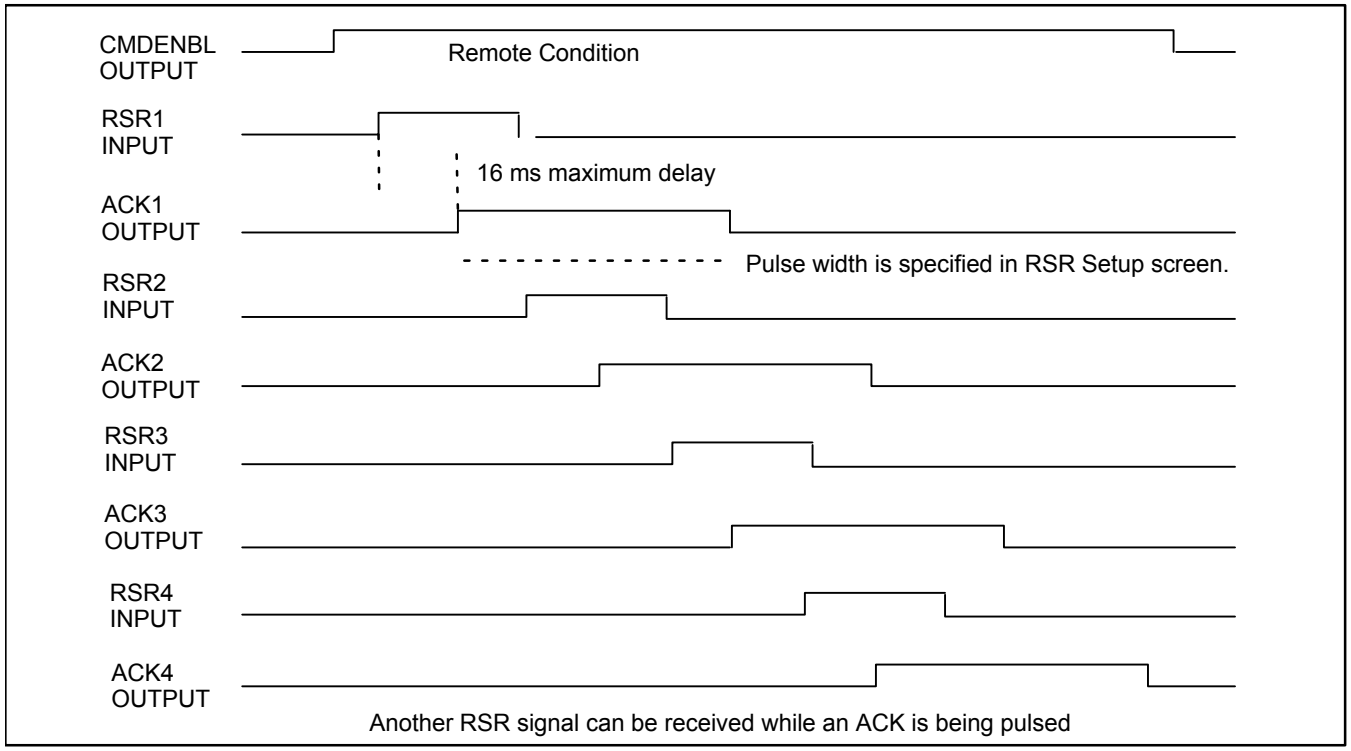
If you are using UOP RSR, when the robot receives the RSR signal, the robot can output the corresponding acknowledge signals (STYLE_ACK1–STYLE_ACK4) if the signals are enabled.

When the robot receives the RSR signal, the robot can output the corresponding acknowledge signals if the signals are enabled.

The sequence and timing diagrams are similar for UOP RSR and digital RSR. See Figure 3–22 .

RSR signals have specific requirements when used with DualARM systems using DualARC software.

Figure 3–22. RSR Timing Diagram



The I/O sequence in Figure 3–22 will cause RSR1 to execute after any currently executing program has completed. RSR2 program will execute when RSR1 program completes. RSR2– RSR4 are queued to execute by the system.

Table 3–13 lists and describes each RSR setup item.

Table 3–13. RSR Setup Item Description

RSR SETUP ITEM	DESCRIPTION
RSR1 Program Number default: 0 min: 0 max: 999	This item allows you to enter a number that when added to the base number defines the program number that will be executed when the RSR1 signal is received. For example, if you entered 0023 for the RSR1 program number and the base number was set to 100, the RSR1 signal would execute program RSR0123. If you enter an invalid program number, the system will ignore this signal.
RSR2– RSR8 Program Number	This item allows you to enter a number that when added to the base number defines the program number that will be executed when the RSR2 – RSR8 signal is received. If you enter an invalid program number or a zero, the system will ignore this signal.
Base Number default: 0 min: 0 max: 9999	This item allows you to enter a number that when added to the RSR1– RSR8 program number defines which program will be executed. This base number can be changed from within your program by using the PARAMETER NAME instruction. The parameter that contains the RSR base number is**$SHELL_CFG.$job_base** . By changing the base number, you can control which group of programs will be executed.
Job Prefix — SpotTool+ default: RSR	This item is the three-letter prefix of the program to be selected. The system variable is $SHELL_CFG.$job.root.
Acknowledge Function default: FALSE	This item allows you to enable or disable robot acknowledge output signals ACK1-8. FALSE means the signals are disabled. TRUE means the signals are enabled.
Acknowledge Pulse Width default: 400 ms min: 0 max: 9999	This item allows you to set the length (in milliseconds) of the ACK1-8 signal when the acknowledge function is enabled. This time depends on the scan time of your PLC program.

Use Procedure 3-16 to set up RSRs.

Procedure 3-16 RSR Setup

Conditions

- UOP signals must be installed and configured.
- The program name must be RSR[nnnn] where [nnnn] represents a four digit number from 0001 to 9999.

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Prog Select. You will see a screen similar to the following.

```
Prog Select
1 Program select mode:              RSR
2 Production start method           DIN[]
Production checks:
3   At home check:                 ENABLED
4   Resume position toler.:       DISABLED
5   Simulated I/O:                DISABLED
6   General override < 100%:      DISABLED
7   Prog override < 100%:         DISABLED
8   Machine lock:                 DISABLED
9   Single step:                   ENABLED
10  Process ready:                DISABLED
General controls:
11  Heartbeat timing:              1000 MS
12  Low TEMP DRAM memory:          100 KB
13  Low PERM CMOS memory:          50  KB
14  RESET when DEADMAN pressed:    ENABLED
```

5. Move the cursor to Program Select Mode and press F4, [CHOICE].
6. Select RSR and press ENTER
7. Press F3, DETAIL. You will see a screen similar to the following.

```
Prog Select
1/7
RSR or PNS
1 RSR1 program number  [ENABLE ]  [0012]
2 RSR2 program number  [ENABLE ]  [0003]
3 RSR3 program number  [ENABLE ]  [0018]
4 RSR4 program number  [ENABLE ]  [0064]
5 RSR5 program number  [ENABLE ]  [    ]
6 RSR6 program number  [ENABLE ]  [    ]
7 RSR7 program number  [ENABLE ]  [    ]
8 RSR8 program number  [ENABLE ]  [    ]
9 Base number                     [ 100 ]
10 Acknowledge function            [FALSE]
11 Acknowledge pulse width(msec)   [  10]
```

8. Move the cursor to the item you want to set and type the value.
9. **When you are finished** making changes, turn off the controller and then turn it back on.

Warning

When you change the kind of production operation, you must turn off the controller and turn the controller on to use the new information; otherwise, the new settings can not be accepted.

3.2.3 Program Number Select (PNS) Setup

A*Program Number Select (PNS)* is a method of selecting the name of a program to be run by some external device. The name of the program to be run comes as a group of input signals from an external device on eight PNS input lines.

Selecting a Program Using UOPs for SpotTool+ and DispenseTool

If UOP signals are used, the name of the program to be run comes as a group of input signals from an external device on eight PNS input lines.

The following sequence takes place with PNS operation:

Caution

Any program that you want to execute by using PNS must be named PNS[nnnn], where [nnnn] represents a four digit number from 0001 to 9999; otherwise, the program will not be executed.

1. The robot must be in REMOTE when the PNSTROBE signal is received. The eight PNS inputs signals are read as a binary number by the system.
2. The eight PNS input signal a binary number to the system.
3. The binary number is added to the base number if a base number is used. Refer to Table 3–14 . This defines the program number to be executed and makes that program the default program.
4. SNO1-8 is loaded with the binary number of the original eight PNS inputs.
5. The robot will run the program when the PROD_START input signal is received.
6. SNACK is pulsed to signal the external device to read, SNO1-8.
7. The PLC can use SNO1-8 and SNACK to check the PNS number. If the number received on SNO 1-8 is the same as the number sent out on PNS 1-8, the PROD_START input signal is sent to the controller.
8. If $SHELL.CFG.$ustart=TRUE, then the program starts when the start turns ON. This is the default setting for SpotTool+ and DispenseTool. If $SHELL.CFG.$ustart=FALSE then the program starts when the start turns OFF. This is the default setting for HandlingTool and ArcTool.
9. The robot will run the program within 32 ms from when PROD_START turns off. Refer to Figure 4–15 in Section 4.9.2 for a timing diagram.

PNS signals can be used for multi-tasking. After a program has started running, PNS signals and the START input can be used to execute a second program. The system variable $SHELL_CFG.$cont_only must be set to FALSE to allow the START input to execute the currently selected program.

A PNS sequence can be initiated in one of two ways depending upon the ENABLE/DISABLE status of the Job Queue item in the Prog Select screen.

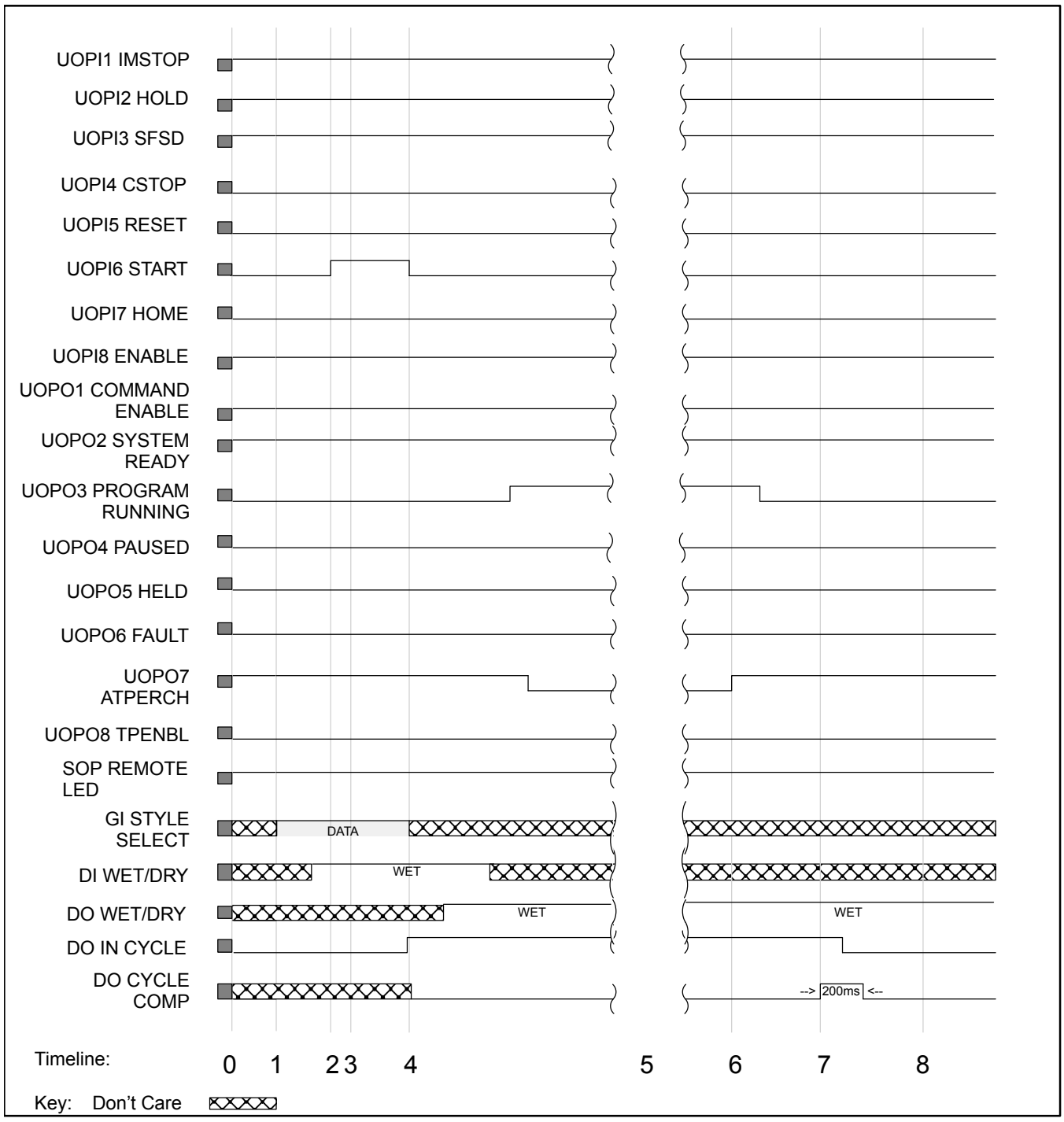
When you are using UOP signals, the PNS sequences are started by the rising or falling edge of the UOP CYCLE START input. Whatever style is on the PNS UOP, inputs are then echoed on the SNO_UOP outputs (if the Cell Setup Ack Style Data item is set to TRUE) and JOB corresponding to the number read from the PNS inputs is executed immediately. For the DispenseTool application, this is the simplest form of PNS handshaking and is usually all that is required in a cell interface.

The timing diagram shown in Figure 3–23 is intended to give a general overview of the cell interface as it applies to starting and ending a JOB for standard DispenseTool. The following conditions apply:

- PNS with Invert Handshaking DISABLED.
- ACKNOWLEDGE STYLE DATA is set to FALSE.
- Robot state reporting is DISABLED.
- Hold cycle complete high ENABLED or DISABLED.
- PLC fault reporting/recovery does not apply in this example.
- Degrade/backup groups DISABLED.
- Interference zones and user inputs and outputs do not apply to this example.

Note If multitasking is used, the PNS programs must have unique (exclusive) group masks. The system will post INTP-105 “Run request failed” and PROG-040 “Already locked by other task” indicating the conflict in group mask use.

Figure 3–23. PNS Timing Diagram for DispenseTool (default and recommended configuration)



The**sequence of actions** as shown in Figure 3–23 is as follows:

Time 0

In order to enter production mode and not be interrupted, the following signals must be set to specified states for the duration of the JOB:

- UOPI1 QSTOP = ON
- UOPI2 HOLD = ON
- UOPI3 SFSD = ON
- UOPI8 ENBL = ON

This will set UOP CMDENBL and UOP SYSRDY to ON. The robot will also have set the POWER ON digital output to ON. UOPI4 CSTOP, UOPI5 RESET, AND UOPI7 HOME are for special cases and will not be changed for this example.

Time 1

The STYLE SELECT group input is set to the desired style number by the cell controller.

Time 2

After a short delay to allow the STYLE SELECT group input to settle (approximately 36 milliseconds), the UOP CYCLE START input is pulsed by the cell controller.

Time 3

At this time, the robot verifies that the STYLE SELECT group does not contain a 0 and that the JOB number produced by adding STYLE SELECT to the job root (set up in the Cell Setup screen) exists in the system.

The robot WET/DRY RUN status is then set to whatever the WET/DRY RUN digital input is. The following signals are then initialized virtually simultaneously in the following order and states:

- DOUT [CYCLE COMPLETE] = OFF
- DOUT [IN CYCLE] = ON

The new JOB is then executed.

Time 4

Because the updating of UOP output signals is not controlled by DispenseTool, the UOPO3 PROGRAM RUNNING output goes ON approximately 250 ms after the program is executed.

Time 5

This is an arbitrary period during which the user teach pendant program is running. The UOP07 AT PERCH output will turn off within 1 second of the robot leaving the HOME position. During the execution of the user program, any number of errors requiring additional cell controller communication could occur.

Time 6

At the end of the JOB, the END JOB should be called. This will initiate the robot's move to the HOME position by running the teach pendant program MOV_HOME. After the robot has returned to the HOME position, the UOP AT PERCH output will turn ON. The UOP PROGRAM RUNNING output will then turn OFF, as the program completes.

Time 7

If the Cell Setup option "Hold cycle complete high" is DISABLED, the software will pulse CYCLE COMPLETE for the pulse signal width (set up for program select mode in the detail screen of the Program Select SETUP screen). If this feature is ENABLED, CYCLE COMPLETE will be turned ON at this time and not turned OFF until the next JOB begins. IN CYCLE will also be turned OFF at this time.

Time 9

Once the cycle complete signal has been pulsed, the robot waits for instructions from the cell controller telling it to start the next job.

Table 3–14 lists and describes each PNS setup item.

Table 3–14. PNS Setup Item Description

PNS SETUP ITEM	Description
Base Number default: 0 min: 0 max: 9999	This item allows you to enter a number that when added to the PNS1-8 binary signal defines which program will be executed. For example, if the PNS1-8 input is 0023, and the base number is 100, then PNS0123 will be executed. This base number can be changed from within your program by using the PARAMETER NAME instruction. The parameter that contains the base number is**$SHELL_CFG.$pns_base** . By changing the base number, you can control which program will be executed.
Job Prefix — SpotTool+ application default:PNS	This item is the three-letter prefix of the program to be selected. The system variable is $PNS_PROGRAM.
Acknowledge Pulse Width default: 400 ms min: 0 ms max: 9999 ms	This item allows you to set the length (in milliseconds) of the SNO1-8 and SNACK signals. This time depends on the scan time of your PLC program.

Use Procedure 3-17 to set up PNS.

Procedure 3-17 PNS Setup

Conditions

- UOP signals must be installed and configured.

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Prog Select. You will see a screen similar to the following.

```
Prog Select
1 Program select mode:              PNS
2 Production start method           DIN[]
Production checks:
3   At home check:                 ENABLED
4   Resume position toler.:       DISABLED
5   Simulated I/O:                DISABLED
6   General override < 100%:      DISABLED
7   Prog override < 100%:         DISABLED
8   Machine lock:                 DISABLED
9   Single step:                   ENABLED
10  Process ready:                DISABLED
General controls:
11  Heartbeat timing:              1000 MS
12  Low TEMP DRAM memory:          100 KB
13  Low PERM CMOS memory:          50  KB
14  RESET when DEADMAN pressed:    ENABLED
```

5. Move the cursor to Program Select mode and press F4, [CHOICE].
6. Select PNS and press ENTER.
7. Press F3, DETAIL. You will see a screen similar to the following.

```
Prog Select                                 1/2
PNS Setup
1 Base number                             [ 100 ]
2 Acknowledge pulse width(mesc)           [  10]
```

8. Move the cursor to the item you want to set and type the value.
9. **When you are finished** making changes, turn off the controller and then turn it back on.

3.2.4 Style Name Setup

Style name setup allows you to define the style and programs to run during production, based on RSR/PNS/STYLE bits input from the cell controller.

Use Procedure 3-18 to set up Program Style select. Table 3–15 describes the items found on the Style Names Table screen.

During production, the robot receives its STYLE selection bits on the UOP inputs followed by a cycle start input from an external device, such as a PLC. The robot checks the program name that corresponds to the STYLE bits. After the robot validates that all safety and I/O signals have been set properly for a production run, the robot runs the program that corresponds to the style number.

Figure 3–24 and Figure 3–25 show the timing sequence between UOP cell and process signals.

Figure 3–24. Typical Style Initiate Sequence

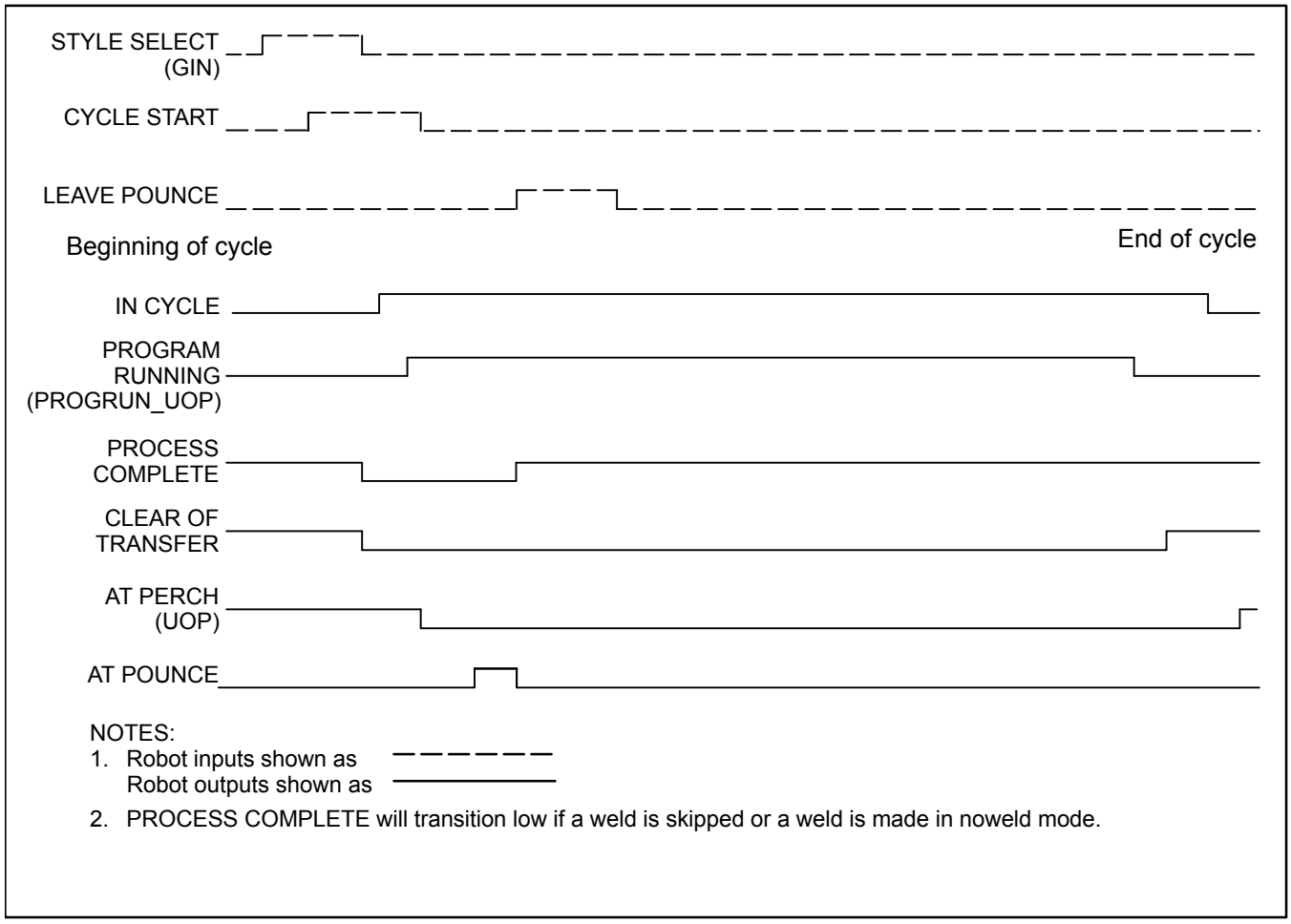
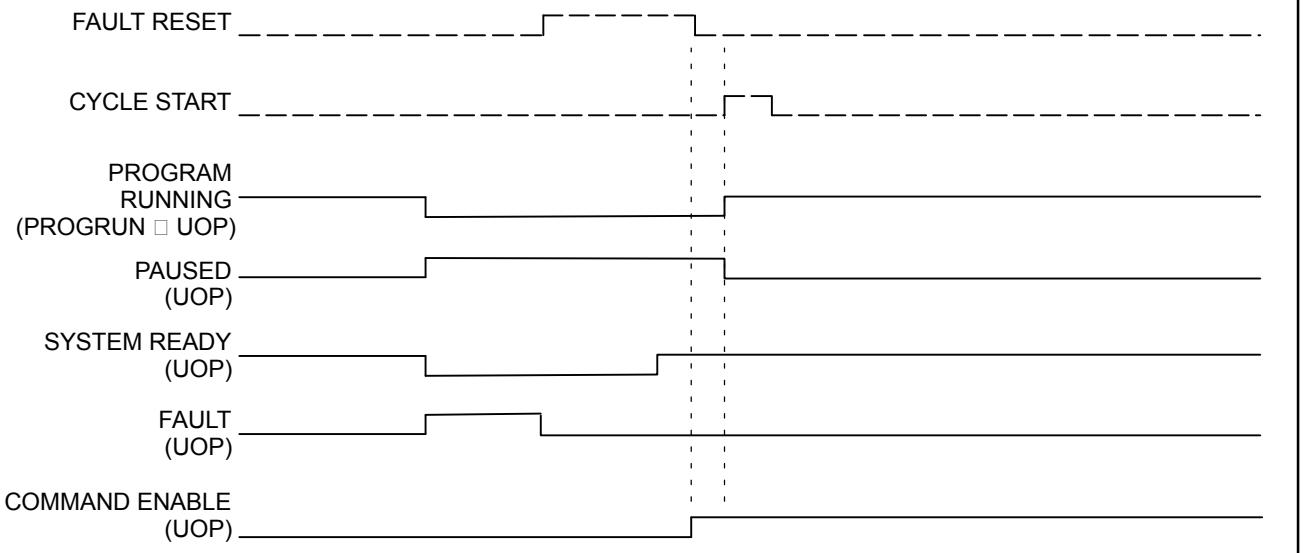


Figure 3–25. Fault Recovery During Style Execution



NOTES:
1. Robot inputs shown as – – – – –
   Robot outputs shown as ————
2. The above sequence would be typical for recovering from an E-Stop type condition.
3. The above sequence assumes that the fault condition has been corrected prior to issuing the fault Reset signal (for example, the E-Stop condition is reset before Fault Reset is issued).
4. After the Fault Reset signal is issued, there is typically a 2□3 second delay before System Ready/Command Enable are set to ON.

Style names are stored in the system variable $STYLE_NAME. Valid flags are stored in the system variable $STYLE_ENABLE. Comments are stored in the system variable $STYLE_COMMENT.

To set up for production operation, you must

- **Configure the RSR/PNS/STYLE UOP inputs (UIN[9–16])** that will receive style numbers from the cell controller.
- **Specify the teach pendant programs** that will run when a style number is received by the robot controller.

Table 3–15. Style Name Setup Items

Style name SETUP ITEMs	Description
Style	This item allows you to specify the style number sent by the cell controller to the robot.
ProgName	This item is the name of the program that is to be run when the associated style number is selected.

Table 3–15. Style Name Setup Items (Cont'd)

Style name SETUP ITEMs	Description
Valid default: yes	This item tells when the style is valid and can be used to run a program. If a style is not valid, it cannot be run, and an error will be displayed.
Comment	This item is a descriptive label for the part being processed.
STYLE CONFIG SETUP	
Acknowledge Function default: false	This item allows you to enable or disable robot acknowledge output signals ACK1-8. FALSE means the signals are disabled. TRUE means the signals are enabled.
Acknowledge Pulse Width (msec): default: 0 min: 0 max: 9999	This item allows you to set the length (in milliseconds) of the ACK1-8 signal when the acknowledge function is enabled. This time depends on the scan time of your PLC program.
Max Number in Style Table default: 32 min: 1 max: 255	This item controls the number of rows in the style table. The length of the $STYLE_NAME, $STYLE_ENABLE, and $STYLE_COMMENT system variables will be adjusted at Controlled start to this value.

Table 3–16. Prog Select STYLE Setup DETAIL Screen Items

ITEMS	DESCRIPTION
Style	This item allows you to specify the style number sent by the cell controller to the robot.
ProgName	This item is the name of the program that is to be run when the associated style number is selected.
Valid default: yes	This item tells when the style is valid and can be used to run a program. If a style is not valid, it cannot be run, and an error will be displayed.
Comment	This item is a descriptive label for the part being processed.
STYLE CONFIG SETUP This item allows you to press the CONFIG function key and display the following items:	
Acknowledge Function default: false	This item allows you to enable or disable robot acknowledge output signals ACK1-8. FALSE means the signals are disabled. TRUE means the signals are enabled.

Table 3–16. Prog Select STYLE Setup DETAIL Screen Items (Cont'd)

Acknowledge Pulse Width (msec) default: min: 0 max: 9999	This item allows you to set the length (in milliseconds) of the ACK1-8 signal when the acknowledge function is enabled. This time depends on the scan time of your PLC program.
Max Number in Style Table default: 32 min: 1 max: 255	This item controls the number of rows in the style table. The length of the $STYLE_NAME, $STYLE_ENABLE, and $STYLE_COMMENT system variables will be adjusted at Controlled start to this value.

Procedure 3-18 Setting Up Style Names

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Prog Select.
5. Move the cursor to Program Select Mode, and press F4, [CHOICE].
6. Select STYLE, and press ENTER.
7. Press F3, DETAIL. You will see a screen similar to the following.

```
Prog Select
Style Table Setup                          1/32
Style   ProgName   Valid   Comment
1     JOB123      YES      2 door
2     RSR001      YES      4 door
3     JB7         YES      Hatchback
4                 YES
5                 YES
Power OFF then ON to enable changes.
```

8. Move the cursor to the style number to which you want to assign a style name.
9. Press F4, [CHOICE]. A list of programs will be displayed. See the following screen for an example.

```
SETUP Style Names
1  Two_Door      5
2  Four_Door     6
3  STYLE1        7
4                8  --NEXT--
SETUP Style Names
Style     Program     Valid    Comment
1         Two_Door     Yes      Short Style
2         Four_Dr      Yes      Long Style
3         ********
4         ********
5         ********
6         ********
```

If the style name you want is not displayed, select 8, --NEXT--. The rest of the programs will be displayed.

10. Move the cursor to the program name you want to select and press ENTER.

11. **To remove a style name from a style number**,

a. Move the cursor to the style name.

b. Press F2, CLEAR.

12. Move the cursor to the valid column, and make the style valid or not valid. The Valid column allows you to de-activate a style without removing it from the style table.

- A YES in the Valid column means this is a valid style
- A NO in the Valid column means this is not a valid style

13. Press F3, CONFIG. You will see a screen similar to the following.

```
Prog Select
Style Config Setup                    1/3
1 Acknowledge function:           [FALSE]
2 Acknowledge pulse width (msec):  [ 400]
3 Max number in style table:        [ 32]
```

14. Set each item as desired.

3.2.5 OTHER Program Select Mode

When you specify OTHER as the program select mode, the program is selected using UOP or OTHER. Also, if the production start method is OTHER, the program is initiated by setting the system variable $SHELL_WRK.$cust_start = 1. You specify the program to run by setting the system variable $SHELL_WRK.$cust_name.

Use Procedure 3-19 to set up the OTHER program select mode.

Table 3–17. Prog Select OTHER DETAIL Screen Items

ITEM	DESCRIPTION
OTHER Program select mode is enabled	This method of selection requires that the application must set the variable $shell_wrk.$cust_name to the desired program name.

Procedure 3-19 Setting Up the OTHER Program Select Mode

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Prog Select.
5. Move the cursor to Program Select Mode and press F4, [CHOICE].
6. Select OTHER and press ENTER.
7. Press F3, DETAIL. You will see a screen similar to the following.

```
Prog Select
OTHER Program select mode is enabled.
With this selection method, the
application must set the variable
$shell_wrk.$cust_name to the desired
program name.
$shell_wrk.$cust_name: ************
```

8. Press F4, [CHOICE].
9. Select the name of the program you want to run and press ENTER.

3.3 MACRO COMMANDS

3.3.1 Overview

A macro command program is a separate program that contains a series of instructions to perform a task. Macro programs can be run by

- Pressing a teach pendant user key
- Selecting an item on the MANUAL FCTNS or the MANUAL Macros screen
- Pressing a button on the operator panel (if available)
- Issuing an input signal: DI, RI, and UI
- Executing an instruction in a program

Refer to Section 6.17 for information on using macro command instructions in a program.

To use a macro command, you must

- Write the macro command program
- Set up the macro command to define how it will be executed
- Execute the macro command program

Note System level macros are application-specific instructions that are predefined and cannot be changed by the user. These macros are identified with the letter "s" at the far right side of the Macro SETUP screen. You cannot change instruction name or program for system level macros on any screen.

3.3.2 Setting Up Macro Commands

Macro commands must be set up before they can be used.

You can select an instruction name and assign a program from the Macro Command screen.

Table 3–18. Macro Command Screen Items

ITEM	DESCRIPTION
Instruction name	This item allows you to select the macro program that you want assigned to the instruction name.
Program Assign	This item allows you to select the macro command assignment.

Teach Pendant User Keys

You can set up a macro command to run when a teach pendant user key is pressed alone or with the SHIFT key. If you want to execute a program that contains robot motion when a user key is pressed, you must set it up to run when the SHIFT key is pressed.



Caution

Make certain that your application has not already assigned functions to the teach pendant user keys; otherwise, execution problems can occur.

When you set up macro commands, you can define up to seven macro commands to run when the user key is pressed alone (UK[1] - UK[7]), and seven macro commands to run when the user key is pressed with the SHIFT key (SU[1] - SU[7]). The macro commands that require the user key to be pressed alone (UK[1] - UK[7]) cannot contain any instructions that move the robot, and the group mask must be set to [*,*,*,*,*] in the program header information.

Note In Table 3–19 , UK indicates that only the key must be pressed - for macro commands that do not include robot motion. SU indicates that SHIFT and the key must be pressed.

Table 3–19. Application-Specific Teach Pendant User Keys

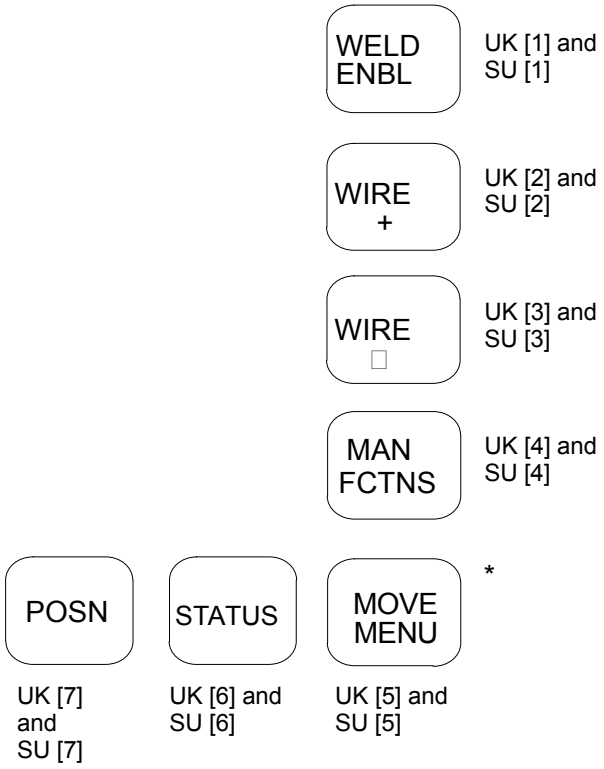
ApplicationTool	Macro Keys
ArcTool	 *OTF on the iPendant

Table 3–19. Application-Specific Teach Pendant User Keys (Cont'd)

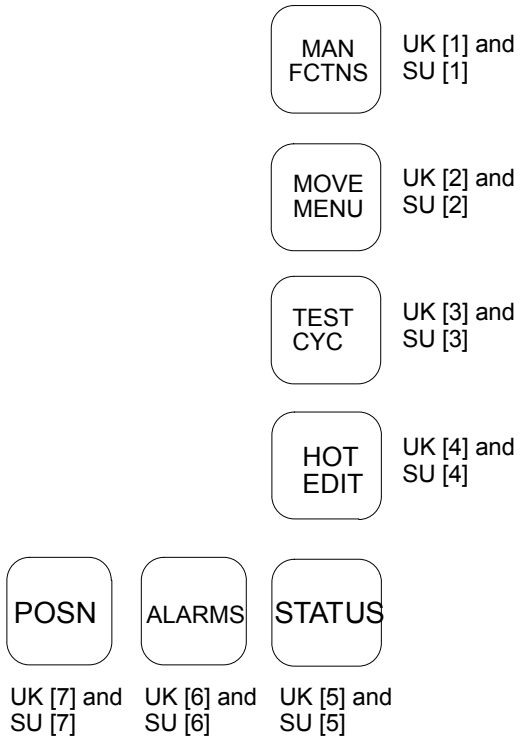
ApplicationTool	Macro Keys
DispenseTool	

Table 3–19. Application-Specific Teach Pendant User Keys (Cont'd)

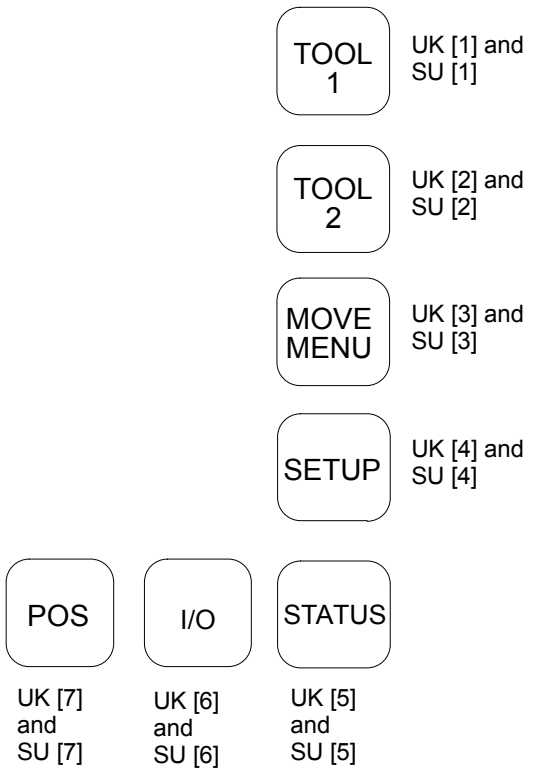
ApplicationTool	Macro Keys
HandlingTool	

Table 3–19. Application-Specific Teach Pendant User Keys (Cont'd)

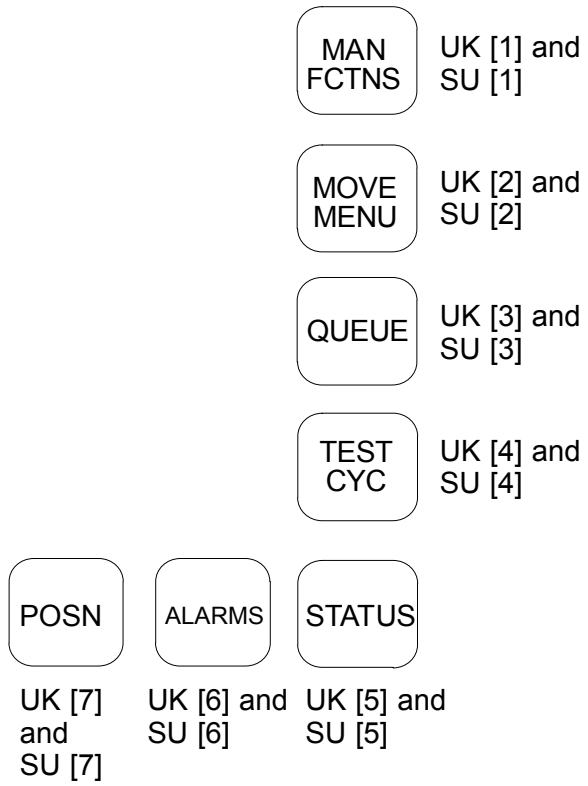
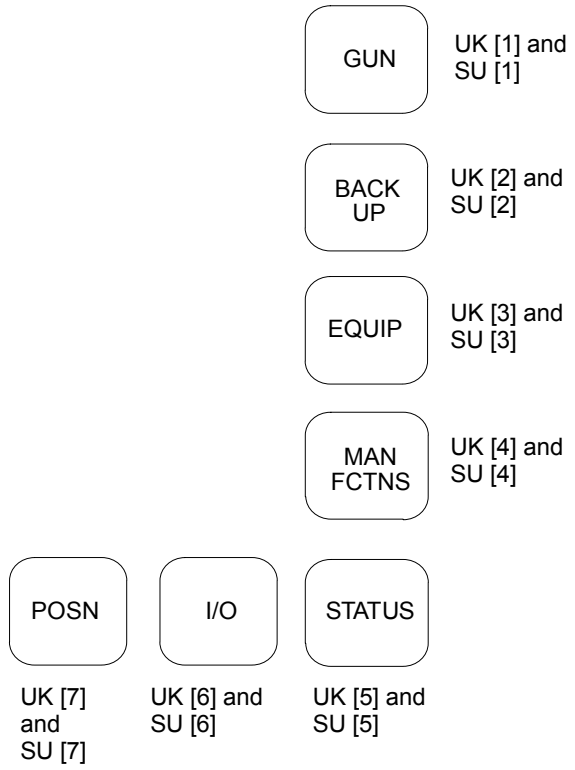
ApplicationTool	Macro Keys
PaintTool	

Table 3–19. Application-Specific Teach Pendant User Keys (Cont'd)

ApplicationTool	Macro Keys
SpotTool+	

MANUAL FCTNS Macro Screen Items

You can set up a macro command program to be executed from the MANUAL FCTNS Macros screen. After you set up a macro command to run from this screen, you can then select a manual functions menu item and press the SHIFT key and the EXEC function key to execute the macro command. Refer to Section 3.3.3 to execute a macro command from the MANUAL FCTNS menu.

Operator Panel Buttons

You can set up a macro command program to run when a button on the operator panel is pressed.

Input Signals

You can set up a macro command program to be executed when the input signal you specify is received. You can assign a macro command to a digital input (DI), robot input (RI) and, in some cases, a user operator panel input (UI). By default, you can assign up to five macro commands as

input signals, or UOP input signals. You can change the number of signals by modifying the value of the $MACROMAXDRI system variable.

For **digital input signals**, indexes 0 through 999 (or the number of digital input signals configured on your system) are available. An index of 0 indicates that no macro is assigned. You can assign any of these index numbers to the macro command, but the digital signal must be configured properly for the macro command to execute.

For **robot input signals**, indexes 0 through the number of robot input signals configured on your system are available. In some cases, this is limited to a maximum of 24. An index of 0 indicates that no macro is assigned.

If **UOP input signals** are available, indexes 0 through the number of UOP input signals configured on your system are available. An index of 0 indicates that no macro is assigned.



Warning

Before you copy a program with embedded macros from one controller to another, compare the SETUP menu macro lists of the two controllers. Be sure that the list on the first controller matches the list on the second controller. If they are not identical, DO NOT copy the program; otherwise, when you run a program that uses macros, you could injure personnel or damage equipment.

Use Procedure 3-20 to set up a macro command.

Procedure 3-20 Setting Up a Macro Command

Conditions

- A macro program has been created. Refer to Section 6.2 .
- The macro program has been tested and runs properly.

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Macro.

If you are using ArcTool, you will see a screen similar to the following.

```
Macro Command
Instruction name  Program  Assign
1 [               ] [      ]--[0]
2 [               ] [      ]  [0]
3 [               ] [      ]  [0]
4 [               ] [      ]--[0]
5 [               ] [      ]--[0]
6 [               ] [      ]--[0]
7 [               ] [      ]--[0]
8 [               ] [      ]--[0]
9 [               ] [      ]--[0]
10 [               ] [      ]--[0]
```

If you are using DispenseTool, you will see a screen similar to the following.

```
Macro Command
Instruction name     Program  Assign
1 [CLR OF TRANSFER  ]  [CLR_TRAN]--[  0]
2 [ENTER I-ZONE     ]  [ENTRZONE]--[  0]
3 [EXIT I-ZONE      ]  [EXITZONE]--[  0]
4 [ENTER I-ZONE 3   ]  [ENTRZON3]--[  0]
5 [ENTER I-ZONE 4   ]  [ENTRZON4]--[  0]
6 [ENTER I-ZONE 5   ]  [ENTRZON5]--[  0]
7 [EXIT I-ZONE 1    ]  [EXITZON1]--[  0]
8 [EXIT I-ZONE 2    ]  [EXITZON2]--[  0]
9 [EXIT I-ZONE 3    ]  [EXITZON3]--[  0]
10 [EXIT I-ZONE 4    ]  [EXITZON4]--[  0]
11 [EXIT I-ZONE 5    ]  [EXITZON5]--[  0]
4 [SAFE ZONE        ]  [SAFEZONE]--[  0]
5 [MOVE TO HOME     ]  [MOV_HOME]MF[  1]
7 [RESERVED: POUNCE ]  [        ]--[  0]
8 [OPEN CLAMP EARLY ]  [OPNCLMER]--[  0]
9 [REPOSITION CLAMP ]  [REPOS_CL]--[  0]
10 [MOVE TO PURGE    ]  [MOV_PURG]MF[  2]
11 [END JOB          ]  [ENDJOB  ]--[  0]
12 [                 ]  [        ]--[  0]
13 [                 ]  [        ]--[  0]


40 [                 ]  [        ]--[  0]
41 [WAIT REPOS DONE  ]  [CHK_RPOS]--[  0]
42 [CHANGE DIRECTION ]  [CHNG_DIR]MF[  4]
43 [METER BYPASS     ]  [ISDBYPAS]MF[ 10]
44 [SET DIRECTION A  ]  [MTR_A   ]--[  0]
45 [SET DIRECTION B  ]  [MTR_B   ]--[  0]
46 [PREPRESSURIZE    ]  [PRESS   ]MF[  8]
47 [RELIEVE PRESSURE ]  [RELV    ]MF[  9]
48 [REPOSITION A     ]  [REPOS_A ]MF[  6]
49 [REPOSITION B     ]  [REPOS_B ]MF[  7]
50 [REPOSITION NEAR  ]  [REPOS_NR]MF[  5]
```

Note Lines 7-42 are available only with the Integral Servo Dispenser option.

If you are using HandlingTool, you will see a screen similar to the following.

```
Macro Command
Instruction name  Program  Assign
1 [               ] [        ]   [ 0]
2 [Hand open      ] [hndopen1]UK[ 1]
3 [Hand close     ] [hndclse1]MF[ 4]
4 [               ] [        ]   [ 0]
5 [               ] [        ]   [ 0]
6 [               ] [        ]   [ 0]
7 [               ] [        ]   [ 0]
8 [               ] [        ]   [ 0]
9 [               ] [        ]   [ 0]
10 [               ] [        ]   [ 0]
```

If you are using PaintTool, you will see a screen similar to the following.

```
Macro Command
Instruction name  Program  Assign
1 [               ] [      ]--[0]
2 [               ] [      ]--[0]
3 [               ] [      ]--[0]
4 [               ] [      ]--[0]
5 [               ] [      ]--[0]
6 [               ] [      ]--[0]
7 [               ] [      ]--[0]
8 [               ] [      ]--[0]
9 [               ] [      ]--[0]
10 [               ] [      ]--[0]
```

If you are using SpotTool+, you will see a screen similar to the following.

```
Macro Command
Instruction name  Program  Assign
1 [CLR OF TRANSFER]  [CLR_TRAN]--[0]
2 [ENTER I-ZONE]     [ENTER1ZON]--[0]
3 [EXIT I-ZONE ]     [EXIT1ZON]--[0]
4 [SAFE ZONE]        [SAFEZONE]--[0]
5 [MOVE TO HOME]     [MOV_HOME]--[0]
6 [MOVE TO REPAIR]   [MOV_REPR]--[0]
7 [AT POUNCE]        [ATPOUNCE]--[0]
8 [OPEN CLAMP EARLY] [OPNCLMER]--[0]
9 [REPOSITION CLAMP] [REPOS_CL]--[0]
```

5. Move the cursor to a blank**Instruction name** and press ENTER.
6. Name the instruction.
   a. Select a method of naming.
   b. Press the appropriate function keys to type a name.
   c. When you are finished press ENTER.

Note Before you perform the next step, you must have a written and tested macro program. Refer to Section 6.2 .

7. **Select the instruction you want to assign:**
   a. Move the cursor to**Program** and press F4, [CHOICE].
   b. Select the macro program you want assigned to the instruction name and press ENTER.
8. **Assign the macro command:**
   a. Move the cursor to**Assign** and press F4, [CHOICE].
   b. Select the macro command assignment you want and press ENTER.

Note You cannot assign macro commands that include motion instructions to UK.

- **For a user key without SHIFT**, select UK.
- **For a user key with SHIFT**, select SU.
- **If you are not using DispenseTool**, select MF for a MANUAL FCTNS menu item.
- **If you are using DispenseTool**, select MF for a MOVE MENU item.

If you are not using PaintTool,

- **For an operator panel button**, select SP.
- **For a digital input**, select DI.
- **For a robot input**, select RI.
- **To remove an assignment**, select **--** .
- **To choose a UOP input in HandlingTool**, select UI.

Table 3–19 lists user key assignment. Table 3–20 lists the ArcTool and HandlingTool Macro Assignments. Table 3–21 for a listing of the PaintTool Macro Assignments. Table 3–22 lists the DispenseTool and SpotTool+ Macro Assignments.

Table 3–20. Macro Command Assignments (ArcTool and HandlingTool)

User Key on Teach Pendant without SHIFT (UK)	User Key on Teach Pendant with SHIFT (SU)	MANUAL FCTNS Menu Item (MF)	Input Signals(DI, RI, UI)
UK [1] UK [2] UK [3] UK [4] UK [5] UK [6] UK [7]	SU [1] SU [2] SU [3] SU [4] SU [5] SU [6] SU [7]	MF [1] - MF[99]	Digital inputs DI[0] - DI[999] Robot inputs RI[0] - RI[24] UOP inputs UI[0] - UI[18] n: number of signals configured in your system

Table 3–21. Macro Command Assignments (PaintTool)

User Key on Teach Pendant without SHIFT (UK)	User Key on Teach Pendant with SHIFT (SU)	MANUAL FCTNS Menu Item (MF)
UK [1]	SU [1]	MF [1]
UK [2]	SU [2]	MF [2]
UK [3]	SU [3]	MF [3]
UK [4]	SU [4]	MF [4]
UK [5]	SU [5]	MF [5]
UK [6]	SU [6]	MF [6]
UK [7]	SU [7]	MF [7]*

* The number of available MANUAL FCTNS menu items varies. The total number of macro commands assigned to all devices must not exceed 20.

Table 3–22. Macro Command Assignments (DispenseTool and SpotTool+)

User Key on Teach Pendant without SHIFT (UK)	User Key on Teach Pendant with SHIFT (SU)	MANUAL FCTNS (SpoTool+) or MOVE MENU (DispenseTool) Screen Item (MF)	Input Signals (DI, RI, UI (SpotTool+))
UK [1] UK [2] UK [3] UK [4] UK [5] UK [6] UK [7]	SU [1] SU [2] SU [3] SU [4] SU [5] SU [6] SU [7]	MF [1] MF [2] MF [3] MF [4] MF [5] MF [6] MF [7]*	Digital inputs DI[0] - DI[99] Robot inputs RI[0] - RI[n] UOP inputs UI[0] - UI[n] n: number of signals configured in your system

* The number of available MOVE MENU items varies. The total number of macro commands assigned to all devices must not exceed 20.

c. Move the cursor to the assignment number, type the number, and press ENTER. The number that you assign to the manual function item defines the item number on the manual functions screen.

Note You cannot modify the instruction name or the program of a system level macro. These macros are identified with the letter "s" at the far right side of the macro SETUP screen.

9. If you want to modify an entry, move the cursor to the item you want to change and enter a new value (or, PRESS F2, CLEAR, to the remove current value from the item and then begin typing).



Caution

When all macro commands have been set up, save the macro information to SYSMACRO.SV and save all *.TP teach pendant program files to the default device. When you reload system software, you should answer no to the "Load macros?" question in SETUP APPLICATION. You will then need to load SYSMACRO.SV and your *.TP teach pendant programs manually.

If you do not save your SYSMACRO.SV file and any teach pendant programs that use macro commands to the default device, if the setup is altered, you will have no record of it.

10. To save the information

a. Press MENUS.
b. Select FILE.
c. Press F1, [TYPE].
d. Select File.
e. Press F5, [UTIL].
f. Select Set Device.
g. Move the cursor to the device you want and press ENTER.
h. Display the macros screen.
i. Press FCTN.
j. Select SAVE. The file will be saved to the SYSMACRO.SV file on the default device.

Note Be sure to save any teach pendant programs that are affected by the macro command assignments you are making.

3.3.3 Executing Macro Commands

3.3.3.1 Overview

After you have set up macro commands you can run them, using one of the following methods:

- Press a teach pendant user key
- Press the SHIFT key and a teach pendant user key
- Select a MANUAL FCTNS (MANUAL Macros in DispenseTool) screen item
- Press an operator panel button (if available)
- Execute a macro program from within another program using the macro command instruction
- Receive an input signal (DI, RI, or UI)
- Execute a macro program by selecting the program, then pressing Shift-FWD on the teach pendant.

The method you use depends on how you set up the macro command to execute. This section describes how to execute a macro command from the MANUAL FCTNS Macros screen (MANUAL Macros screen in DispenseTool). Refer to Section 6.17 for information about the macro command instruction.

Warning

Before you copy a program with embedded macros from one controller to another, compare the SETUP screen macro lists of the two controllers. Be sure that the list on the first controller matches the list on the second controller. If they are not identical, DO NOT copy the program; otherwise, when you run a program that uses those macros, the robot could injure personnel or damage equipment.

3.3.3.2 Teach Pendant User Keys

Use Procedure 3-21 to execute a macro command that has been assigned to a teach pendant user key.

Procedure 3-21 Executing a Macro Command from a Teach Pendant User Key

Conditions

- The program you want to use as a macro command has been tested.
- The macro command has been set up to execute when a teach pendant user key is pressed.

Steps

1. Make sure the teach pendant is ON and the DEADMAN switch is pressed.

Warning

In the next step, the robot could move. Make sure that personnel and unnecessary equipment are out of the workcell; otherwise, the robot could injure personnel or damage equipment.

2. Press the teach pendant user key that corresponds to the macro command you assigned. If you assigned the key to be pressed with the SHIFT key, press and hold SHIFT and press the user key. Refer to Table 3–19 .

3.3.3.3 MANUAL FCTNS Menu Items

You can execute preassigned macros from the MANUAL Macros screen.

Refer to Table 3–23 for a description of the Manual Macros Screen Items. Use Procedure 3-22 to execute a macro command that has been assigned to a MANUAL FCTNS menu item.

Table 3–23. MANUAL Macros Screen Items

ITEM	DESCRIPTION
Instruction	This item is a listing of preassigned macros that you can execute. To execute a macro, 1. Move the cursor to the line containing the macro. 2. Continuously press and hold the DEADMAN switch and turn the teach pendant ON/OFF switch to ON. 3. Press and hold the SHIFT key and press F3, EXEC. The F3 key can be released,but the SHIFT key must be held continuously until the instruction has completed executing.

Procedure 3-22 Executing a Macro Command from the MANUAL FCTNS Menu

Conditions

- The program you want to use as a macro command has been tested.
- The macro command has been set up to execute when an item on the MANUAL FCTNS screen is selected.

Steps

1. Press MENUS (MAN FCTNS if you are using DispenseTool).
2. Select MANUAL FCTNS.
3. Press F1, [TYPE].
4. Select Macros.

If you are using ArcTool, you will see a screen similar to the following.

```
Manual Macros
Instruction
1  Clean torch
2  Change torch
```

If you are using DispenseTool, you will see a screen similar to the following.

```
MANUAL Macros
Instruction
1  MOVE TO HOME
2  MOE TO PURGE
3  MOVE TO REPAIR
4  CHANGE TO DIRECTION
5  REPOSITION NEAR
6  REPOSITION A
7  REPOSITION B
8  PREPRESSURIZE
9  RELIEVE PRESSURE
10  METER BYPASS
Press SHIFT-EXEC (F3) to run program
```

If you are using HandlingTool, you will see a screen similar to the following.

```
Manual Macros
Instruction
1  OPEN HAND
2  CLOSE HAND
```

If you are using PaintTool, you will see a screen similar to the following.

```
MANUAL Macros
Instruction
1  Tip_DRESS                Not Assign
```

If you are using SpotTool+, you will see a screen similar to the following.

```
Manual Macros
Instruction
1  MOVE TO HOME
2
3
4
5
6
```

5. Select an item on the menu.
6. Continuously press and hold in the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.

> **Warning**
>
> **In the next step, the robot could move. Make sure that personnel and unnecessary equipment are out of the workcell; otherwise, you could injure personnel or damage equipment.**

7. Press and hold the SHIFT key and press F3, EXEC. The F3 key can be released, but the SHIFT key must be held continuously until the instruction has completed executing.

Note If the SHIFT key is released, the Macro program is aborted and cannot be resumed.

3.3.4 Assigning I/O and Recording Positions in Macro Command Programs

You must assign the I/O for the macro programs to match the I/O signals defined for workcell I/O. In addition, some macro command programs require you to record specific positions. Use Procedure 3-23 to assign I/O and record positions in macro command programs.

Note Chapter 5 *PLANNING AND CREATING A PROGRAM* contains detailed information about modifying a program.

Procedure 3-23 Assigning I/O and Recording Positions in Macro Command Programs

Conditions

- All personnel and unnecessary equipment are out of the workcell.

Steps

1. Press SELECT.
2. Press F1, [TYPE].
3. Select Macros.
4. Select the predefined macro program name you want to use.
5. Press EDIT.
6. Continuously press the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.
7. **To assign I/O,**
   a. Move the cursor to the I/O instruction.
   b. Move the cursor to the I/O number.

c. Type the appropriate signal number.

8. **To record a position,** insert the appropriate number of motion instructions and record the positions.
9. Continue modifying instructions as necessary.
10. Turn the teach pendant ON/OFF switch to OFF and release the DEADMAN switch.

The following predefined macro command programs are assigned to the following manual function menu items:

- **MOVE TO HOME** - MF [1]

Use Procedure 3-20 to set up a macro command to run from the Manual Functions screen.

3.3.5 HandlingTool Macro Commands

The HandlingTool software provides six predefined macro commands. These are:

Teach Pendant Keys TOOL 1 and TOOL 2

- Open hand
- Close hand
- Relax hand
- Open hand
- Close hand
- Relax hand

All of the hand 1 macros have been defined to work with the teach pendant key TOOL 1. All of the hand 2 macros have been defined to work with the teach pendant key TOOL 2.

You can set up more macro commands for anything you would like to be able to do manually, for example, to repair or maintain equipment.

Before you can use these six pre-defined macro commands, you must;

- Create and write a macro program for each macro command.
- Make sure the macro programs control the RI/RO so that the hand
  - — Opens
  - — Closes
  - — Relaxes

- Assigned each macro program to its corresponding macro command. You do this by using the Macros screen.

Procedure 3-24 Using the Teach Pendant Keys TOOL 1 and TOOL

Conditions

- A macro program has been created and written for each pre-defined macro command that you want to use.
- The macro programs have been assigned to their corresponding macro commands.

Steps

1. Press either TOOL1 or TOOL2. You will see a screen similar to the following.

```
Manual Macros
Instruction
1  OPEN HAND 1
2  CLOSE HAND 1
3  RELAX HAND 1
```

2. Move the cursor to the macro command you want to perform.
3. Press and hold in the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.
4. Press and hold the SHIFT key and press F3, EXEC. The F3 key can then be released, but the SHIFT key must be held continuously until the instruction has completed executing.

3.4 AXIS LIMITS SETUP

Axis limits define the motion range of the robot. The operating range of the robot axes can be restricted because of:

- Work area limitations
- Tooling and fixture interference points
- Cable and hose lengths

There are three methods used to prevent the robot from going beyond the necessary motion range. These are

- Axis limit software settings
- Axis limit switches - optional

- Axis limit hardstops

> **Warning**
>
> **Do not use axis software limits as the only method for restricting robot motion. Change the hard stops to match the software modifications; otherwise, you could injure personnel or damage equipment.**

Software Settings

Axis limit **software settings** are upper and lower motion degree limitations. The limits can be set for all robot axes and will stop robot motion if the robot is calibrated. If the robot is not calibrated, overtravel limit switches or hardstops are contacted two to three degrees beyond the software limits. Overtravel switches for axis 1 are available as an option.

Limit Switches

Axis **limit switches** are overtravel switches that, when tripped, cut power to the servo motors. These are located two or three degrees beyond the software limits. Overtravel switches for axis 1 are available as an option.

Hardstops

Axis limit **hardstops** are physical barriers that are located two or three degrees beyond the overtravel limit switch or software setting on the three major axes. The robot cannot move beyond a hardstop.

Setting the axis limits software settings changes the motion range of the robot. The axis limit screen displays the current upper and lower axis limits, for each robot axis, in degrees.

Upper Limits

Displays the upper limits of each axis, or the axis limits, in a more positive direction.

Lower Limits

Displays the lower limits of each axis, or the axis limits, in a more negative direction.

Saving Limits

After you change the axis limits, turn off the controller and then turn it on again so the new settings can be used.

Caution

Changing the axis limits will affect the robot work area, and could change robot motion. Anticipate the effects of changing axis limits before you change them; otherwise unexpected results could occur, such as errors in previously recorded positions.

Refer to Table 3–24 for information on the SYSTEM Axis Limit items.

Use Procedure 3-25 to set up axis limits.

Table 3–24. SYSTEM Axis Limit Items

ITEM	DESCRIPTION
AXIS	This item displays the number for each axis.
GROUP	This item indicates to which group an axis belongs. A "0" indicates the robot does not have these axes.
LOWER	This item displays the lower limits of each axis, or the axis limits, in a more negative direction. You can change these values.
UPPER	This item displays the upper limits of each axis, or the axis limits, in a more positive direction. You can change these values.

Procedure 3-25 Setting Up Axis Limits

Steps

1. Press MENUS.
2. Select SYSTEM.
3. Press F1, [TYPE].
4. Select Axis Limits. See the following screen for an example of the axis limits for an M-16i robot. The values for your robot could be different.

```
SYSTEM Axis Limits
AXIS    GROUP       LOWER     UPPER
1       1        -165.00     165.00 dg
2       1         -78.00     162.00 dg
3       1        -170.50     285.00 dg
4       1        -200.00     200.00 dg
5       1        -140.00     140.00 dg
6       1        -450.00     450.00 dg
7       0           0.00       0.00 mm
8       0           0.00       0.00 mm
9       0           0.00       0.00 mm
```

Note A "**0** " indicates the robot does not have these axes.

5. Move the cursor to the axis limit you want to set, and press ENTER.

> **Warning**
>
> **Do not depend on axis limit software settings to control the motion range of your robot. Use the axis limit switches or hardstops also; otherwise, you could injure personnel or damage equipment.**

6. Type the new value using the numeric keys on the teach pendant.

7. Repeat Step 5 through Step 6 until you are finished setting the axis limits.

> **Warning**
>
> **You must turn off the controller and then turn it back on to use the new information; otherwise, you could injure personnel or damage equipment.**

8. Turn off the controller and then turn it back on again so the new information can be used.

3.5 BRAKE TIMERS SETUP

Brake timers define the amount of time the robot remains idle (i.e. no robot motion) before the brakes are applied and servo power is removed. Brake timers are specified in milliseconds. For example, if you want the timer to be set to 2 seconds, you must set it to a value of 2000.

The brake timers are always operational even when a program is running. After the brake timers activate, the servo power will remain off and the brakes will remain applied until the next motion command is issued, at which time servo power will be turned back on and the brakes will be released.

After you set the brake timers, you must turn off the controller and then turn it back on again so the new information can be used.

Use Procedure 3-26 to set brake timers.

Note If you are using PaintTool with the Line/Rail Tracking feature, do not set brake timers.

Procedure 3-26 Setting Brake Timers

Steps

1. Press MENUS.

2. Select SYSTEM.
3. Press F1, [TYPE].
4. Select Variables. You will see a screen similar to the following.

```
SYSTEM Variables
285 $RSR             [8] of BYTE
286 $RSR_OPTION      1
287 $SBR             [16] of SBR_T
288 $SCAN_TIME       16
289 $SCR             SCR_T
290 $SCR_GRP         SCR_GRP_T
291 $SEL_DEFAULT     16
292 $SEMIPOWERFL     FALSE
293 $SEMIPWFDO       0
294 $SERVENT         [12] of SERVENT_T
```

To move quickly through the information, press and hold down the SHIFT key and press the down or up arrow keys.

5. Determine which brakes control each axis:
   a. Move the cursor to $SCR_GRP and press ENTER.

```
SYSTEM Variables
$SCR_GRP                          1/1
1   [1]           SCR_GRP_T
```

   b. If you have more than one motion group, select the motion group number of the axes and press ENTER. If you have only one motion group, select group 1 and press ENTER.

```
SYSTEM Variables
$SCR_GRP[1]
6 $SYNC_M_AXIS    0
7 $SYNC_S_AXIS    0
8 $WRIST_TYPE     0
9 $HW_STRT_AXS    1
10 $AXISORDER      [9] of BYTE
11 $DUMMY91        255
12 $BRK_NUMBER     [9] of BYTE
13 $DUMMY92        255
14 $DD_MOTOR       [9] of BOOLEAN
15 $ROTARY_AXS     [9] of BOOLEAN
```

c. Move the cursor to $BRK_NUMBER and press ENTER.

```
SYSTEM Variables
$SCR_GRP[1].$BRK_NUMBER
1   [1]           1
2   [2]           1
3   [3]           1
4   [4]           1
5   [5]           1
6   [6]           1
7   [7]           0
8   [8]           0
9   [9]           0
```

d. Determine the brakes that control each axis.

e. Press PREV **three times**, or until the first system variable screen is displayed.

```
SYSTEM Variables
205/368
199 $ORG_DSBL       [16] of BYTE
200 $ORIENTTOL      10.000
201 $OVRDSLCT       OVRDSLCT_T
202 $OVRD_PEXE      FALSE
203 $OVRD_RATE      5
204 $PADJ_SCHNUM    10
205 $PARAM_GROUP    MRR_GRP_T
206 $PARAM_MENU     [21] of STRING[21]
207 $PASSNAME       [10] of PASSNAME_T
208 $PASSSUPER      PASSNAME_T
```

6. Make sure that the brakes are enabled for the axes you want to control:

a. Move the cursor to $PARAM_GROUP and press ENTER.

```
SYSTEM Variables
$PARAM_GROUP                    1/1
1   [1]           MRR_GRP_T
```

b. If you have more than one motion group, select the motion group number of the axes and press ENTER. If you have only one motion group, select 1 and press ENTER.

```
SYSTEM Variables
$PARAM_GROUP[1]                  46/227
42 $STOPERLIM      [9] of INTEGER
43 $STOPTOL        [9] of INTEGER
44 $SERVO_CTRL     0
45 $SV_OFF_ALL     TRUE
46 $SV_OFF_ENB     [9] of BOOLEAN
47 $SV_OFF_TIME    [9] of INTEGER
48 $UPPERLIMS      [9] of REAL
49 $UPPERLIMSDF    [9] of REAL
50 $TRKERRLIM      524288
51 $PAYLOAD        130
```

c. Move the cursor to $SV_OFF_ENB and press ENTER.

```
SYSTEM Variables
$PARAM_GROUP[1].$SV_OFF_ENB       1/9
1    [1]           TRUE
2    [2]           TRUE
3    [3]           TRUE
4    [4]           TRUE
5    [5]           TRUE
6    [6]           TRUE
7    [7]           FALSE
8    [8]           FALSE
9    [9]           FALSE
```

d. Move the cursor to the axis you want to check and set.

The number in the left column is the axis number. The number in the far right column is the value. For example, the value for axis 3 is TRUE.

If the value is TRUE, the brakes are enabled for the axis and you can define a brake timer.

If the value is FALSE, the brakes are not enabled for the axis and you cannot define a brake timer. Press F4, TRUE and press ENTER.

e. Press PREV **three times**, or until the first system variable screen is displayed.

7. Set the brake timer for the axes you want:

a. Move the cursor to $PARAM_GROUP and press ENTER.

```
SYSTEM Variables
$PARAM_GROUP                        1/1
1   [1]           MRR_GRP_T
```

b. If you have more than one motion group, select the motion group number of the axes and press ENTER. If you have only one motion group, select 1 and press ENTER.

```
SYSTEM Variables
$PARAM_GROUP[1]                   46/227
42 $STOPERLIM      [9] of INTEGER
43 $STOPTOL        [9] of INTEGER
44 $SERVO_CTRL     0
45 $SV_OFF_ALL     TRUE
46 $SV_OFF_ENB     [9] of BOOLEAN
47 $SV_OFF_TIME    [9] of INTEGER
48 $UPPERLIMS      [9] of REAL
49 $UPPERLIMSDF    [9] of REAL
50 $TRKERRLIM      524288
51 $PAYLOAD        130
```

c. Move the cursor to $SV_OFF_TIME and press ENTER.

```
SYSTEM Variables
$PARAM_GROUP[1].$SV_OFF_TIME      1/9
1   [1]           20000
2   [2]           20000
3   [3]           20000
4   [4]           20000
5   [5]           20000
6   [6]           20000
7   [7]           20000
8   [8]           20000
9   [9]           20000
```

The number in the left column is the axis number. The number in the right column is the time the robot remains idle before brakes are applied.

d. Select an axis, type the new time (in milliseconds), and press ENTER.

Note If the same brake controls multiple axes, and you set brake timers for more than one of these axes, the **shortest** brake timer is effective.

Warning

You must turn off the controller and then turn it back on to use the new information; otherwise, you could injure personnel or damage equipment.

8. When you are finished, turn off the controller and then turn it back on so the new information can be used.

3.6 SETUP GENERAL SETUP SCREEN

3.6.1 Overview

You can access various SETUP items from the SETUP GENERAL screen.

Table 3–25. SETUP General Screen Items

ITEM	DESCRIPTION
Brake on hold	This item defines whether the robot brakes are engaged (enabled) or disengaged (disabled) when the robot is placed in a hold condition.
Current language	This item allows you to change the current language that is used in teach pendant screens.
Ignore Offset command	This item ignores the position offset specified by an OFFSET instruction.
Ignore Tool_offset	This item ignores the position offset specified by a TOOL OFFSET instruction.

3.6.2 Brake on Hold Setup

Brake on hold defines whether the robot brakes are engaged (enabled) or disengaged (disabled) when the robot is placed in a hold condition. The available settings are summarized in Table 3–26 . Use Procedure 3-27 to set brake on hold.

Table 3–26. Brake On Hold Settings

BRAKE ON HOLD SETTING	DESCRIPTION
DISABLED	The brakes**are not applied** when the robot is in a hold condition.
ENABLED	The brakes**are applied** when the robot is in a hold condition after a period of time.

Warning

Not all axes have brakes. Enabling Brake on Hold has NO EFFECT on axes that do not have brakes. Make certain that you understand which axes have brakes before you enable Brake on Hold; otherwise, you could injure personnel or damage equipment.

Procedure 3-27 Setting Brake On Hold

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select General. You will see a screen similar to the following.

```
SETUP General
1 Brake on hold:           DISABLED
2 Current language:        DEFAULT
3 Ignore Offset command:   DISABLED
4 Ignore Tool_offset:      DISABLED
```

5. Move the cursor to Brake on hold.

   Note Brake on Hold is disabled by default.

6. Enable or disable the brake on hold:

   - **To enable the brake on hold,** press F4, ENABLED.
   - **To disable the brake on hold,** press F5, DISABLED.

7. Turn the controller off and back on.

3.6.3 Current Language Setup

Current language allows you to change the current language that is used in teach pendant screens. You can select from only those languages that have dictionaries.

Use Procedure 3-28 to set the current language.

Procedure 3-28 Setting Current Language

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select General. You will see a screen similar to the following.

```
SETUP General
1 Brake on hold:           DISABLED
2 Current language:        DEFAULT
3 Ignore Offset command:   DISABLED
4 Ignore Tool_offset:      DISABLED
```

5. Move the cursor to Current language.
6. Press F4, [CHOICE].
7. Select the language.

3.6.4 Ignore Offset Setup

This function ignores the position offset specified by an OFFSET instruction.

- When this function is disabled, the robot moves to the position to which a position offset has been applied (default setting).
- When this function is enabled, the robot moves to the taught position (position offset is not applied).

For details of the OFFSET instructions, refer to Section 6.4.10 and Section 6.14 .

Use Procedure 3-29 to set ignore offset.

Procedure 3-29 Setting Ignore Offset

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select General. You will see a screen similar to the following.

```
SETUP General
1 Brake on hold:           DISABLED
2 Current language:        DEFAULT
3 Ignore Offset command:   DISABLED
4 Ignore Tool_offset:      DISABLED
```

5. Move the cursor to Ignore Offset command.
6. Enable or disable the Ignore Offset command:
   - To enable this, press F4, ENABLED.
   - To disable this, press F5, DISABLED.
7. Turn the controller off and back on.

3.6.5 Ignore Tool Offset Setup

This function ignores the position offset specified by a TOOL OFFSET instruction.

- When this function is disabled, the robot moves to the position for which a tool offset has been applied (default setting).
- When this function is enabled, the robot moves to the taught position (tool offset is not applied).

For details of the tool offset instructions, refer to Section 6.4.10 and Section 6.15 .

Use Procedure 3-30 to set ignore tool offset.

Procedure 3-30 Setting Ignore Tool Offset

Steps

1. Press MENUS.

2. Select SETUP.
3. Press F1, [TYPE].
4. Select General. You will see a screen similar to the following.

```
SETUP General
1 Brake on hold:           DISABLED
2 Current language:        DEFAULT
3 Ignore Offset command:   DISABLED
4 Ignore Tool_offset:      DISABLED
```

5. Move the cursor to Ignore Tool_offset.
6. Enable or disable the Ignore Tool_offset:
   - To enable this, press F4, ENABLED.
   - To disable this, press F5, DISABLED.

3.7 USER ALARM SETUP

The Setting User Alarm screen allows you to define a message that will be displayed on the teach pendant status line. This message is displayed when a user alarm instruction is executed in a teach pendant program.

For example, if you define the message of user alarm 1 (UALM[1]) to be "Perform repair procedure," and the instruction UALM[1] is executed in a teach pendant program, then the message will be displayed on the status line of the teach pendant as:

INTP-213 Perform repair procedure (name, line) UALM[1]

where*name* is the name of the current program and *line* is the line number in which the UALM[1] instruction was executed. Refer to Section 6.11.3 for more information on the user alarm instruction. Use Procedure 3-31 to set the user alarm.

User Alarm Severity

By default, the severity of a user alarm is STOP, which pauses the program and stops robot motion. If you want to change the severity of the user alarm, you must set the appropriate $UALRM_SEV[n] system variable to a value that corresponds to the severity you want. "n" corresponds to the number of the user alarm.

Refer to Table 3–27 for the actions associated with $UALRM_SEV[n] values. Use Procedure 3-32 to set user alarm severity.

Table 3–27. $UALRM_SEV[n] Severity Values

Value	Action
0	No action
2	Pause program
3	Abort program with error
4	Stop program motion
6	Pause program and stop its motion
8	Cancel program motion
10	Pause program and cancel its motion
11	Abort program and cancel its motion

16 added to any value causes servomotors to be turned off.

32 added to any value causes the action to apply to all programs and all motions.

64 added to any value requires a Cold start to reset the controller.

For example,

- A value of 0 causes a warning message to be displayed.
- A value of 6 pauses the program and stops its motion.
- A value of 43 aborts all programs and cancels all motions (11 + 32)

Table 3–28. Setting/User Alarm Screen Items

ITEM	DESCRIPTION
Alarm No.	This item allows you move the cursor to the user alarm message you want.
User Message	This item allows you to select the appropriate function keys to add the message.

Procedure 3-31 Setting the User Alarm

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select User Alarm. You will see a screen similar to the following.

```
Setting/User Alarm
Alarm No.        User Message
[1]:      [                              ]
[2]:      [                              ]
[3]:      [                              ]
[4]:      [                              ]
[5]:      [                              ]
[6]:      [                              ]
[7]:      [                              ]
[8]:      [                              ]
[9]:      [                              ]
```

5. Move the cursor to the user alarm message you want to set and press ENTER.
6. To make the message:
   a. Select a method of naming the message.
   b. Press the appropriate function keys to add the message. The alarm message can contain up to 29 characters. The amount of alarm message displayed will vary depending on the number of characters in the program name.
   c. When you are finished, press ENTER.
7. **If you want to set the severity**, perform Procedure 3-32 .
8. Add the corresponding user alarm instruction to the program. Refer to Section 6.11.3 . The alarm and message will be displayed when the instruction is executed in test cycle or production.

Procedure 3-32 Setting the User Alarm Severity

Steps

1. Press MENUS.
2. Select SYSTEM.
3. Press F1, [TYPE].
4. Select Variables. You will see a screen similar to the following.

```
SYSTEM Variables
351 $TP_USESTAT     FALSE
352 $TSR_GRP        TSR_GRP_T
353 $TX             TX_T
354 $UALRM_MSG      [10] of STRING[29]
355 $UALRM_SEV      [10] of BYTE
356 $UMR            UMR_T
357 $UNDO_CFG       UNDO_CFG_T
358 $USEUFRAME      TRUE
359 $VDATE          '01/16/200x'
360 $VERSION        'Vx.x041      01/15/
```

5. Move the cursor to $UALRM_SEV and press ENTER.

```
SYSTEM Variables
$UALRM_SEV                          1/10
1    [1]            6
2    [2]            6
3    [3]            6
4    [4]            6
5    [5]            6
6    [6]            6
7    [7]            6
8    [8]            6
9    [9]            6
10    [10]           6
```

6. Move the cursor to the number that corresponds to the number of the user alarm for which you want to set the severity.

7. Type the number that corresponds to the severity you want and press ENTER. Refer to Table 3–27 for a list of values.

3.8 OVERRIDE SELECT SETUP

Override select setup allows you to specify four different speed limiting percentages for production operation. When enabled, override select is in effect when the teach pendant is disabled and the system is ready to run production.

Using Override Select

You specify two digital inputs to control override select. The four combinations of the values of these digital inputs (ON ON, ON OFF, OFF OFF, OFF ON) correspond to four override percentages. To use override select, you must do the following:

1. Define the digital input signals that will be used.
2. Specify the override percentage that corresponds to each of the four digital input value combinations.
3. Enable or activate the override select function when you are ready to use it.

Effect of Override Select

When override select is enabled, the following occurs:

- The jog speed keys on the teach pendant are practically disabled, as follows: when you use these keys to change the speed value, the value is quickly changed back to the value set by override select.
- The override instruction has no effect on the speed value set by override select.
- You cannot change the settings of the digital input signal number and override. If you want to change these settings, disable override select in advance.
- If override select is enabled when controller power is turned off, when the controller is turned on again, the speed will return to the value set by override select.
- It is possible to specify the same number as two digital input signal numbers. In this case, only the combinations ON ON and OFF OFF are meaningful.
- If override select is disabled by setting the REMOTE/LOCAL item on the System Config screen to LOCAL, the speed stays at the override select value until it is changed by the teach pendant jog speed keys or the override instruction.

Override Select Setup

Table 3–29 lists and describes the items on the override select screen you must set.

Table 3–29. Override Select Menu Listing

ITEM	DESCRIPTION
Function Enable	This item allows you to specify whether or not the override select will function. When set to **ENABLE** , the override select will limit the speed of the robot during production operation. When set to **DISABLE** , the speed of the robot will not be clamped during production operation. You must set Function Enable to **DISABLE** before the Override select percentage can be changed.

Table 3–29. Override Select Menu Listing (Cont'd)

ITEM	DESCRIPTION
Signal 1 DI	This item allows you to specify the digital input (DI) signal whose status (ON or OFF) determines, along with the status of Signal 2 DI, which of the four override selects are used to clamp the speed of the robot during production operation.
Signal 2 DI	This item allows you to specify the digital input (DI) signal whose status (ON or OFF) determines, along with the status of Signal 1 DI, which of the four override selects are used to clamp the speed of the robot during production.
Signal 1 ON/OFF	This item indicates how the status of the two digital inputs determines which of the four override selects are used to clamp the speed of the robot during production operation.
Signal 2 ON/OFF	This item indicates how the status of the two digital inputs determines which of the four override selects are used to clamp the speed of the robot during production operation.
Override	This item allows you to enter the override select percentage. You must set Function Enable to **DISABLE** before the override select percentage can be changed.

Use Procedure 3-33 to set up the override select.

Procedure 3-33 Setting Up Override Select

Conditions

- You have set up the digital input signals you want to use for override select.

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Ovrd Select. You will see a screen similar to the following.

```
OVERRIDE SELECT
1 Function Enable: DISABLE
2 Signal 1:           DI[  1] [ ON]
3 Signal 2:           DI[ 32] [OFF]
Signal 1      Signal 2    Override
4    OFF           OFF          10%
5    OFF           ON           10%
6    ON            OFF          10%
7    ON            ON           10%
```

5. Make sure Function Enable is set to DISABLE. If it is set to ENABLE, move the cursor to Function Enable and press F5, DISABLE.
6. **To specify the digital Signal 1 or Signal 2**, move the cursor to Signal or Signal 2 and enter the digital input number.
7. **To specify the override select percentage**, move the cursor to the override percentage of each combination and enter the new number.
8. **To enable the override select function**, move the cursor to Function Enable and press F4, ENABLE.
9. Turn the controller off and back on.

3.9 PASSWORD SETUP

3.9.1 Password Operations Overview

A *password* is a combination of up to 12 letters, numbers, and symbols, used to allow authorized personnel access to various operations and screens. The password feature can be an option and might not be used at your site. Password protection is inactive unless the password feature is installed and the Install user is defined.

Eight password *levels* provide access to specific operations and menus. Levels 3 — 7 are user-defined levels. Table 3–30 summarizes the four levels of password authorization.

Table 3–30. Password Levels

Level	Operations
Install	Assigns user **names**, **passwords**, and **levels**. Clears usernames and passwords. Disables and enables the Password Log. Sets the number of Password users in the system. Can perform all Setup, Program, and Operator operations. **Note** There can be only **one** install user.
Setup	Performs operations typically used to set up your system.
Program	Performs more advanced operations.
Operator	Performs basic operations.
User-Defined	Performs basic operations unless defined otherwise in the Password Configuration File.

Note By default, the system is set to Operator level access when the controller is turned on.

Caution

If you do not know the Install password, you will be unable to perform several functions. Contact your FANUC Robotics technical representative if you lose or forget your Install password.

Password Operations

If you want to use passwords, you must first identify the Install User for your site. The Install user must assign the Install username and password and then *log in*. After logging in, the Install user *assigns* usernames, levels, and passwords for each user.

Note No passwords can be used until the Install username and password are assigned.

After the Install User assigns your username, password level, and password, you must *log in* to work at your assigned level. When you log in, you select your username and type your password. Only one user can be logged in for a given device. Valid devices include the following: teach pendant, CRT/keyboard, and KCL.

When you are finished working, you should *log out*. If you do not log out, the system will *timeout* in the number of minutes specified as the Default User Timeout. After the Default User Timeout expires, or you log out, the system reverts to the Operator level and other users can log in. If you forget to log out, other users can log you out.

If Log events is set to ENABLE by the Install User on the SETUP Passwords screen, password information is logged on the ALARM screen. The Password Log contains information about changes to important data, which user made the changes, and when the changes were made. Refer to Procedure 3-41 .

If you are the Install User, refer to Section 3.9.2 for information on assigning usernames, password levels, and passwords. If you are an Operator, Program, or Setup User, refer to Section 3.9.4 .

3.9.2 Install User Password Operations

The Install User must:

- Assign the Install username and password (Procedure 3-34)
- Assign usernames, levels, and passwords for all other users (Procedure 3-34)
- Enable, disable, and display the Password Log (Procedure 3-40 and Procedure 3-41 in Section 3.9.6)

Use Procedure 3-34 to assign password levels.

Procedure 3-34 Assigning Usernames and Default Passwords for each Password Level

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Passwords. You will see a screen similar to the following.

```
SETUP Passwords
Current user:                  None
Current level:                 OPERATOR
Default user timeout:          15 min
Timeout occurs in:             15 min
Log events:                    DISABLE
Number of users:               10
```

5. Press F2, LOGIN. You will see a screen similar to the following.

```
SETUP Passwords
USERNAME    PWD     LEVEL    TIME(min)
1             *       INSTALL      0
```

Note The Install username and password **must** be set up first.

6. **Assign the Install username and password:**
   a. Press ENTER.
   b. Use the arrow and function keys to type the Install username. When you are finished, press ENTER. You will see a screen similar to the following.

```
1  Uppercase
2  Lower Case
3  Punctuation
4  Options                        --Insert--
SETUP Passwords
--Set password for BOB
Old password:   '          '
New password:   '          '
Verification:   '          '
Old Value:
```

Note The password must contain at least three characters.



Caution

Make a written note of the Install password. If you do not know the Install password, you will be unable to perform several functions. Contact your FANUC Robotics technical representative if you lose or forget your Install password.

c. Type the new password and press ENTER.

d. Type the new password again to verify that the first one is correct and press ENTER. You will see a screen similar to the following.

```
Would you like to be logged in?[YES]
YES
```

7. If you want to log in press F4, YES. If you do not want to log in press F5, NO.

Note You must log in as the Install User to enter other users.

If you press F4, YES, you will see a screen similar to the following.

```
SETUP Passwords
USERNAME    PWD     LEVEL      TIME(min)
1  @BOB       *       INSTALL    15
2             *                   0
3             *                   0
4             *                   0
5             *                   0
6             *                   0
7             *                   0
8             *                   0
9             *                   0
Password has been set
```

Note If you are logged in, the @ will be displayed to indicate the current user name.

8. **To assign the next username, password, and level**,

   a. Move the cursor to the next available username, press ENTER, and use the function keys to type the username.

   b. Move the cursor to PWD, press ENTER, and use the function keys to type the password.

   c. Move the cursor to LEVEL, press F4, [CHOICE], and select a level.

   d. Move the cursor TIME and type a Default User Timeout value. You can adjust the Default User Timeout value from 0 to 10080 minutes (seven days).

   Note If the Default User Timeout value is 0 when you log in, a timeout will not occur. A value greater than 0 indicates the amount of inactivity time, in minutes, before the system times out.

   e. Repeat Step 8a through Step 8d for each user you want to have access to the system.

9. **To clear the current username and password**, press NEXT, >, and then press F2, CLEAR.

10. **To clear all usernames and passwords** for all users except the Install user, press NEXT, >, and then press F3, CLR_ALL.

11. **To modify the number of usernames in the system**,

> **Caution**
>
> If you modify the number of usernames to be fewer than the number of users currently assigned, some users will be deleted from the system.

   a. Press PREV to display the first SETUP Passwords screen.

   b. Move the cursor to Number of users and press ENTER. You can set the number of users to a minimum of 10 and a maximum of 100.

If you are increasing the number of users, you will see the following prompt.

```
Enter number of users for passwords:
```

c. Type the new number of users and press ENTER. You will see a screen similar to the following.

```
Changing number of users.
```

If you want to decrease the number of users, you will see the following prompt.

```
Reconfiguring.  DELETE users?[NO]
YES
```

To delete the users press F4, YES. To cancel the operation press F5, NO.

d. Turn off the controller then turn it on again to accept the new list of users.

12. **To log out** press F3, LOGOUT.

Note After the Default User Timeout expires, or you log out, the system reverts to the Operator level for that device. If the controller is turned off, all devices revert back to the Operator level.

3.9.3 Disabling Passwords

You can disable passwords in a system in which passwords have been set up. Disabling passwords causes the INSTALL user to be cleared. This will not clear other users. The INSTALL level is required to disable passwords. Use Procedure 3-35 to disable passwords.

Procedure 3-35 Disabling Passwords

Conditions

- You are logged in as an INSTALL user.

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].

4. Select Passwords. You will see a screen similar to the following.

```
SETUP Passwords
Current user:                  JOE
Current level:                 INSTALL
Default user timeout:          15 min
Timeout occurs in:             15 min
Log events:                    DISABLE
Number of users:               10
```

5. Press >, NEXT.
6. Press F3, DISABLE. You will see a screen similar to the following.

```
SETUP Passwords
Current user:                  JOE
Current level:                 INSTALL
Default user timeout:          15 min
Timeout occurs in:             15 min
Log events:                    DISABLE
Number of users:               10
Disable passwords?[NO]
```

Note Disabling passwords causes the INSTALL user to be cleared. This will not clear other users. The INSTALL level is required to disable passwords.

7. **To disable passwords**, press F4, YES. **If you do not want to disable passwords** , press F5, NO.

3.9.4 Program and Set Up User Password Operations

Program and Setup users can:

- Log in (Procedure 3-36)
- Log out (Procedure 3-37)
- Change their password (Procedure 3-38)
- Display the Password Log (Procedure 3-41 in Section 3.9.6)

Table 3–31. SETUP Passwords Screen Items — Screen

ITEM	DESCRIPTION
Current user	This item denotes who is currently logged in.
Current level	This item denotes the current system protection.
Default user timeout	This item shows how long the user will stay logged in before automatically returning to the operator level if no activity is performed.
Timeout occurs in	This item is the current amount of time left before the system returns to operator level.
Log events	This item tells whether events will be logged.
Number of users	This item is the total number of users allowed in the system.

Table 3–32. SETUP Passwords Screen Items — Screen

ITEM	**DESCRIPTION**
USERNAME	This column displays the name associated with a password user.
PWD	This column indicates whether a password has been set for a user.
LEVEL	This column shows which level of password protection of a user.
TIME	This column shows the timeout value for a user. The timeout value is how long a user will stay logged in at a specified password level if no activity is performed.
INSTALL user	This item assigns the username and password and then *logs in*. After logging in, the Install user *assigns* usernames, levels, and passwords for each user.

Procedure 3-36 Logging In

Conditions

- Passwords have already been set up. (Refer to Section 3.9.2)
- No user is currently logged in. Only one user can be logged in at a time.

Note If you do not know your username and password, contact the Install User.

Steps

1. Display the Setup Password Screen:

   - **If you are using FULL menus,**
     1. Press MENUS.
     2. Select SETUP.
     3. Press F1, [TYPE].

4. Select Passwords.

- **If you are using QUICK menus,**
  1. Press MENUS.
  2. Select SETUP PASSWORDS.

2. Press F2, USERS.

3. Move the cursor to your username.

4. **To log in**, press F2, LOGIN. You will see a screen similar to the following.

```
1  Uppercase
2  Lower Case
3  Punctuation
4  Options                        --Insert--
SETUP Passwords
--Password for MARY
Enter password:   '          '
Old Value:
```

5. Type your password and press ENTER.

6. **If you want to change the timeout value**, move the cursor to the TIMEOUT value for the current user and type a new timeout value. You can adjust the Default User Timeout value from 0 to 10080 minutes (seven days).

Note If the Default User Timeout value is 0 when you log in to the teach pendant or CRT device. The timeout will not occur.

Note Only one user can be logged in at a time for a given device. If another user is currently logged in, you must choose whether or not to log them out before you can log in. You will see the following prompt only on the teach pendant or the CRT window.

```
User JACK logged in.  Force logout?[NO]
YES        NO
```

7. **To log out the current user** press F4, YES. Otherwise press F5, NO.

If you select F4, YES, you will see a screen similar to the following.

```
SETUP Passwords
USERNAME    PWD     LEVEL      TIME(min)
1  JACK       *       INSTALL     0
2  @MARY      -       SETUP      15
3             *                   0
4             *                   0
5             *                   0
6             *                   0
7             *                   0
8             *                   0
9             *                   0
```

Note If you are logged in, the @ will be displayed to indicate the current user name.

Procedure 3-37 Logging Out

Conditions

- Passwords have already been set up. (Refer to Section 3.9.2)
- You are currently logged in. (Refer to Procedure 3-36)

Steps

1. Display the Setup Password Screen:
   - **If you are using FULL menus,**
     1. Press MENUS.
     2. Select SETUP.
     3. Press F1, [TYPE].
     4. Select Passwords.
   - **If you are using QUICK menus**,
     1. Press MENUS.
     2. Select SETUP PASSWORDS.
2. **To log out**, press F3, LOGOUT. After you log out the system reverts to the Operator level.

Procedure 3-38 Changing Your Password

Conditions

- Passwords have already been set up. (Refer to Section 3.9.2)
- You are currently logged in. (Refer to Procedure 3-36)

Steps

1. Display the Setup Password Screen:

   - **If you are using FULL menus**,
     1. Press MENUS.
     2. Select SETUP.
     3. Press F1, [TYPE].
     4. Select Passwords.
   - **If you are using QUICK menus**,
     1. Press MENUS.
     2. Select SETUP PASSWORDS. You will see a screen similar to the following.

```
SETUP Passwords
Current user:                  AAAA
Current level:                 INSTALL
1  Default user timeout:          15 min
2  Timeout occurs in:             4 min
3  Log events:                    DISABLE
4  Number of users:               10
```

2. Press F4, PASSWRD. You will see a screen similar to the following.

```
1  Uppercase
2  Lower Case
3  Punctuation
4  Options                      --Insert--
SETUP Passwords
--Set password for AAAA
Old password:   '          '
New password:   '          '
Verification:   '          '
Old Value:
```

3. Type the old password and press ENTER.
4. Type the new password and press ENTER.
5. Type the new password again to verify the first one is correct, and press ENTER.
6. Press F3, Logout. Immediately follow Procedure 3-36 , Logging In to set your new password.

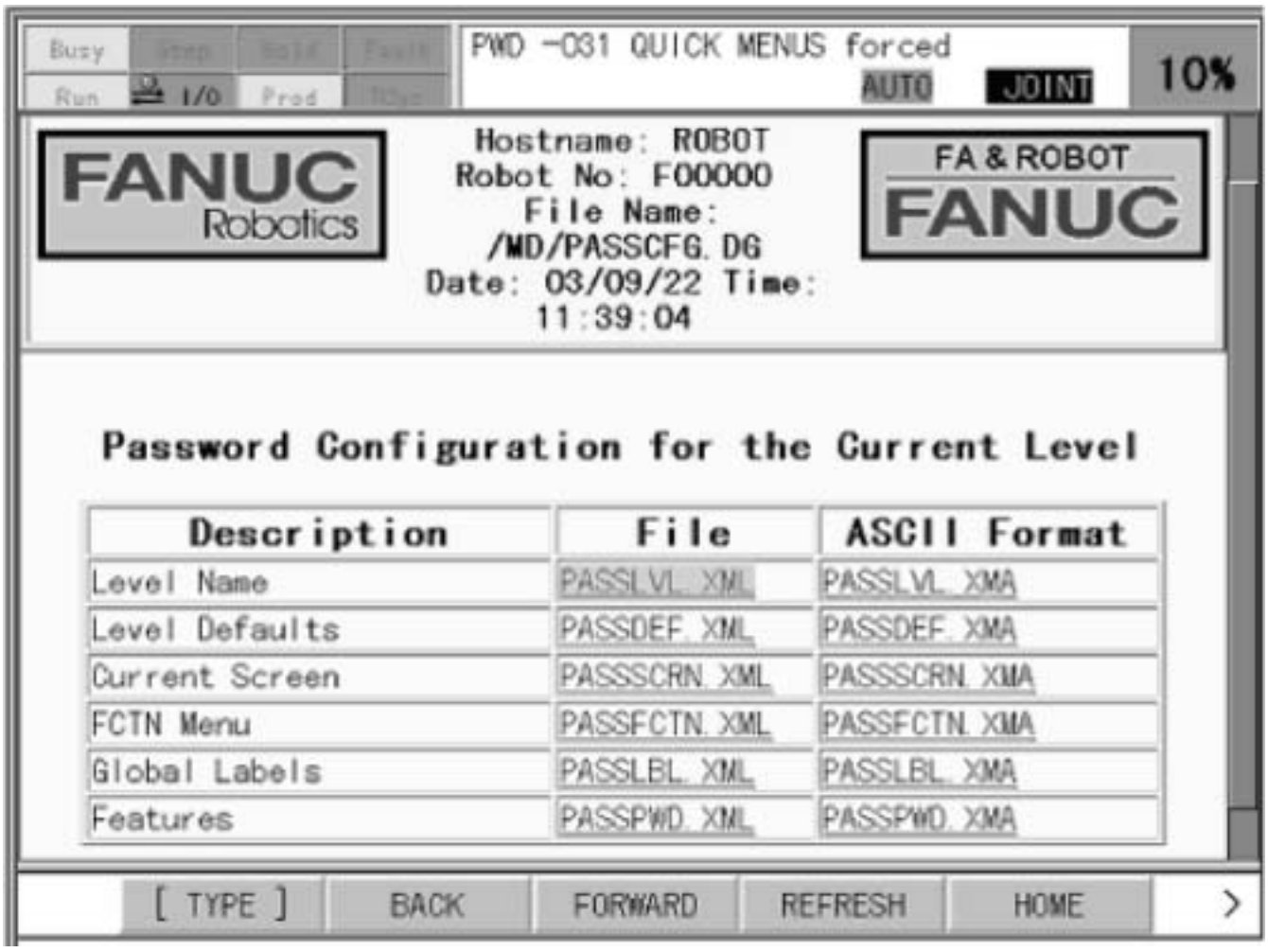
3.9.5 Configuring Passwords

3.9.5.1 Overview

Configuring passwords allows you to use an XML file to set up how the passwords function works.

To create a Password Configuration XML file, you might want to start with some examples. If you have the *i*Pendant or Internet Explorer connected to your robot, you can view the robot's HOME page. Select the link "Error/Diagnostic files (text) available on MD:" and select the link "PASSCFG.DG" Password Configuration. You will see a screen similar to the following

Figure 3–26. Password Configuration Screen



Each of these links contains a sample XML file. If you cannot view the robot's HOME page or you want to have your own copy, you can copy the files from the MD: device. For example, copy

MD:PASSLVL.XML to MC: to get your own copy of the file. The samples will show the Password Configuration for the current level of the user who is logged in.

The password configuration is stored as FRS:PASSWORD.DT. This file is read and configures passwords every time the system is turned on. The file FRS:PASSWORD.DT can be backed up and restored as an Application file by using the FILE menu.

Refer to Section 9.3 for more information.

Table 3–33 describes each item you can use to configure passwords. Use Procedure 3-39 to configure passwords.

Table 3–33. Password Configuration Setup Items

ITEM	DESCRIPTION
VERIFY	This item will parse the selected XML file and write any errors to a log file. It will not actually create the configuration data. It allows you to select an XML file from the default device. Use the FILE Menu to set the default device.
IMPORT	This item allows you to select an XML file from the default device. Use the FILE Menu to set the default device. IMPORT will parse the selected XML file and create the configuration data.
EXPORT	This item will copy the current configuration data to the default device as password.xml. **Note** You must first import a file in order for this to work.

Conditions

- You are logged in as an INSTALL user.

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Passwords. You will see a screen similar to the following.

```
SETUP Passwords
Current user:                   JOE
Current level:                  INSTALL
Default user timeout:           15 min
Timeout occurs in:              15 min
Log events:                     DISABLE
Number of users:                10
```

5. Press >, NEXT, and then press F6, CONFIG. You will see a screen similar to the following.

```
Configure Passwords
Password configuration is performed
by importing an XML file from the
default device.
The XML file can be verified first.
The current configuration can be
exported to password.xml.
```

3.9.5.2 XML Syntax for Password Configuration File

Table 3–34. XML Command Syntax

Name	Syntax and Description
Starting command	**Syntax**
	`<?xml version=”1.0”?>`
	Description
	This tag should be at the start of your XML file.

Table 3–34. XML Command Syntax (Cont'd)

Name	Syntax and Description
Comment	**Syntax**
	`<!--Password configuration file -->`
	Description
	This is a comment. All comments are included within <! -- -->
First tag	**Syntax**
	`<PASSWORD>`
	Description
	This must be the first tag. Only one of these can exist in your file.
Last tag	**Syntax**
	`</PASSWORD>`
	Description
	This must be the last tag. Only one of these can exist in your file.

Table 3–34. XML Command Syntax (Cont'd)

Name	Syntax and Description
Level name	**Syntax**
	<!– Level name –> LEVEL=”3” name=Maintenance” lange=”eg”/>
	Description
	This tag defines the level name up to 12 characters. All levels can be renamed including the“INSTALL” level. The Default level names are as follows: • Level 0— Operator • Level 1— Program • Level 2— Setup • level 3— Level 3 • Level 4— Level 4 • Level 5— Level 5 • Level 6— Level 6 • Level 7— Level 7 • Level 8— Install The lang parameter is optional. It should only be used in a multi-language system. It defines the language. • lang =eg— English • lang =jp— Japanese • lang =kn— Kanji • lang =fr— French • lang =gr— German • lang =sp— Spanish • lang =ch— Chinese • lang =tw— Taiwanese Each level tag defines one level name in one language. Therefore, multiple tags are needed to define multiple levels or multiple languages.

Table 3–34. XML Command Syntax (Cont'd)

Name	Syntax and Description
DefaultMenu Type	**Syntax**
	`<!-- Default menutype --> <DEFMENU level="3" type="2"/>`
	Description
	This tag defines the default menu type for a given level. The default menu type for all levels is Full menus. All levels except INSTALL can be changed to have a different default menu type. When a level changes such as when a user logs in, logs out, or times out, then the menu type will automatically change to the default menu type. The level parameter defines the level. The type parameter defines the menu type. • type = 0— Full menus • type = 1— Quick menus • type = 2— HMI Full menus • type = 3— HMI Quick menus **Note** HMI is only available on the *i*Pendant. The monochrome pendant will ignore the HMI setting.

Table 3–34. XML Command Syntax (Cont'd)

Name	Syntax and Description
Default Screen	**Syntax**
	<!– Default screen –> < !– Passwords –> < DEFSCREEN level="0" sp_id="935" scrn_id="1"/>
	Description
	This tag defines the default screen for a given level. The default screen for all levels is not defined. All levels except INSTALL can be changed to have a default screen. When a level changes such as when a user logs in, logs out, or times out, then the screen will automatically change to the default screen. If the default screen is not defined, then the original screen will continue to display if access is allowed; otherwise the first menu will be displayed. The level parameter defines the level. A unique Softpart ID and Screen ID define each screen. A screen may also be referred to as a menu. • sp_id="0", scrn_id="0" — Not defined • sp_id="935", scrn_id="1" — SETUP Passwords • sp_id="18", scrn_id="1" — ALARM • sp_id="71", scrn_id="1" — SELECT • sp_id="64", scrn_id="1" — EDIT For a complete list of Softpart Ids and Screen Ids: • On the *i*Pendant, press DISPLAY Menu, 5 Help/Diagnostics, 5 Menu Help • From Internet Explorer connected to your robot, browse to /md/tpmenus.dg • For a single menu, type md:passscrn.xml while the teach pendant is displaying that menu.

Table 3–34. XML Command Syntax (Cont'd)

Name	Syntax and Description
Screen Access	**Syntax**
	<!-- Screen access --> <!-- SELECT --> SCREEN level="3" sp_id="71" scrn_id="1" access="1" rw_access="1"/>
	Description
	This tag defines the screen access for a given level. All levels except INSTALL can be changed to allow/disallow screen access or to allow/disallow write access. The exception is SETUP Passwords menu. A screen tag for this menu is ignored. The default screen access is pre-defined by the system. Typically OPERATOR and LEVEL3-LEVEL7 have read-only access to all menus. PROGRAM level has read-write access to menus that program the robot. SETUP level has read-write access to menus that program and setup the robot. The level parameter defines the level. The sp_id and scrn_id parameters define the screen. • access=0— Screen cannot be entered • access=1— Screen can be entered • rw_access=0— If screen can be entered, it will only operate in read-only mode • rw_access=1— If screen can be entered, it can operate in read-write mode. If a screen has no access, the entry will still be visible in the MENUS or [TYPE] popup, but on the *i*Pendant it will not be selectable. On the monochrome teach pendant, it will be unnumbered and not selectable.

Table 3–34. XML Command Syntax (Cont'd)

Name	Syntax and Description
Local Labels	**Syntax**
	<!--Local labels for Host Comm --> < LLABEL level="3" name="[ACTION]" lang="eg" sp_id="195" scrn_id="1" access="1"/> < LLABEL level="3" name="DEFINE" lang="eg" sp_id="195" scrn_id="1" access="0”/> < LLABEL level="3" name="UNDEFINE" lang="eg" sp_id="195" scrn_id="1" access="0"/>
	Description
	This tag defines the function key access locally within a screen for a given level. All levels except INSTALL can be changed to allow/disallow function key access. The level parameter defines the level. The sp_id and scrn_id parameters define the screen. The name parameter defines the function key label in the screen or the pullup entry. The lang parameter is optional. It should only be used in a multi-language system. It defines the language. • access=0— Function key or pullup entry cannot be pressed. • access=1— Function key or pullup entry can be pressed. In this example, the Host Comm menu will allow [ACTION] function key to be pressed. ACTION is a pullup menu that contains DEFINE, UNDEFINE, START, and STOP. By allowing ACTION, the system automatically allows all entries in the pullup menu to be selected. However, DEFINE and UNDEFINE are disallowed because the local label defines no access for them.

Table 3–34. XML Command Syntax (Cont'd)

Name	Syntax and Description
FCTN Access	**Syntax**

```
<!-- FCTN access -- QUICK/FULL MENUS -->
<
SCREEN level="3" sp_id="64" scrn_id="20" access="0”/>
```

Description

The screen tag can also be used to define the access for each FCTN entry for a given level. All levels except INSTALL can be changed to allow/disallow FCTN entry access. The default is to allow access to all FCTN entries. The level parameter defines the level. The sp_id and scrn_id parameters define the FCTN entry.

- sp_id=”20”, scrn_id=“1” — ABORT (ALL)
- sp_id=”20”, scrn_id=”9” — Disable FWD/BWD
- sp_id=”931”, scrn_id=”1” — RELEASE WAIT
- sp_id=”64”, scrn_id=”20”— QUICK/FULL MENUS
- sp_id=”67”, scrn_id=”1” — SAVE
- sp_id=”171”, scrn_id=”1” — PRINT SCREEN
- sp_id=”20”, scrn_id=”2” — UNSIM ALL I/O
- sp_id=”381”, scrn_id=”21” — REFRESH PANE
- sp_id=”1304”, scrn_id=”2” — ENABLE/DISABLE HMI MENUS

Other FCTN entries might be available in your system. View MD:PASSFCTN.XML for a complete list. In the following example, the OPERATOR will be forced into HMI Quick Menus and cannot exit.

```
<!-- Level 0 will be stuck in HMI Quick Menus -->
<
!-- HMI Quick menutype -->
<
DEFMENU level=”0” type=”3”/>
<
!-- QUICK/FULL MENUS -->
<
SCREEN level=”0” sp_id=”64” scrn_id=”20” access=”0”/>
<
!-- ENABLE HMI MENUS -->
<
SCREEN level=”0” sp_id=’1304” scrn_id=”2” access=”0”/>
```

Table 3–34. XML Command Syntax (Cont'd)

Name	Syntax and Description
EDCMD Access	**Syntax**
	<!--EDCMD Insert --> <SCREEN level="1" sp_id="64" scrn_id="3" access="0"/>
	Description
	The screen tag can also be used to define the access for each EDCMD entry in the teach pendant Editor. All levels except INSTALL can be changed to allow/disallow EDCMD access. The default is to allow access to all EDCMD entries if the level has RW access to the teach pendant editor. The level parameter defines the level. The sp_id and scrn_id parameters define the EDCMD entry. • sp_id=”64”, scrn_id=“3” — Insert • sp_id=”64”, scrn_id=”4” — Delete • sp_id=”64”, scrn_id=”5” — Find • sp_id=”64”, scrn_id=”6”— Replace • sp_id=”64”, scrn_id=”7” — Renumber • sp_id=”64”, scrn_id=”9” — Copy • sp_id=”64”, scrn_id=”16” — Comment • sp_id=”64”, scrn_id=”17” — Undo • sp_id=”64”, scrn_id=”18” — Position number display • sp_id=”64”, scrn_id=”19” — Line number display • sp_id=”64”, scrn_id=”24” — Motion status display • sp_id=”31”, scrn_id=”127” — $VSHIFTMENU[1] • sp_id=”31”, scrn_id=”126” — $VSHIFTMENU[2] In the following example, the PROGRAM level cannot insert or delete lines. <!--EDCMD Insert --> <SCREEN level="1" sp_id="64" scrn_id="3" access="0"/> <!--EDMCD Delete --> <SCREEN level="1" sp_id="64" scrn_id="4" access="0"/>

Table 3–34. XML Command Syntax (Cont'd)

Name	Syntax and Description
Display Menu Access	**Syntax**
	<!--Display Menu Double --> <SCREEN level="0" sp_id="37" scrn_id="102" access="0"/>
	Description
	The screen tag can also be used to define the access for each Display Menu entry for a given level. All levels except INSTALL can be changed to allow/disallow Display Menu entry access. The default is to allow access to all Display Menu entries. The level parameter defines the level. The sp_id and scrn_id parameters define the Display Menu entry. • sp_id="37", scrn_id="101" — Single • sp_id="37", scrn_id="102" — Double • sp_id="37", scrn_id="103" — Triple sp_id="37", scrn_id="104"— Status/Single • sp_id="37", scrn_id="105" — Related Views • sp_id="37", scrn_id="107" — User Views • sp_id="37", scrn_id="108" — Menu Favorites • sp_id="37", scrn_id="109" — Menu History • sp_id="37", scrn_id="110" — Change Robot • sp_id="37", scrn_id="111" — Display Equip • sp_id="37", scrn_id="112" — EDIT<->EDIT In the following example, the OPERATOR level cannot change the screen configuration. <!--Display Menu Double --> <SCREEN level="0" sp_id="37" scrn_id="102" access="0"/> <!--Display Menu Triple --> <SCREEN level="0" sp_id="37" scrn_id="103" access="0"/> <!--Display Menu Status/Single --> <SCREEN level="0" sp_id="37" scrn_id="104" access="0"/>

Table 3–34. XML Command Syntax (Cont'd)

Name	Syntax and Description
Global Labels	**Syntax**
	<!-- Global labels --> GLABEL level="3" name="GROUP" lang="EG" access="1"/> GLABEL level="3" name="PU-1" lang="JP" access="1"/> GLABEL level="3" name="GROUP" lang="KN" access="1"/>
	Description
	This tag defines globally the function key access for a given level. All levels except INSTALL can be changed to allow/disallow access to a function key. If a screen is in read-only mode, then typically a function key cannot be pressed. Some function key labels are considered by the system as harmless and are by default globally allowed. The level parameter defines the level. The name parameter defines the function key label. Because it is global, it will take effect in any screen. However, a local label defined for a screen has precedence. The lang parameter is optional. It should only be used in a multi-language system. It defines the language. • access=0— Function key cannot be pressed • access=1— Function key can be pressed
	Syntax
	<!-- Features --> !-- PWD_KCL_C KCL --> PWD level="3" const="1" access="1"/>
	Description

Table 3–34. XML Command Syntax (Cont'd)

Name	Syntax and Description
	This tag defines the access for a feature for a given level. All levels except INSTALL can be changed to allow/disallow access to a feature. The level parameter defines the level. The constant parameter defines the feature. • const=1— KCL (default is SETUP) • const=2— I/O Configuration (default is SETUP) • const=3— FTP Device Access (default is SETUP and PROGRAM) • const=4— FTP Memory Access (default is SETUP) • const=5— HTTP Server Side Includes (default is SETUP and PROGRAM) • const=6— HTTP iPendant (default is INSTALL) • const=7— HTTP KAREL (default is INSTALL) • const=8— HTTP KCL (default is INSTALL) • const=9— HTTP Resource 4 (default is INSTALL) • const=10— HTTP Resource 5 (default is INSTALL) • const=11— HTTP Resource 6 (default is INSTALL) • const=12— HTTP Resource 7 (default is INSTALL)

Table 3–34. XML Command Syntax (Cont'd)

Name	Syntax and Description
Features cont'd	• const=13— HTTP Resource 8 (default is INSTALL) • const=14— User Defined 1 (default is PROGRAM) • const=15— User Defined 2 (default is SETUP) • const=16— User Defined 3 (default is PROGRAM and SETUP) • const=17— User Defined 4 (default is INSTALL) • const=18— Display Menu (default is OPERATOR) • const=19— SHIFT—COORD Menu (default is OPERATOR) • const=20— Jog Access (default is OPERATOR). Prohibits jogging w,p,r but x,y,z is allowed. JGFRM and Group 1 are forced. • const=21— Production Speed Access (default is OPERATOR). Prohibits lowering production speed below 80%. • access=0— Access to the feature is not allowed • access=1— Access to the feature is allowed

3.9.6 Password Log

If the Log Events item is set to ENABLE by the Install user on the SETUP Passwords screen, the following events will be displayed in the Password Log:

- Password events
- Programming events
- File manipulation events
- SpotTool application events

Each time an event occurs, such as when a user logs in or when a program is created, the event is logged in the Password Log.

The Password Log screen displays the items in Table 3–35 .

Table 3–35. Password Log Screen Items

ITEM	DESCRIPTION
Password Log Status	This item allows you to monitor password log ins and logouts. If the password log is enabled, you are able to tell who is logged in and what changes have been made.

Only the Install user can enable the Log Events item. Use Procedure 3-40 to enable the Password Log.

Any user can display the Password Log. Use Procedure 3-41 to display the Password Log.

Refer to the *Error Code Manual* for more information on PWD error codes.

Procedure 3-40 Enabling the Password Log

Conditions

- You are logged in as the Install User. (Procedure 3-34)

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Passwords. You will see a screen similar to the following.

```
SETUP Passwords
Current user:                AAAA
Current level:               INSTALL
1  Default user timeout:        15 min
2  Timeout occurs in:           4 min
3  Log events:                  DISABLE
4  Number of users:             10
```

5. **To disable or enable the Password Log**,
   a. Move the cursor to Log events.
   b. **To enable log events**, press F4, ENABLE.
   c. **To disable log events**, press F5, DISABLE.

Procedure 3-41 Displaying the Password Log

Conditions

- The Install User has set Log events to ENABLE. (Procedure 3-40)
- You are logged in at the Install, Program, or Setup level.
- FULL menus are displayed.

Steps

1. Press MENUS.
2. Select ALARM.
3. Press F1, [TYPE].
4. Select Password Log. You will see a screen similar to the following.

```
Password Log
1 PWD -035 Login (BOB) Install from CR
2 PWD -031 QUICK MENUS forced
3 PWD -039 Logout (BOB) Install from C
4 PWD -052 Pwd Timeout (MARY) from Teac=
5 PWD -034 Login (MARY) Install from Te
6 PWD -031 QUICK MENUS forced
7 PWD -038 Logout (MARY) Install from T
```

5. To get more information about a specific error, move the cursor to the error and press F5, HELP. You will see a screen similar to the following.

```
DETAIL Alarm
PWD -035 Login (BOB) Install from
CRT/Keyboard
Password Log
1 PWD -035 Login (BOB) Install from CR
2 PWD -031 QUICK MENUS forced
3 PWD -039 Logout (BOB) Install from C
4 PWD -052 Pwd Timeout (MARY) from Teac=
5 PWD -034 Login (MARY) Install from Te
```

Note Refer to the *Error Code Manual* for more information on PWD error messages.

3.9.7 Password Level Screen Permissions for HandlingTool

The following password screen permissions are available at different login levels:

- n/a = The screen is not displayed
- C = The information on the screen can be displayed, changed and operations can be performed
- D = The screen can only be displayed (you cannot change any information on the screen)

Table 3–36 lists each device type and the corresponding password level permissions for each device.

Table 3–36. Devices for Password Level Screen Permissions

Device	Operation Access
Teach Pendant	Allows all password levels access to screens based on Table 3–37 .
CRT/Keyboard	Allows all password levels access to screens based on Table 3–37 .
KCL (option only)	If the CRT/Keyboard option is not loaded, then only the install or setup password levels are granted access to perform KCL commands.

Table 3–37 lists each screen and the corresponding password level permissions for each level.

Table 3–37. Password Level Screen Permissions

Menu	Teach Pendant Screen	Password Level				
		Install	Setup	Program	Full Operator	Quick Operator
Utilities	Hints	D	D	D	D	D
	Program Adjust	C	C	D	D	n/a
	Auto TCP	C	C	C	C	n/a
	Program Toolbox	C	C	C	C	n/a
	Program Shift	C	C	D	D	n/a
	Mirror Image	C	C	D	D	n/a
	Tool Offset	C	C	C	C	n/a
	Frame Offset	C	C	C	C	n/a
Test Cycle	Test Run	C	C	C	C	C
	Debug Ctl	C	C	C	C	n/a
Manual Functions	Macro Manual Functions	C	C	C	C	C
	Error Recovery	C	C	C	C	n/a

Table 3–37. Password Level Screen Permissions (Cont'd)

Menu	Teach Pendant Screen	Password Level				
		Install	Setup	Program	Full Operator	Quick Operator
Alarm	Alarm	C	D	D	D	D
	Recovery	C	C	C	C	n/a
	Password Log	C	D	D	D	n/a
I/O	Digital	C	C	D	D	D
	Analog	C	C	D	D	n/a
	Group	C	C	D	D	D
	Robot	C	C	D	D	D
	UOP	C	C	D	D	n/a
	SOP	C	C	D	D	n/a
	Inter Connect	C	C	D	D	n/a
	I/O Link Device	C	C	D	D	n/a
	PLC I/O	C	C	D	D	n/a

Table 3–37. Password Level Screen Permissions (Cont'd)

Menu	Teach Pendant Screen	Password Level				
		Install	Setup	Program	Full Operator	Quick Operator
Setup	General Setup	C	C	D	D	n/a
	Frame Setup	C	C	D	D	D
	Port Init	C	C	D	D	n/a
	Macro	C	C	D	D	n/a
	Reference Position	C	C	D	D	n/a
	User Alarm	C	C	D	D	n/a
	Override Select	C	C	D	D	n/a
	RSR/PNS	C	D	D	D	n/a
	Passwords	C	C	C	C	C
	Host Communications	C	D	D	D	n/a
	Space Check	C	C	C	C	n/a
	Error Recovery	C	C	C	C	n/a
	Group Motion	C	C	C	C	n/a
	Coordinated Motion	C	D	D	D	n/a
	Line Track Setup	C	C	C	C	n/a
	Encoder Setup	C	C	C	C	n/a
	Continuous Turn	C	C	C	C	n/a
	Softfloat	C	C	C	C	n/a
	Shell Config	C	C	D	D	n/a
	Menu Utility	C	C	D	D	n/a
	Auto Normal	C	C	D	D	n/a
	Profibus	C	C	C	C	n/a
	Collision Guard	C	C	C	C	n/a
	Shapes	C	C	D	D	n/a

Table 3–37. Password Level Screen Permissions (Cont'd)

Menu	Teach Pendant Screen	Password Level				
		Install	Setup	Program	Full Operator	Quick Operator
File	File	C	C	C	D	n/a
	File Memory	C	C	C	D	n/a
	Controller Backup	C	C	C	C	n/a
Status	Axis Status	C	D	D	D	D
	Software Version	C	C	C	C	n/a
	Safety Signals	C	C	C	C	n/a
	Display Memory	C	C	C	C	n/a
	Program Timer	C	D	D	D	n/a
	System Timer	C	D	D	D	n/a
	Exec-History	C	D	D	D	n/a
	Conditions	C	D	D	D	n/a
	Prod Status	C	D	D	D	n/a
	User Status	C	D	D	D	n/a
	Program	C	D	D	D	n/a
	Profibus	C	C	C	C	n/a
User	User	C	C	C	C	n/a
User2	User2	C	C	C	C	n/a
Select	Select	C	C	C	C	D
Edit	Edit	C	C	C	C	D
Data	Register	C	C	C	D	D
	Position Register	C	C	C	D	D
	Palletizing Register	C	C	C	C	n/a
	KAREL Vars	C	C	C	C	n/a
	KAREL Posns	C	C	C	C	n/a
Position	Position	C	C	C	C	C

Table 3–37. Password Level Screen Permissions (Cont'd)

Menu	Teach Pendant Screen	Password Level				
		Install	Setup	Program	Full Operator	Quick Operator
System	System Variables	C	C	C	D	n/a
	Master/Calibrate	C	D	D	D	n/a
	Axis Limits	C	C	D	D	n/a
	Clock	C	C	D	D	n/a
	Overtravel Release	C	C	C	C	n/a
	System Config	C	D	D	D	n/a
	Motion Performance	C	D	D	D	n/a
Tools	Tool1	C	C	C	C	C
	Tool2	C	C	C	C	C
Vision	Vision	C	C	C	C	n/a
Other	KCL	C	C	D	D	D
	KAREL Editor	C	C	C	C	C

3.9.8 USB Password Function

By using USB password function, it is possible to automatically login a user by inserting USB device. This is only used to log in a user to the Teach Pendant device.

Procedure 3-42 Usage of USB Password Function

Conditions

- You are logged in as the Install User. (Procedure 3-34)

Steps

1. Select MENUS -> 0 NEXT -> 6 SYSTEM-> VARIABLES, Set $PASSWORD.$ENB_PCMPWD to TRUE
2. Select MENUS -> 6 SETUP
3. Press F1, [TYPE].
4. Select Passwords.

5. Press F2, LOGIN.

6. Move the cursor to the USER name that you want to use USB password function.

```
SETUP Passwords
USERNAME   PWD    LEVEL     TIME(min)
1  JACK       *       INSTALL    0
2  @MARY      -       SETUP     15
3             *                  0
4             *                  0
5             *                  0
6             *                  0
7             *                  0
8             *                  0
9             *                  0
[ TYPE ] LOGIN   LOGOUT          HELP
```

7. Press NEXT function key to show USB function key. If $PASSWORD.$ENB_PCMPWD is not TRUE or INSTALL level is not logged in, USB is not displayed.

```
[ TYPE ] CLEAR  CLR_ALL   USB   HELP
```

8. Press F4 USB in situation without inserting USB device to USB slot.

```
Insert USB device for password device.
                              OK      CANCEL
```

9. Press F4 OK after inserting USB for password device into USB slot. The next screen appears.

```
Write password to USB device, Proceed?
                              OK      CANCEL
```

10. Press F4 OK if you have no problem.

Caution

When USB device is made as password device, the contents of USB device is lost. Check the contents of USB device in advance.

11. The next screen appears. USB device in USB slot is now recognized as password device of USER name that the cursor highlighted in procedure 6.

```
Password device is created successfully.
```

12. If next conditions are met, it is possible to automatically login by inserting the USB device recognized as password device.

- $PASSWORD.$ENB_PCMPWD is TRUE.
- The USER name corresponding to the password device is setup in the controller.
- All users are logout.

13. When the USB device is removed from the USB slot, the user will be automatically logged out only if the timeout is 0, otherwise the user will timeout after the specified time. If the card is reinserted within the specified time, the user's time is extended until the USB device is removed again.

3.9.9 Using KCL with Passwords Enabled

Use Procedure 3-43 to use KCL with passwords enabled.

Procedure 3-43 Using KCL with Passwords Enabled

Conditions

- Use a standard terminator emulator (which supports DEC VT100/VT220 and handles escape sequences) to display a KCL window. Or, if the TELNET option is available, refer to the *Internet Options Setup and Operations Manual* to set up TELNET.
- The following screen is displayed.

```
KCL                              Timeout(min): 0
KCL USERNAME>
F10-MENUS
```

Steps

1. **Log in to KCL:**

   a. Use the arrow keys to type your user name, and press ENTER. You will see a screen similar to the following.

```
KCL                              Timeout(min): 0
KCL PASSWORD>
F10-MENUS
```

Note Every user has a specific password, so you must type in the password that is unique to a particular user. At the password prompt the key strokes will not echo back to the monitor.

Note You will need to type a valid user name that has at least SETUP Password level access.

   b. Use the arrow keys to type your user password, and press ENTER. You will see a screen similar to the following.

```
KCL                              Timeout(min): 0
KCL TIMEOUT(min)>45
F10-MENUS
```

Note Type in a valid timeout for the user between 0 and 10080. If no value is entered for the timeout, the default user timeout value will be used.

   c. Type the timeout value in minutes, and press ENTER.

2. To log out, you must type LOGOUT at the KCL prompt. The device will display the KCL USERNAME> prompt.

3.9.10 Password Auto Login Function

This feature is used to restrict access to the Teach Pendant device while the TP is disabled and/or in AUTO mode. It typically is not used with normal password operation, as the other devices are not restricted and when an auto login occurs, all operations are allowed.

For instance, when the Teach Pendant is disabled, the Teach Pendant operates in OPERATOR mode, but KCL is still allowed access. When the Teach Pendant is enabled, then INSTALL level is always granted.

If the user name for the install level of the password function is one of the user names in the table below, a login is automatically performed at the install level if the corresponding condition is met. If the corresponding condition is no longer met, a logout is automatically performed and the Teach Pendant is set to the operator level.

User name	Condition
_AUTOLOGIN_0	The teach pendant enable switch is ON, and the 3-mode switch is in T1 or T2.
_AUTOLOGIN_1	The teach pendant enable switch is ON.
_AUTOLOGIN_2	The 3-mode switch is in T1 or T2.

The time at which a login or logout is automatically performed is when the system changes from a status in which the corresponding condition is not met to a status in which it is met or when the system changes from a status in which the corresponding condition is met to a status in which is not met. Even if one of the user names above is set, it is possible to perform a login or logout on the password screen. In this case, an automatic login or logout may be performed later, when the corresponding condition is met or when it is not met. If auto logins are enabled, the alarm message displayed if operation restrictions are imposed due to a password is not the ordinary one, "PWD-069 Operation password protected", but one of the alarm messages below.

User name	TP enable switch	3–mode switch	Alarm Message
_AUTOLOGIN_0	ON	T1/T2	No restrictions.
_AUTOLOGIN_0	ON	AUTO	PWD-72 Protected in AUTO mode
_AUTOLOGIN_0	OFF	T1/T2	PWD-71 Protected in TP disable
_AUTOLOGIN_0	OFF	AUTO	PWD-72 Protected in AUTO mode
_AUTOLOGIN_1	ON	Not applicable	No restrictions.
_AUTOLOGIN_1	OFF	Not applicable	PWD-71 Protected in TP disable
_AUTOLOGIN_2	Not applicable	T1/T2	No restrictions.
_AUTOLOGIN_2	Not applicable	AUTO	PWD-72 Protected in AUTO mode

If the password log is enabled, auto logins and auto logouts are recorded in the log, as follows:

```
PWD-073  'Auto login '"(%s) %s"
PWD-074  'Auto logout '"(%s) %s"
```

3.10 ERROR SEVERITY TABLE

3.10.1 Error Severity Table Overview

The Error Severity Table screen allows you to modify severity levels for error codes in the direction of a higher severity level. In addition, you can output an error code to a PLC system with a severity level of PLCWARN, or use miscellaneous error logger functions.

Note Refer to Section 3.11 for more information about the Error Code output option.

3.10.2 Modifying Error Severity

When the system is about to display an error message, it first checks the severity level set in the system variable $ERROR_TABLE[ ] and compares the level with the default setting. If the severity level is higher than the default setting, the error code is then posted with the modified severity level setting. The system also checks the error logger functionality and logs the error accordingly.

When software is installed for the first time on the controller, the Error Severity Table screen looks similar to Figure 3–27 .

Figure 3–27. Error Severity Table

```
ERROR SEVERITY
Error Severity Table
#  FCode  FName  ECode    Sever    Erlog
1   ****   ****     0    DEFAULT   DEFAULT
2   ****   ****     0    DEFAULT   DEFAULT
3   ****   ****     0    DEFAULT   DEFAULT
4   ****   ****     0    DEFAULT   DEFAULT
5   ****   ****     0    DEFAULT   DEFAULT
6   ****   ****     0    DEFAULT   DEFAULT
7   ****   ****     0    DEFAULT   DEFAULT
8   ****   ****     0    DEFAULT   DEFAULT
9   ****   ****     0    DEFAULT   DEFAULT
```

Note The default number of entries is 20, but can be extended to 999.

Refer to Table 3–38 for a listing and description of the items on the severity table that you can modify or edit for each kind of error.

Table 3–38. Error Severity Table Items

Item	Description
FCode (Facility Code) and FName	This item is a decimal number which represents the subsystem in which an error has occurred. When you enter a number for the facility code, the system checks whether the number is a valid facility code. If not, an error message is displayed and you are prompted to enter a valid number. If the number is valid, the Facility Name will be displayed in the FName field of the table.
ECode (Error Code)	This item is a decimal number which is used by the system to report the proper error on the error message line of the teach pendant screen. Refer to the *Error Code Manual* for a complete listing of error codes and descriptions.

Table 3–38. Error Severity Table Items (Cont'd)

Item	Description
Sever (Severity Level)	This item indicates how serious the error is. This item can be modified, but only towards a higher severity level. For each possible error, you can set the following severity levels: • DEFAULT - This setting means that the error table has no effect on severity level for this error, and the default value of severity will be displayed. If the error output option is loaded, error codes with DEFAULT error severity levels will be output to the PLC system if the severity level is higher than WARNING. • STOP - This setting pauses program execution and stops robot motion. When a motion is stopped, the robot decelerates to a stop and any remaining part of the current motion segment is saved, meaning that motion can be resumed. STOP errors usually indicate that some action must be taken before the motion and program execution can be resumed. With these kinds of errors, the operator panel FAULT light and the teach pendant FAULT LED will be turned on. This severity level is for local tasks only. • STOPALL - This setting pauses program execution and stops robot motion. This severity level is for all tasks. • ABORT - This setting aborts program execution and stops robot motion. When an abort error occurs, the robot decelerates to a stop and the remainder of the motion segment is cancelled. This severity level is for local tasks only. • ABORTALL - This setting aborts program execution and stops robot motion. This severity level is for all tasks. • PLCWARN - If the error code output option is loaded, error codes with PLCWARN severity levels will be output to the PLC system with global severity bits set. In order to use this output option, you must have 33 digital outputs and one digital input which can be dedicated to this option.
Erlog (Error Logger)	This item provides different actions for each possible error. For each possible error, you can choose one of the following: • DEFAULT - This setting is the default setting. • ACTIVE - This setting displays an alarm in the active alarm log. • NODISP - This setting indicates that an error will not be displayed in either the error log or on the error line. • NOERLOG - This setting indicates that an error will not be displayed in the error log. • NOERLIN - This setting indicates that an error will not be displayed on the error line.

Use Procedure 3-44 to modify the error severity table. Use Procedure 3-45 to modify the number of entries in the error severity table.

Procedure 3-44 Modifying the Error Severity Table

Conditions

- You need to know the decimal numbers for the facility code and the error code so that you can modify the ERROR SEVERITY setup screen. (Section A.2.2)

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select 0, [NEXT], then select Error Table. You will see a screen similar to the following.

```
ERROR SEVERITY
Error Severity Table
#  FCode  FName  ECode    Sever    Erlog
1     11   SRVO     6    STOP     DEFAULT
2     11   SRVO     4    STOP     DEFAULT
3     10   FLPY     2    DEFAULT  ACTIVE
4     10   FLPY     3    DEFAULT  NODISPL
5      3   PROG     5    STOPALL  DEFAULT
6      3   PROG     6    STOPALL  DEFAULT
7      2   FILE     1    DEFAULT  DEFAULT
8      2   FILE     2    DEFAULT  DEFAULT
9     33   DICT     8    DEFAULT  DEFAULT
```

5. **If you want to add or modify an entry:**
   a. Move the cursor to the FCode field for the row you want to edit and type a decimal number. If the number is not valid, the error message "Invalid Facility Code entered" will be displayed. After you type a valid number, the FName field will automatically display the new facility code that corresponds to the new FCode.
   b. Move the cursor to the ECode field and type a decimal number within the range of valid numbers (0 - 999). If the number is not valid, the error message "Invalid integer (0 - 999)" will be displayed.
   c. Move the cursor to the Sever field and press F4, [CHOICE]. You will see a screen similar to the following.

```
1 DEFAULT    5 ABORTALL
2 STOP       6 PLCWARN
3 STOPALL    7
4 ABORT      8 NEXT
```

d. Move the cursor to the desired severity level and press ENTER.

Note The PLCWARN choice is only available if the system variable $ER_OUT_PUT.$plcwarn is set to TRUE. The default setting for this variable is FALSE. To set this variable to TRUE, refer to Section 3.11 .

6. **To modify Error Logger functionality** , move the cursor to the Erlog field and press F4, [CHOICE]. You will see a screen similar to the following.

```
1 DEFAULT    5 NOERLIN
2 ACTIVE     6
3 NODISP     7
4 NOERLOG    8 NEXT
```

a. Move the cursor to the desired error logger functionality and press ENTER.

b. **To verify that the proper error code was entered**, press F2, VERIFY.

c. **To display help information**, press F5, HELP. When you are finished, press PREV to return to the Error Severity Table screen.

d. **To clear information from a row**, move the cursor to the row whose information you want to clear, and press F3, [CLEAR]. The message "Clear this entry?" will be displayed.

Press F4, YES, to clear the current row.

Procedure 3-45 Modifying the Number of Entries in the Error Severity Table

Steps

1. Perform a Controlled start. Refer to Section C.1.5 for information on performing a Controlled start.
2. Press MENUS.
3. Press 0, NEXT, and select Program Setup. You will see a screen similar to the following.

```
Program Limits
Program Limits Setup
1  User Tasks                    1
2  Numeric Registers             200
3  Position Registers            100
4  Macros                        120
5  User Alarms                   10
6  Trace Length                  200
7  Num. Dig. Ports               512
8  Error Severity Table          20
```

4. Move the cursor to Error Severity Table, type the desired number of entries (1 - 999), and press ENTER.
5. Press FCTN.
6. Select Start (Cold).

3.11 ERROR CODE OUTPUT SETUP (OPTION)

3.11.1 Error Code Output Setup Overview

If you have the error code output option, you can send error codes to another device (such as a PLC), as numbers. You can use either of two methods to output error code information:

- Method 1: Output errors using 33 digital outputs
- Method 2: Output errors using 3 group outputs

In addition, you can output error code parameters.

Note If you want to use the Error Output option for warning errors listed in the Error Severity Table, note the following conditions:

- The Error Output option is installed.
- The numbers of the required signals are set in the appropriate system variables.
- The system variable $ER_OUT_PUT.$plcwarn is set to TRUE.

Refer to Section 3.10 for more information on the Error Severity Table.

3.11.2 Method 1: Output Errors Using 33 Digital Outputs

In order to use this option, you must have 33 digital outputs and one digital input that can be dedicated to this option. You set up the error code output option for this method by defining

- The number of the first digital output of the 33
- The number of the digital input to be used as the retrieval signal for the next alarm

Output Signals $ER_OUT_PUT.$out_num

You must define the starting number of the 33 output signals in the system variable $ER_OUT_PUT.$out_num. For example, if $ER_OUT_PUT.$out_num = 1, DO[1] through DO[33] are used. Refer to Table 3–39 for descriptions of the 33 output signals.

Table 3–39. Error Code Output Signal Definition ($ER_OUT_PUT.$out_num=1)

Signal Number(s)	Description
1 - 16	Define the error number
17 - 24	Define the subsystem reporting the error
25 - 32	Define the error severity
33	Used as the strobe signal

Error Code Number, 16 Output Signals

The first sixteen signals define the number of the error code, in binary notation.

Subsystem Reporting the Error, 8 Output Signals

The decimal value of this group of signals defines the alarm subsystem facility code. Refer to Appendix A for a listing of alarm subsystem facility names and codes.

Error Code Severity, 8 Output Signals

The next eight signals define the error code severity. Table 3–40 lists the severities associated with different values of these signals.

Table 3–40. Error Code Severity Definition ($ER_OUT_PUT.$out_num = 1)

Severity	Program Control		Motion Control		Servo Control	Local/ Global	Recovery	Display
	DO[25]	DO[26]	DO[27]	DO[28]	DO[29]	DO[30]	DO[31]	DO[32]
NONE*	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON
WARNING*	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
PAUSE.L**	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF
PAUSE.G***	OFF	ON	OFF	OFF	OFF	ON	OFF	OFF
STOP.L**	OFF	ON	ON	OFF	OFF	OFF	OFF	OFF
STOP.G***	OFF	ON	ON	OFF	OFF	ON	OFF	OFF
SERVO	OFF	ON	ON	OFF	ON	ON	OFF	OFF
SERVO2	ON	ON	OFF	ON	ON	ON	OFF	OFF
SYSTEM	ON	ON	OFF	ON	ON	ON	ON	OFF
ABORT.L	ON	ON	OFF	ON	OFF	OFF	OFF	OFF
ABORT.G	ON	ON	OFF	ON	OFF	ON	OFF	OFF

***** Errors with NONE or WARNING severity will not be output unless you also use the Error Severity Table. Refer to Section 3.10 .

****** Local severity; affects only the task from which the error is issued.

******* Global severity; affects all running tasks.

Program control is defined by DO[25] and DO[26], as shown in Table 3–41 .

Table 3–41. Program Control: DO[25] and DO[26]

DO[25]	DO[26]	Result
OFF	OFF	Program execution is not affected.
OFF	ON	The program is paused.
ON	ON	The program is aborted.

Motion control is defined by DO[27] and DO[28], as shown in Table 3–42 .

Table 3–42. Motion Control: DO[27] and DO[28]

DO[27]	DO[28]	Result
OFF	OFF	Motion execution is not affected.
OFF	ON	Motion is stopped.
ON	ON	Motion is stopped and canceled.

Servo control is defined by DO[29], as follows:

- OFF indicates that the servo power supply remains on.
- ON indicates that the servo power supply is off.

Local/global, whether the error will affect one task or all running tasks in a multi-tasking system, is defined by DO[30], as follows:

- OFF indicates that the error is effective only for one task.
- ON indicates that the error is effective for all tasks.

Recovery is defined by DO[31], as follows:

- OFF indicates that you do not need to cycle power to recover.
- ON indicates that you must cycle power to recover.

Display of the error message is defined by DO[32], as follows:

- OFF indicates that the error is displayed.
- ON indicates that the error is not displayed.

Example Output Definition

The example in Figure 3–28 shows the values of DO[1] through DO[32] ($ER_OUT_PUT.$OUT_NUM = 1) for SRVO-002:

- The alarm number is 2.
- The subsystem name is SRVO; this is value 11.
- The severity is SERVO.

Figure 3–28. Example Output Definition

Alarm Number = 2		Subsystem = 11 (SRVO)		Severity = SERVO	
DO[1]	OFF	DO[17]	ON	DO[25]	OFF
DO[2]	ON	DO[18]	ON	DO[26]	OFF
DO[3][	OFF	DO[19]	OFF	DO[27]	ON
DO[4]	OFF	DO[20]	ON	DO[28]	OFF
DO[5]	OFF	DO[21]	OFF	DO[29]	ON
DO[6]	OFF	DO[22]	OFF	DO[30]	ON
DO[7]	OFF	DO[23]	OFF	DO[31]	OFF
DO[8]	OFF	DO[24]	OFF	DO[32]	OFF
DO[9]	OFF				
DO[10]	OFF				
DO[11]	OFF				
DO[12]	OFF				
DO[13]	OFF				
DO[14]	OFF				
DO[15]	OFF				
DO[16]	OFF				

Note If you use the Error Output option for the errors listed in the Error Severity Table, the following conditions apply:

- Errors with NONE or WARNING severity levels can be output to the PLC system if they are listed in the Error Severity Table with severity set to PLCWARN.
- If errors with PLCWARN severity levels are output, the global severity bit (signal 30) will be set.
- If errors with PAUSE, PAUSEALL, ABORT, and ABORTALL severity levels are output, then the same digital output signals will be set as shown in Table 3–40 for PAUSE.L, PAUSE.G, ABORT.L, and ABORT.G severity levels respectively.

Input Signal $ER_OUT_PUT.$in_num

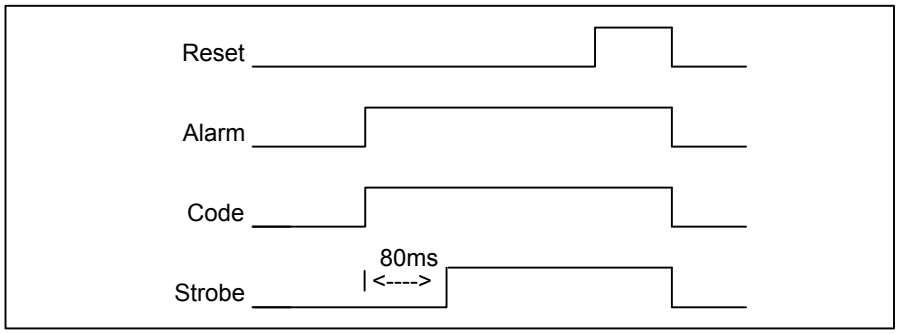
The input signal is used as the retrieval signal for the next alarm, when multiple alarms are output.

The number of the digital input signal is defined in the system variable $ER_OUT_PUT.$in_num. For example, when $ER_OUT_PUT.$in_num=1, DI[1] is the retrieval signal.

Timing

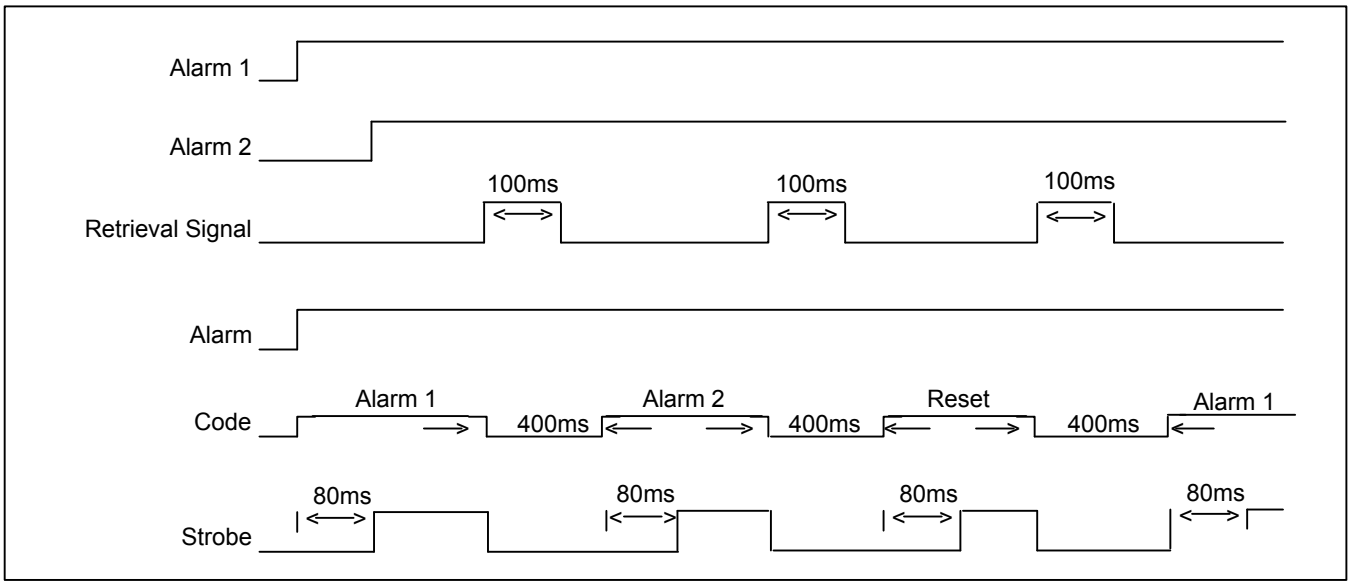
The signal timing for one alarm is shown in Figure 3–29 .

Figure 3–29. Timing - One Alarm



The signal timing for multiple alarms is shown in Figure 3–30 . When multiple alarms occur, the first alarm is output first. The signal is output one-at-a-time, in order of occurrence, whenever the retrieval signal is input. When all of the alarms have been output, the last alarm will be a reset alarm, which has a value of zero.

Figure 3–30. Timing - Multiple Alarms



3.11.3 Method 2: Output Errors Using 3 Group Outputs

In order to use this option, you must have 3 group outputs and one digital input that can be dedicated to this option. You set up the error code output option for this method by defining

- The numbers of the group output signals
- The number of the digital input to be used as the retrieval signal for the next alarm

Refer to the *Error Code Manual* for more information on error codes.

Output Signals

You must define one group output signal for each of the following components of an error message:

- Error code number
- Alarm facility code
- Error code severity

In addition, you define a digital output to be used for the strobe signal.

Note The maximum number of group outputs you can configure is 25.

Error Code Number $ER_OUT_OPUT.$error_num

The error code number comprises the first sixteen bits of the error code. It is recommended to configure the corresponding group output to use 10 digital output points. With a group output configured with 10 I/O points, the maximum value for the error number is 1024, which should be sufficient for current error code numbers.

You specify the number of the group output signal to be used to output the error code number in the system variable, $ER_OUT_PUT.$error_num.

Alarm Facility Code $ER_OUT_PUT.fac_num

The alarm facility code comprises the next seven bits of the error code. The decimal value of this group of signals defines the alarm subsystem facility code. It is recommended to configure the corresponding group output to use seven digital output points. Refer to Appendix A for a listing of alarm subsystem facility names and codes.

You specify the number of the group output signal to be used to output the alarm facility code in the system variable, $ER_OUT_PUT.$fac_num.

Error Code Severity $ER_OUT_PUT.$sev_num

The error code severity comprises the final eight bits of the error code. It is recommended to configure the corresponding group output to use eight digital output points. Refer to Table 3–40 and subsequent descriptions for information on how the values of eight bits correspond to the different error code severities.

You specify the number of the group output signal to be used to output the error code severity in the system variable, $ER_OUT_PUT.$sev_num.

Strobe Signal $ER_OUT_PUT.$grp_str

You specify the number of the digital output to be used as the strobe signal in the system variable, $ER_OUT_PUT.$grp_str. If this system variable is set to zero (which is its default value), this error output method will not be used.

Input Signal $ER_OUT_PUT.$in_num

The input signal is used as the retrieval signal for the next alarm, when multiple alarms are output.

The number of the digital input signal is defined in the system variable $ER_OUT_PUT.$in_num. For example, when $ER_OUT_PUT.$in_num=1, DI[1] is the retrieval signal.

3.11.4 Output Error Parameters

You can set up group outputs to convey the first two parameters of an error message, if the error message uses parameters. You do this by defining the numbers of the group outputs to hold the parameters.

You specify the number of the group output signals to be used to output the parameters as follows:

- Specify the number of the group output for the first error parameter in the system variable $ER_OUT_PUT.$parm1_num.
- Specify the number of the group output for the second error parameter in the system variable $ER_OUT_PUT.$parm2_num.

If the value of $ER_OUT_PUT.$parm1_num is zero, the system ignores the value of $ER_OUT_PUT.$parm2_num.

This feature works with both error code output methods.

3.11.5 Procedure

Use Procedure 3-46 to define the appropriate system variables for the error code output option.

Procedure 3-46 Setting Up the Error Code Output Option

Conditions

- The Error Code Output option is loaded.
- You have configured the signals that you are using for error code output.

Steps

1. Press MENUS.
2. Select SYSTEM.
3. Press F1, [TYPE].
4. Select Variables. You will see a screen similar to the following.

```
SYSTEM Variables
1 $ANGTOL                   [9] of REAL
2 $APPLICATION              [3] of STRING [21]
3 $AP_MAXAX                 0
4 $AP_PLUGGED               2
5 $AP_TOTALAX               16777216
6 $AP_USENUM                [32] of BYTE
7 $ASCII_SAVE               FALSE
8 AUTOINIT                  2
9 $BLT                      0
10 $CHECKCONFIG              FALSE
```

To move quickly through the information, press and hold down the SHIFT key and press the up or down arrow keys.

5. Move the cursor to**$ER_OUT_PUT** and press ENTER.
6. Move the cursor to item 1 and press ENTER.
7. Define the system variables that apply to the method of error code output you are using.
8. **Define $ER_OUT_PUT.$PLCWARN:**
   a. Move the cursor to $PLCWARN.
   b. Press F4, TRUE.
   c. Press PREV**two times**, or until the first system variable screen is displayed.

3.12 ROBOT PAYLOAD SETTING

3.12.1 Overview

Robot payload is the weight, or mass, of the robot end-of-arm tooling and workpiece. If you have not set up the proper robot payload during software installation, or if you need to change the robot payload because you have changed end-of-arm tooling or the workpiece, you must set robot payload.

Note It is important that the payload data used by the robot be as accurate as possible. The more accurate the values, the more effective features such as Collision Guard and Soft Float will be. Accurate values will also improve positional accuracy, cycle time, and general motion performance that are critical for today's processing.

You can define up to ten different payload schedules. You can then specify a payload schedule by using the payload setup screens and by using the payload teach pendant program instructions. You can set up payload schedules from the MOTION PERFORMANCE screen.

Refer to Section 6.23 for more information on the payload teach pendant program instructions.

3.12.2 Setting the Active Payload Schedule

You can set the active payload schedule:

- Manually (Procedure 3-47)
- In a teach pendant instruction (Procedure 3-48)

Refer to Table 3–43 through Table 3–46 for information on each Payload screen item.

You can set up the payload schedule parameters from the MOTION/PAYLOAD SET screen.

Procedure 3-47 Manually Setting the Active Payload Schedule

Note You cannot update payload values when a program is running or if the active schedule number is the same as the schedule you want to modify.

Conditions

- SRDY is on.
- No motion commands have been issued.
- $PARAM_GROUP[].$MOUNT_ANGLE has not been set.
- Robot mastering/calibration has been performed.

Steps

1. Press MENUS.
2. Select SYSTEM.
3. Press F1, [TYPE].
4. Select Motion. You will see a screen similar to the following.

```
MOTION PERFORMANCE
Group 1
No.   PAYLOAD[kg]       Comment
1         120.00        [                ]
2         120.00        [                ]
3         120.00        [                ]
4         120.00        [                ]
5         120.00        [                ]
6         120.00        [                ]
7         120.00        [                ]
8         120.00        [                ]
9         120.00        [                ]
10         120.00        [                ]
Active PAYLOAD number = 1
```

Table 3–43. MOTION PERFORMANCE Screen Items

ITEM	DESCRIPTION
No.	This item displays the payload schedule number.
PAYLOAD[kg]	This item displays the payload in kilograms.
Comment	This item displays any comments associated with each payload schedule.

5. To set the active payload, you must do the following:
   a. Press PREV until the payload schedule listing screen is displayed.
   b. Press F5, SETIND.

      Note If you see the message, "Running program already exists," you will not be able to change the index. You cannot change the index when a teach pendant program is running. Check whether any teach pendant program is running and abort it, if possible. Then, repeat the step.

   c. Type the number of the payload schedule you want and press ENTER.

Procedure 3-48 Using Teach Pendant Instructions to Set the Active Payload Schedule

Note Some applications and the Collision Guard function require the proper setting of payload information. If the payload changes during your application, you must use the PAYLOAD[x] instruction to select the appropriate payload schedule.

Note The PAYLOAD[x] instruction allows you to specify the *payload schedule* to use. You can specify up to 10 different sets of payload information. Each set of payload information corresponds to a schedule number.

Conditions

- You have set up the payload schedule that corresponds to the one you specify.
- For a multi-group application, you must use the PAYLOAD[GPx:y] instruction to select the appropriate payload schedule, y, for the specified motion group, x. Refer to Section 6.23 for details on the PAYLOAD[GPx:y] instruction.

Steps

1. Edit the teach pendant program in which you want to include PAYLOAD instructions.
   a. Press NEXT, until F2, [INST], is displayed then press F2, [INST].
   b. Select Payload and press ENTER.
   c. Select PAYLOAD[...] and press ENTER.
2. Type the value of the of the payload schedule:
   - Direct - type a schedule number and press ENTER.
   - Indirect - select INDIRECT, type a register number, and press ENTER.

3.12.3 Setup of Payload Schedules

There are two ways to set up the payload schedule parameters:

- **Manual entry** - Refer to Procedure 3-49 for more details.
- **Automatic estimation using Payload Ident. (Option J669)** - Refer to Procedure 3-50 for more details.

  Note Automatic robot payload estimation is available for some robot models. If your robot model does not have the payload setting feature, the message, "IDENT is not supported to this robot," will be displayed when you press F2, IDENT.

If you change any mechanical parts on the robot, such as a motor, you need to perform payload calibration before payload estimation.

If you do not want to perform payload estimation, but want to return the payload settings to the default values, you can reset them to the default values.

Use Procedure 3-49 to set up the Robot Payload Schedule manually.

Procedure 3-49 Manually Setting Robot Payload Schedule

Note You cannot update payload values when a program is running and the active schedule number is the same as the schedule you want to modify.

Conditions

- SRDY is on.
- No motion commands have been issued.
- $PARAM_GROUP[].$MOUNT_ANGLE has not been set.
- Robot mastering/calibration has been performed.
- The active payload schedule is set. Refer to Section 3.12.2 .

Steps

1. Press MENUS.
2. Select SYSTEM.
3. Press F1, [TYPE].
4. Select Motion. You will see a screen similar to the following.

```
MOTION PERFORMANCE
Group 1
No.   PAYLOAD[kg]       Comment
1         120.00        [                ]
2         120.00        [                ]
3         120.00        [                ]
4         120.00        [                ]
5         120.00        [                ]
6         120.00        [                ]
7         120.00        [                ]
8         120.00        [                ]
9         120.00        [                ]
10         120.00        [                ]
Active PAYLOAD number = 1
```

5. **To set up payload information manually** for the schedule you chose, move the cursor the payload schedule you want and press F3, DETAIL. You will see a screen similar to the following.

```
MOTION/PAYLOAD SET
Group 1
1 Schedule No[  1]:     [******************]
2 PAYLOAD               [kg]          120.00
3 PAYLOAD CENTER X      [cm]           30.00
4 PAYLOAD CENTER Y      [cm]           25.00
5 PAYLOAD CENTER Z      [cm]           25.00
6 PAYLOAD INERTIA X     [kgfcms^2]     77.00
7 PAYLOAD INERTIA Y     [kgfcms^2]    306.00
8 PAYLOAD INERTIA Z     [kgfcms^2]    306.00
```

a. **To display help** for the items on the screen, press F5, HELP. To display more information, use the arrow keys. When you are finished displaying help information, press PREV.

b. **To set the value**, move the cursor to the items you want to set and set them as desired. When you set payload, you set the values of several items related to payload. Refer to Table 3–44 for a short description of the items you must set.

Direction is relative to the robot tool frame with x, y, z, w, p, and r set to zero, and robot joint angles at the zero positions.

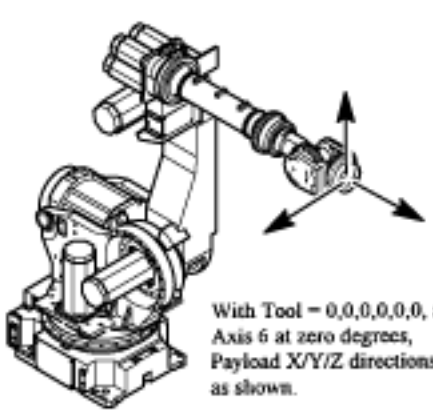
Refer to Figure 3–31 for additional equations used to calculate payload inertias.

Table 3–44. MOTION/PAYLOAD SET Screen Items

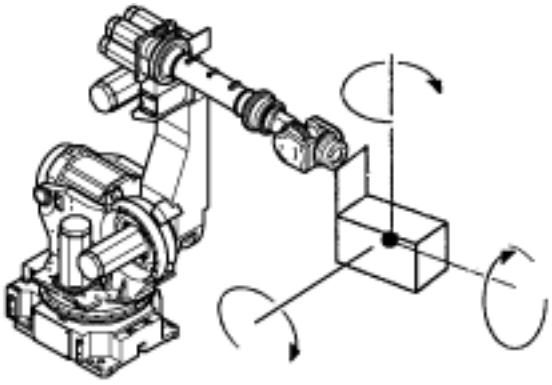
ITEM	DESCRIPTION
Schedule No [ ]	This item displays the payload schedule number.
Payload Units: kg	This item is the mass (weight) of the end-of-arm tooling. There are 2.21 pounds in a kilogram.
Payload Center X (cm)	This item is the up and down offset of the payload center of gravity from the center of the faceplate. Positive (+) values are up. There are 0.39 inches per centimeter.
Payload Center Y Units: cm	This item is the side offset of the payload center of gravity from the center of the faceplate. Positive (+) values are to the right of the faceplate when viewed from behind the faceplate. There are 0.39 inches per centimeter.
Payload Center Z Units: cm	This item is the offset of the payload center of gravity from the center of the faceplate. Positive (+) values are out from the faceplate. There are 0.39 inches per centimeter.
Payload Inertia X Units: kgfcms^2	This item is the moment of inertia of the payload around an axis parallel to the X-direction for the tool frame and through the center of gravity of the payload.
Payload Inertia Y Units: kgfcms^2	This item is the moment of inertia of the payload around an axis parallel to the Y-direction for the tool frame and through the center of gravity of the payload.
Payload Inertia Z Units: kgfcms^2	This item is the moment of inertia of the payload around an axis parallel to the Z-direction for the tool frame and through the center of gravity of the payload.

Table 3–44. MOTION/PAYLOAD SET Screen Items (Cont'd)

ITEM	DESCRIPTION
Arm load axis #1 Units: kg	This item is the additional mass mounted to axis 1.
Arm load axis #3 Units: kg	This item is the additional mass mounted to axis 3.



With Tool = 0,0,0,0,0,0, and Axis 6 at zero degrees, Payload X/Y/Z directions are as shown.



With Tool = 0,0,0,0,0,0, and Axis 6 at zero degrees
JX = Payload Inertia X
JY = Payload Inertia Y
JZ = Payload Inertia Z

In the example shown above, assume a = 60cm, b = 15cm, c = 10cm
(Material = steel, density = 0.00783 kg/cm^3)
Then M = Mass = 60 x 15 x 10 x 0.00783 = 70kg

From this, using standard Inertia formulae, the Payload Inertias can be calculated:

$$J_X = \frac{M}{12}\left(a^2 + c^2\right) = \frac{70}{12}\left(60^2 + 10^2\right) = 21583 kg.cm^2 = 22.0 kgf.cm.s^2$$

$$J_Y = \frac{M}{12}\left(a^2 + b^2\right) = \frac{70}{12}\left(60^2 + 15^2\right) = 22312 kg.cm^2 = 22.7 kgf.cm.s^2$$

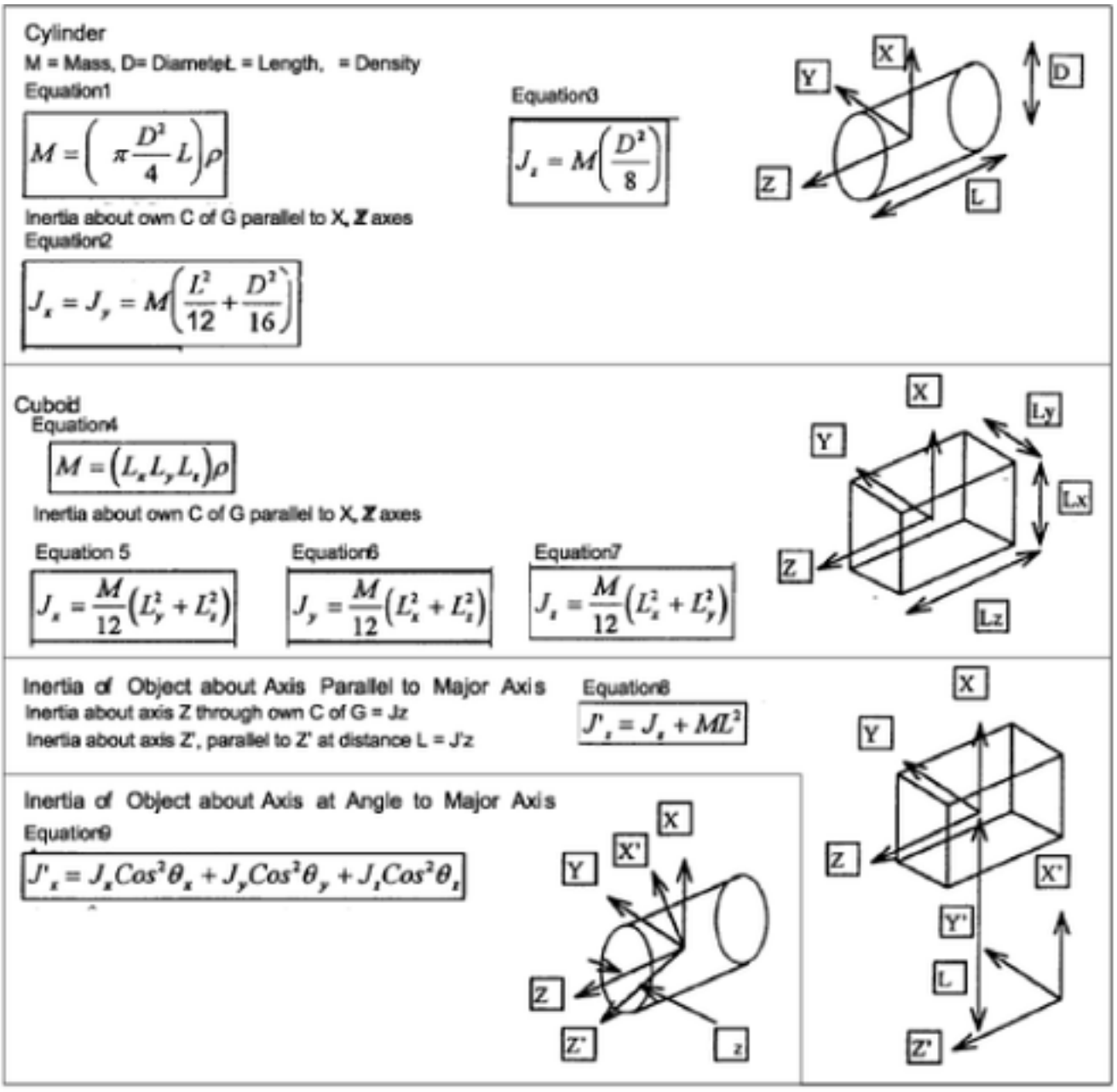
$$J_Z = \frac{M}{12}\left(b^2 + c^2\right) = \frac{70}{12}\left(15^2 + 10^2\right) = 1896 kg.cm^2 = 1.9 kgf.cm.s^2$$

Note 1: $(1 kgf.cm.s^2 = 980 kg.cm^2)$

Note 2: Equation $\left(J_x = \frac{M}{12}\left(width^2 + depth^2\right)\right)$ only applies to simple load shape shown. Other shape loads will require different inertia calculation formulae.

Refer to Figure 3–31 for inertia equations to use in calculating inertia.

Figure 3–31. Inertia Equations



6. **To select a different motion group**, press F2, GROUP, and specify the motion group you want.
7. **To select a different schedule number**, press F3, NUMBER, and enter the number of the payload schedule for which you want to set up payload information manually.
8. **To set payload values to the default values set at FANUC Robotics**, press and hold SHIFT and press F4, DEFAULT, and then perform one of the following:

- To confirm the change to the factory default values, press F4, YES.
- To cancel the default settings and return to the previous settings, press F4, NO.

3.12.4 Using Payload Ident. (Option 669) to Set Up Payload Schedules

One of the tools available to assist with accurate payload setting for most body assembly robots is Payload ID.

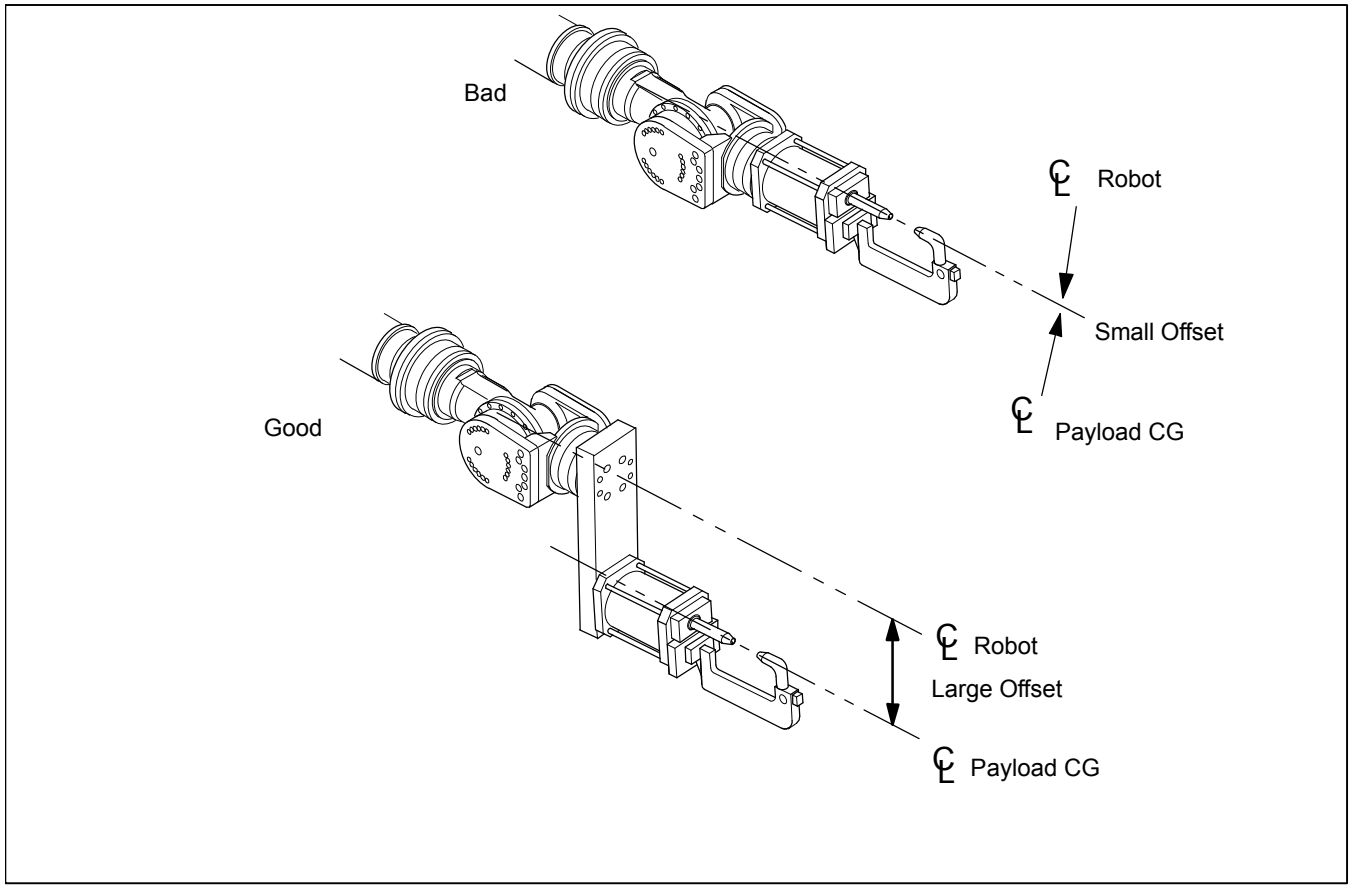
The Payload ID option allows the robot to calculate the properties of the payload, so that the features listed above can be optimized right out of the box. The Payload ID option consists of the robot moving the payload, axis 5 and 6 only, between two user taught positions, first at slow speed, and then at full speed. While Payload ID can be used to estimate the mass, COG and inertia of the unknown payload, it is preferred to use a known mass. The use of known mass eliminates one of the variables Payload ID must solve for, and increases the accuracy of the payload estimation.

Prior to running Payload ID, it is necessary to understand if the tool to be estimated is a good candidate for PayloadID.

A good estimation needs the following conditions:

- The range of J5 between POS1 and POS2 is 180 degrees.
- The range of J6 between POS1 and POS2 is 180 degrees.
- The x and/or y CG values of the payload to be estimated must be significant for the payload to be a viable candidate for estimation. If both the x and y values are near or at zero the estimation might be inaccurate. See Figure 3–32 for more detail.

Figure 3–32. Valid Payload Configuration for Accurate Estimation



If the Payload to be estimated appears to be a good candidate, perform Procedure 3-50 . If the payload is not a good candidate, use an alternative method for determining the properties of the tool.

Table 3–45 lists the items on the Motion/Payload ID screen.

Table 3–46 lists the items on the Motion/ID POS1 ID screen.

Table 3–45. MOTION/PAYLOAD ID Items

ITEM	DESCRIPTION
PAYLOAD ESTIMATION	This item allows you to estimate payload automatically.
MASS IS KNOWN	This item allows you to set whether or not the mass is known and the value, ranging from 0 to 999.99 kg.

Table 3–45. MOTION/PAYLOAD ID Items (Cont'd)

CALIBRATION MODE	This item allows you to calibrate the robot.
CALIBRATION STATUS	This item indicates whether the robot has been calibrated.

Table 3–46. MOTION/ID POS1 ID Items

ITEM	DESCRIPTION
POSITION FOR ESTIMATION	This item can be changed, depending on whether you want to use POSITION1 or POSITION2 for estimation.
J5 Default: -90.000	This item displays the J5 robot position.
J6 Default: -90.000	This item displays the J6 robot position.
SPEED Range: 1% - 100% Low default: 1% High default: 100%	This item displays the Low and High speeds.
ACCEL Range: 1% - 100% Low default: 1% High default: 100%	This item displays the Low and High acceleration rates.

Procedure 3-50 Automatic Payload Estimation using Payload Ident (Option J669)

Note You cannot update payload values when a program is running and the active schedule number is the same as the displayed schedule you want to modify.

Conditions

- SRDY is on.
- No motion commands have been issued.
- $PARAM_GROUP[].$MOUNT_ANGLE has not been set.
- Robot mastering/calibration has been performed.
  - — If software was reloaded, ensure that the master counts have been re-entered and the robot calibrated.
- You have weighed the tool prior to mounting, if possible. Having a known mass allows the robot to better estimate the COG and Inertia of the payload and will increase the overall accuracy of the Payload ID process. .

- There is no excess cable or dressout attached to Axis 5 or 6
  - — The LEONI bracket or any other dressout components at J6 must be removed for calibration.
  - — The dress on J3 / J4 can be left in place as these axes do not move during calibration.
- The robot is at the preferred Location/ Position
  - — The preferred robot position for running the calibration routine is the All Axis 0 position. This is to ensure an accurate calibration.
  - — At the minimum, Axis 3 must be such that the robot arm is parallel to the ground.
  - — Axis 4 must be at 0 degree. This will ensure that the J5 axis of rotation is parallel with the ground.
- The robot is in Automatic Mode. It is necessary to run the calibration routine in Automatic.

Steps

1. Press MENUS.
2. Select SYSTEM.
3. Press F1, [TYPE].
4. Select Motion. You will see a screen similar to the following.

```
MOTION PERFORMANCE
Group 1
No.   PAYLOAD[kg]        Comment
1          120.00         [                ]
2          120.00         [                ]
3          120.00         [                ]
4          120.00         [                ]
5          120.00         [                ]
6          120.00         [                ]
7          120.00         [                ]
8          120.00         [                ]
9          120.00         [                ]
10          120.00         [                ]
Active PAYLOAD number = 1
```

Note Automatic robot payload estimation is available for some robot models. If your robot model does not have the payload setting feature, the message, "IDENT is not supported to this robot," will be displayed when you press F2, IDENT.

5. **To perform automatic payload estimation:**
   a. Press PREV until the payload schedule listing screen is displayed.
   b. Press NEXT, >, and then press F2, IDENT. You will see a screen similar to the following.

```
MOTION/PAYLOAD ID
Group 1
Schedule No[  1]:     [****************]
1  PAYLOAD ESTIMATION            **********
Previous Estimated value     (Maximum)
Payload [Kg] :       0.00    (  165.00)****
Axis Moment  [Nm]
J4:              0.00E+00  (  9.02E+02)
J5:              0.00E+00  (  9.02E+02)
J6:              0.00E+00  (  4.41E+02)
Axis Inertia [Kgf cm^2]
J4:              0.00E+00  (  8.82E+05)
J5:              0.00E+00  (  8.82E+05)
J6:              0.00E+00  (  4.41E+05)
2 MASS IS KNOWN   [NO ]       165.000  [Kg]
3 CALIBRATION MODE                    [OFF]
4 CALIBRATION STATUS
```

c. Check the calibration status:

- If the status is DONE, continue to Step 7 .
- If the status is not DONE, you must perform payload calibration first. Refer to Step 6 .

6. If calibration status is not DONE, perform the following steps for Payload Calibration:

Note If the calibration values for the robot are known, they may be entered directly into the system variables. The values are entered at $PLCL_GRP [n] .$TRQ_MGN []. Set the proper values for each axis. When this is done, set $PLCL_GRP [n] .$CALIB_STAT=1.

Note If axis 5 or axis 6 motors are replaced, the robot must be re-calibrated. To ensure that this is done, set $PLCL_GRP[n].$CALIB_STAT=0, when one of these motors are replaced. This will change the calibration status from DONE, back to ****. After calibration is completed, $PLCL_GRP[n].$CALIB.STAT will be set to 1.



Caution

Do not use calibration data from a different robot. Otherwise, estimation might become inaccurate.



Caution

Make sure the end-of-arm tooling is not attached to the robot arm. Otherwise, payload calibration will not be correct.

a. Jog the robot to the position of the calibration. The suggested position for calibration is the zero position for all axes.

Note When the calibration is executed, the J5 and J6 axes move between POS1 and POS2, set on the ID POS1 and ID POS2 screens. Axes 1-4 will not move during calibration.

b. For calibration, set the values of POS1 and POS2 to the default.

a. Press NEXT, >, and then press F4, DETAIL. You will see a screen similar to the following.

```
MOTION/ID POS1
Group 1
1  POSITION for ESTIMATION          POSITION1
J1                               <**********>
J2                               <**********>
J3                               <**********>
J4                               <**********>
2  J5                               <   -90.000>
3  J6                               <   -90.000>
J7                               <**********>
J8                               <**********>
J9                               <**********>
4   SPEED           Low  <  1%>  High <100%>
5   ACCEL           Low  <100%>  High <100%>
```

b. Press F3, DEFAULT.

Note Speed and accel must be set to DEFAULT. The default values are shown on the above screen.

c. Press F2, POS2. You will see a screen similar to the following.

```
MOTION/ID POS2
Group 1
1  POSITION for ESTIMATION         POSITION2
J1                              <**********>
J2                              <**********>
J3                              <**********>
J4                              <**********>
2  J5                              <    90.000>
3  J6                              <    90.000>
J7                              <**********>
J8                              <**********>
J9                              <**********>
4   SPEED           Low  <  1%>  High <100%>
5   ACCEL           Low  <100%>  High <100%>
```

d. Press F3, DEFAULT.

a. Press PREV until the PAYLOAD ID screen is displayed.

```
MOTION/PAYLOAD ID
Group 1
Schedule No[  1]:     [****************]
1  PAYLOAD ESTIMATION            **********
Previous Estimated value    (Maximum)
Payload [Kg] :       0.00   (  165.00)****
Axis Moment  [Nm]
J4:              0.00E+00  (  9.02E+02)
J5:              0.00E+00  (  9.02E+02)
J6:              0.00E+00  (  4.41E+02)
Axis Inertia [Kgf cm^2]
J4:              0.00E+00  (  8.82E+05)
J5:              0.00E+00  (  8.82E+05)
J6:              0.00E+00  (  4.41E+05)
2 MASS IS KNOWN   [NO ]       165.000  [Kg]
3 CALIBRATION MODE                    [OFF]
4 CALIBRATION STATUS
```

b. Move the cursor to CALIBRATION MODE and press F4, ON.

Note After calibration has been performed, CALIBRATION MODE automatically turns OFF.

Caution

Do not change the CALIBRATION MODE during the calibration estimation process. Otherwise, the calibration will be incorrect or incomplete.

c. Turn the teach pendant ON/OFF switch to OFF and release the DEADMAN switch.

d. Move the cursor to Calibration Status.

e. Press F4, EXEC. You will see a screen similar to the following.

```
Robot moves and estimates.  Ready?
YES    NO
```

Warning

In the next step, the robot will move. Make sure all personnel and unnecessary equipment are out of the workcell and that all safeguards are in place; otherwise, you could injure personnel or damage equipment.

f. Decide whether to run the calibration:

- To run the calibration, press F4, YES.
- To cancel the calibration, press F5, NO.

When the robot has stopped moving, the payload calibration has been completed and you can continue to payload estimation.

g. Update the robot mastering sheet with the values stored in the following System Variables

```
PLCL_GRP[1].$TRQ_MGN[5]
PLCL_GRP[1].$TRQ_MGN[6]
```

7. Continue the automatic payload estimation:

a. Jog the robot to the position of the estimation.

Caution

If calibration status is not DONE, you must perform calibration (Step 6) before payload estimation. If you do not perform calibration first, the payload estimation will be incorrect.

Note When the estimation is executed, the J5 and J6 axes move between POS1 and POS2, set on the ID POS1 and ID POS2 screens, and the other axes keep the current positions. (Step 7g and Step 7i)

Note If the axis of J5 rotation is vertical (J4 is at ±90 degrees), the estimation cannot be done. Make sure that the axis of J5 rotation is as horizontal as possible.

b. Move the cursor to the Payload to be estimated

If necessary, press PREV until the payload schedule listing screen is displayed

```
MOTION/PAYLOAD ID
Group 1
Schedule No[  1]:     [****************]
1  PAYLOAD ESTIMATION            **********
Previous Estimated value    (Maximum)
Payload [Kg] :       0.00   (  165.00)****
Axis Moment  [Nm]
J4:              0.00E+00  (  9.02E+02)
J5:              0.00E+00  (  9.02E+02)
J6:              0.00E+00  (  4.41E+02)
Axis Inertia [Kgf cm^2]
J4:              0.00E+00  (  8.82E+05)
J5:              0.00E+00  (  8.82E+05)
J6:              0.00E+00  (  4.41E+05)
2 MASS IS KNOWN   [NO ]       165.000  [Kg]
3 CALIBRATION MODE                    [OFF]
4 CALIBRATION STATUS
```

c. Verify that the correct Payload is the active Payload. Press F3, NUMBER, and enter the number of the payload schedule for which you want to perform payload estimation.

Note Be sure that the active payload value used during estimation is close to the actual value. For example, if the actual payload is about 100 kg, then

```
Active Payload [Kg] = 0.00   <— NOT OK
Active Payload [Kg] = 100.00 <— OK
```

Using a value that is close to the actual mass of the payload ensures that the robot motion during estimation is as smooth as possible, providing a better payload estimation, and that the results are applied to the correct payload.

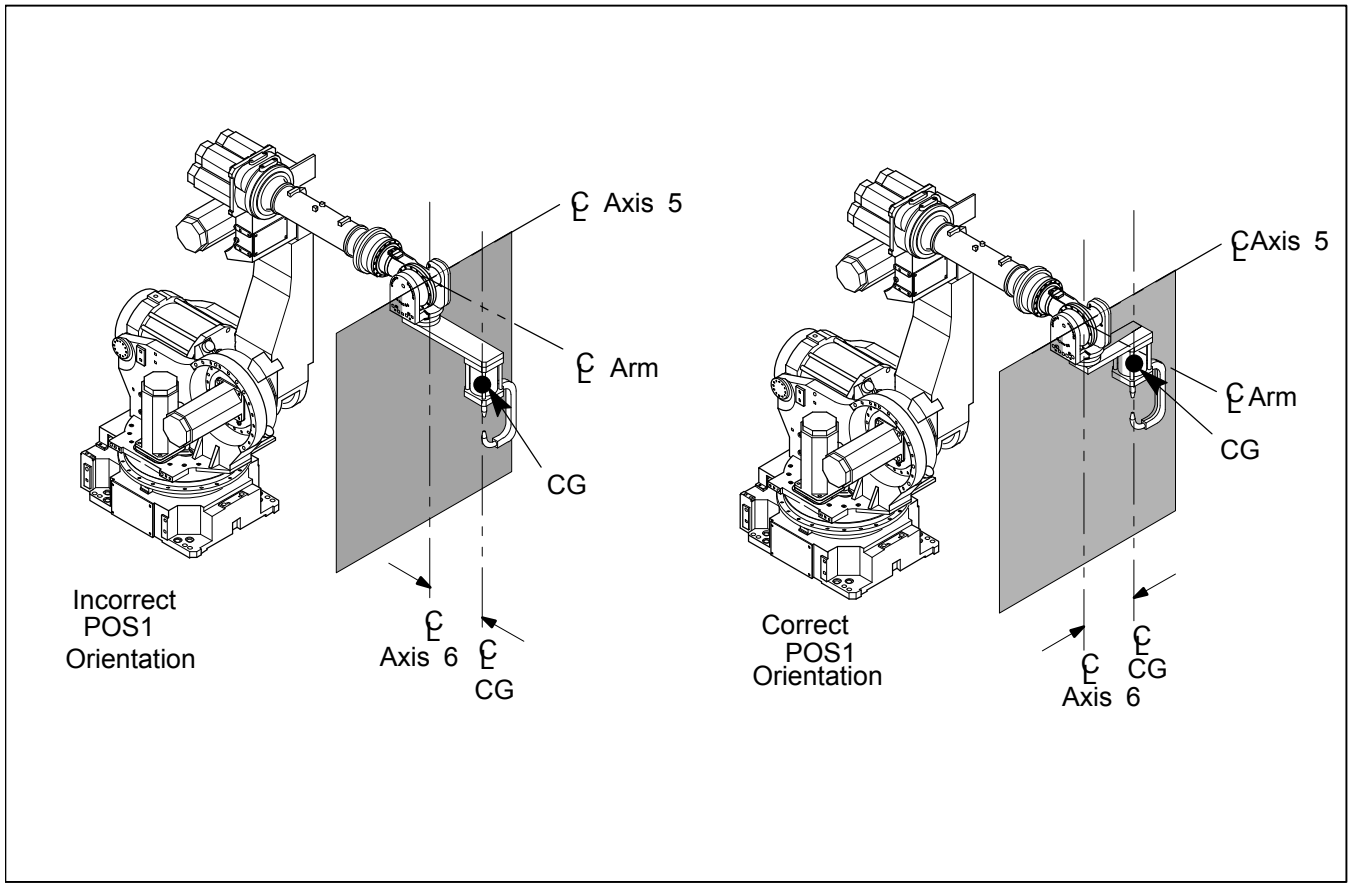
d. Move the cursor to MASS IS KNOWN.

Note You should know the actual mass of the tool, if possible. This will increase the overall accuracy of the Payload ID process. Having a known mass allows the robot to better estimate the COG and Inertia of the payload. If the actual mass of the tool is not known, use the value from design.

- **If there is no design information for the payload to be estimated but the actual mass is known**, select YES. Type the mass of the payload, using the actual mass of the tool, and press ENTER.
- **If there is design information for the payload to be estimated but the actual mass is not known**, select YES. Type the mass of the payload, using the design mass of the tool, and press ENTER.
- **If you do not know the actual or design mass of the tool**, leave the PayloadID setting for Mass Is Known set to NO.

e. POS1 and POS2 are recommended to be set such that the (user-estimated) center of gravity (CG) will be on the plane formed by the axis of rotation of J5 and the axis or rotation of J6. See Figure 3–33 .

Figure 3–33. J5 and J6 Orientation for POS1



f. Press NEXT, >, and then press F4, DETAIL. You will see a screen similar to the following.

```
MOTION/ID POS1
Group 1
1  POSITION for ESTIMATION          POSITION1
J1                               <**********>
J2                               <**********>
J3                               <**********>
J4                               <**********>
2  J5                               <   -90.000>
3  J6                               <   -90.000>
J7                               <**********>
J8                               <**********>
J9                               <**********>
4   SPEED           Low  <  1%>  High <100%>
5   ACCEL           Low  <100%>  High <100%>
```

g. Perform **one** of the following:

- Type the angle of each axis of POS1 (in degrees), **or**
- Jog axes J5 and J6 to the position you want to record as POS1 and press SHIFT and F5, RECORD, to record the J5 and J6 axis positions.

h. Press F2, POS.2. You will see a screen similar to the following.

```
MOTION/ID POS2
Group 1
1  POSITION for ESTIMATION         POSITION2
J1                              <**********>
J2                              <**********>
J3                              <**********>
J4                              <**********>
2  J5                              <    90.000>
3  J6                              <    90.000>
J7                              <**********>
J8                              <**********>
J9                              <**********>
4   SPEED           Low  <  1%>  High <100%>
5   ACCEL           Low  <100%>  High <100%>
```

i. Perform **one** of the following:

- Type the angle of each axis of POS2 (in degrees),**or**
- Jog axes J5 and J6 to the position you want to record as POS2 and press SHIFT and F5, RECORD, to record the J5 and J6 axis positions.

Note Guidelines for POS1 and POS2

The range of axis 5 and 6 between POS1 and POS2 must be as large as possible, up to the recommended 180 degrees, to ensure the most accurate payload estimation possible. Choose the axis 1 - 3 position to provide the largest range of motion for the axis 5 and 6, while making sure that axis 4 is at 0 degrees, and the robot arm is parallel to the floor.

j. The high and low accel values (item 5 on the previous Motion/ID POS1 screen) can be modified, under certain conditions, to optimize estimation. If the actual mass of the payload is less than the maximum for the robot model, then refer to Figure 3–34 for a guideline on how to calculate the new values.

Figure 3–34. Acceleration Equation

$$\frac{\text{Maximum Payload}}{\text{Actual Payload}} \times 100 = \text{Accel value (max 255)}$$

- If calculated values exceed the maximum value of 255, set the values to 255. The high and low accel values must be set to the same number.
- If excessive vibration occurs during estimation, reduce the values and repeat the estimation. Continue to reduce the values until the vibration diminishes.

Caution

Changes to the speed settings will greatly reduce the accuracy of the payload estimation.

- The low speed setting **must** be set to 1%. DO NOT CHANGE THE LOW SPEED SETTINGS.
- Leave the High speed setting at 100% for better estimation.

k. Press PREV until the PAYLOAD ID screen is displayed.

l. Turn the teach pendant ON/OFF switch to OFF and release the DEADMAN switch.

m. Press F4, EXEC. You will see a screen similar to the following.

```
Robot moves and estimates.  Ready?
YES    NO
```

Warning

In the next step, the robot will move. Make sure all personnel and unnecessary equipment are out of the workcell and that all safeguards are in place; otherwise, you could injure personnel or damage equipment.

n. Decide whether to run the payload estimation:

- To run the estimation, press F4, YES.
- To cancel the estimation, press F5, NO.

The robot will perform the estimation at the current robot position for axes 1 through 4. When the robot has stopped moving, the payload has been estimated and the screen will be updated to the new values.

Warning

Do not attempt to adjust the override while the robot performs the estimation. This will cause the results to be inaccurate.

o. Press F5, APPLY, to apply the estimated value to the payload schedule. You will see a screen similar to the following.

```
Path and Cycletime will change. Set it?
YES    NO
```

Warning

Make sure that the payload schedule you define matches the correct payload information before you continue; otherwise, the robot will not move the way you expect, and could injure personnel or damage equipment.

p. Decide whether to accept the estimated payload:

- To accept the payload, press F4, YES.
- To reject the payload, press F5, NO.

q. **If the load is over the specification**, you will see a screen similar to the following.

```
Load is OVER spec !  Accept ?
YES    NO
```

r. Decide whether to accept the estimated payload:

- To accept the payload, press F4, YES.

- To reject the payload, press F5, NO.

Note Once estimation is completed, the process can be repeated to improve accuracy. When repeating the process, ensure that the estimated payload is the current active payload and the results are APPLIED each cycle. This repeating allows the robot to produce a more accurate estimation because an accurate payload is used during estimation motion, thus fine-tuning the estimated value.

8. **To select a different motion group**, press F2, GROUP, and specify the motion group you want.

Note If you see the message, "Running program already exists," you will not be able to change the index. You cannot change the index when a teach pendant program is running.

Note If you have installed the Material Handling Shell option, the program MULTI_IO.TP runs every time you turn on the controller. You cannot abort this program. To avoid getting this message, go to MENUS, SETUP, Shell Config. Set the Using Multiio task item to NO from YES and then turn off the controller and turn it on. After you have set the index and want to run MULTI_IO.TP again, go back to the Shell Config screen, set NO to YES, and then cycle power again.

3.12.5 Setting Up Arm Load Information

Table 3–47 displays the items you can set up on the Motion/Armload screen.

Use Procedure 3-51 to set up arm load information.

Table 3–47. MOTION/ARMLOAD SET Items

ITEM	DESCRIPTION
ARM LOAD AXIS #1 [kg] Units: kg	This item is the additional mass mounted to axis 1.
ARM LOAD AXIS #3 [kg] Units: kg	This item is the additional mass mounted to axis 3.

Procedure 3-51 Arm Load Information

Steps

1. **To set arm load information**, press PREV until the payload schedule listing screen is displayed, and press F4, ARMLOAD. You will see a screen similar to the following.

```
MOTION PERFORMANCE
Group 1
1  ARM LOAD AXIS #1  [kg]   250.00
2  ARM LOAD AXIS #3  [kg]    20.00
Please power off/on after modification
```

a. **To display help** for the items on the screen, press F5, HELP. To display more information, use the arrow keys. When you are finished displaying help information, press PREV.

b. Move the cursor to the item you want to set and set it as desired.

Note Arm load includes any additional mass , such as tooling that is mounted on robot axes.

2. **When you are finished setting all arm load payload information**, CYCLE POWER.

a. Press FCTN.

b. Select CYCLE POWER, and press ENTER.

c. Select OK and press ENTER.

3.13 STROKE LIMIT SETUP

You use the Stroke Limit Setup screen to set three sets of stroke limits for the J1 axis and the E1 extended axis. You must have the extended axis option installed on your robot to set and use stroke limits for the E1 axis. The stroke limit function can then select these stroke limit settings during program execution. Refer to Table 3–48 .

Table 3–48. Stroke Limit Setup Screen Items

ITEM	DESCRIPTION
Upper limit	This item is the upper limit of the axis stroke. It is the stroke limit in the positive direction.
Lower limit	This item is the lower limit of the axis stroke. It is the stroke limit in the negative direction.

When you change the upper limit or lower limit, you must perform a cold start for the new settings to take effect. When the new upper and lower limits are enabled, the stroke limit becomes the default value. This default value is set in the system variable $PARAM_GROUP.$SLMT_**_NUM, where ** is the axis number (J1 or E1).

Warning

When you change the stroke limit, the stroke of the robot is changed. Consider the effect of your stroke setting before you change the stroke limit; otherwise, you could injure personnel or damage equipment.

Use Procedure 3-52 to set up stroke limits.

Procedure 3-52 Setting Up Stroke Limits

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Stroke Limit. You will see a screen similar to the following.

```
Stroke Limit Setup
GROUP :1          AXIS :J1
No. LOWER >- 150.0   UPPER < 150.0
1: ******** deg     ******** deg
2: ******** deg     ******** deg
3: ******** deg     ******** deg
Default
0: -150.0   deg     150.0    deg
Active limit:
$MRR_GRP[*].$SLMT_**_NUM = 0
```

5. **To select the motion group**, press F2, GROUP#, type the number of the group you want, and press ENTER.
6. **To set up an extended axis stroke limit**, press F3, AXIS#. The extended axis stroke limit setup screen will be displayed.
7. Move the cursor to the stroke limit value you want to set.



Warning

When you change the stroke limit, the stroke of the robot is changed. Consider the effect of your stroke setting before you change the stroke limit; otherwise, you could injure personnel or damage equipment.

Note The upper limit and lower limit must be in the stroke limit defined by the system. If you set a value out of this stroke limit, the value is changed to the stroke limit defined by the system.

8. Type the stroke limit value you want and press ENTER.
9. Perform a Cold start:

3.14 TEACHING KAREL VARIABLES (DispenseTool and HandlingTool)

3.14.1 Overview

KAREL variables consist of the following:

- Positions
- Paths
- Variables

Positions and **paths** used in a program must be *taught* , or defined, before you can run the program or run production.

Variables used in a program can be defined outside of a program, if they have not been defined within the program.

This section contains information about teaching KAREL positions, paths, and variables.

3.14.2 KAREL Positions

Before you can run a KAREL program or run production using a KAREL program, you must *teach* all KAREL positions within the program. You can use two methods to teach a KAREL position:

- **Jog** the robot to the position and **record** it.
- **Set** the value of each positional component.

You can set KAREL positions from the DATA KAREL Posns screen.

Table 3–49. DATA KAREL Posns Screen Items

ITEM	DESCRIPTION
Position name	This item is the name of the position.
Motion group	This item indicates that the position is in a particular motion group.
Position status	This item indicates whether the position has been taught.

This section contains a procedure for teaching KAREL positions. Refer to Chapter 2 *TURNING ON AND JOGGING THE ROBOT* for information on jogging the robot. Refer to Chapter 6 *PROGRAM ELEMENTS* for information on the components of a position. Use Procedure 3-53 to teach KAREL positions.

Procedure 3-53 Teaching KAREL Positions

Conditions

- All personnel and unnecessary equipment are out of the workcell.
- The KAREL program that contains the positions you are teaching has been loaded into controller memory. Refer to Chapter 9 *PROGRAM AND FILE MANIPULATION* .

Steps

1. Select the KAREL program for which you want to modify KAREL variables:
   a. Press SELECT.
   b. Press F1, [TYPE].
   c. Select KAREL Progs.
   d. Move the cursor to the KAREL program or variable file you want and press ENTER.
2. Press DATA.
3. Press F1, [TYPE].
4. Select KAREL Posns. See the following screen for an example of the *top level* KAREL Posns screen. It lists both position and path variables.

```
DATA KAREL Posns
1 POS1          G1  Not Recorded
2 POS2          G2  Not Recorded
3 WPR1          G1  Not Recorded
4 POS_ARR1      G1  [5] of Position
5 WPR_ARR2      G1  [4,5] of XYZWPR
6 VEC1              Vector
7 PTH1              2 nodes
8 PTH2              0 nodes
9 PTH3              0 nodes
10 PTH4             3 nodes
```

Note G1 indicates that the position is in motion group 1.

@, when displayed, indicates that the robot is currently near the position.

Note If KAREL positions are not displayed, make sure the $KAREL_ENB system variable is set to 1.

Note You can now teach a position by recording it or by setting position components. After you have taught a position you can test it by moving the robot to it.

5. To teach a position by recording it:

a. On the top level KAREL Posns screen, move the cursor to the position you want to teach.

b. Continuously press the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.

c. Jog the robot to the position you want to record.

d. Press and hold the SHIFT key and press F3, RECORD. The status of the position changes from Not Recorded to Recorded.

6. To teach a position by setting position components:

a. On the top level KAREL Posns screen, move the cursor to the position you want to teach.

b. Press ENTER. You will see a screen similar to the following.

```
DATA KAREL Posns
POS1 IN GROUP[1]
1  C                    Not Recorded
2  X                    Not Recorded
3  Y                    Not Recorded
4  Z                    Not Recorded
5  W                    Not Recorded
6  P                    Not Recorded
7  R                    Not Recorded
```

c. Select the position component you want to set:

- **To set configuration, C,** move the cursor to C and press ENTER. Select the configuration settings you want and set them. Press PREV when you are finished setting configuration.

- **To set X, Y, Z, W, P, or R,** move the cursor to the component and use the numeric keys to type the value. Press ENTER.

d. Press PREV to return to the top level KAREL Posns screen.

7. If you want to move the robot to a taught position:

a. On the top level KAREL Posns screen, move the cursor to the position to which you want to move the robot.

Note The position you want to move the robot to must have been previously taught.

b. Press and hold in the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.

> **Warning**
>
> **In the next step, the robot will move. Make sure that personnel and unnecessary equipment are out of the workcell, otherwise, personnel could be injured and equipment damaged.**

c. Press the key for the type of move you want the robot to perform:

- **For a linear move,** press and hold in the SHIFT key and press F4, MOVE_LN. When the robot starts moving, you can release F4 but you must continue pressing the SHIFT key.

- **For a joint move,** press and hold in the SHIFT key and press F5, MOVE_JT. When the robot starts moving, you can release F5 but you must continue pressing the SHIFT key.

8. To save the positions to a file:

a. From any of the KAREL Posn screens, press FCTN.

b. Select SAVE. All the variables in the selected KAREL program will be saved to the file,*PROGRAM.VR,* on the default device. Refer to Chapter 9 *PROGRAM AND FILE MANIPULATION* for information on setting the device.

3.14.3 KAREL Paths

Before you can run a KAREL program or run production using a KAREL program, you must *teach* all KAREL paths within the program. You can use two methods to teach a KAREL path:

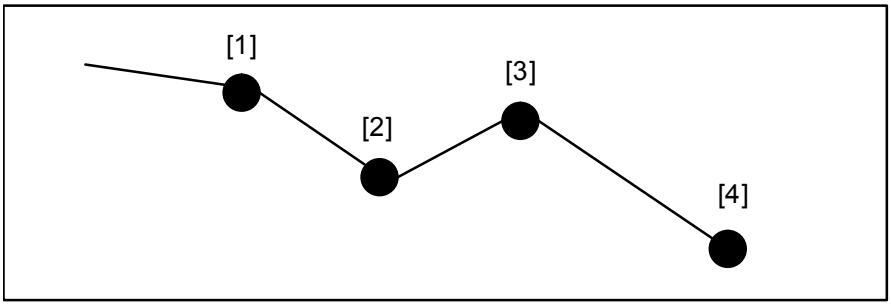
- Jog the robot to the position of each path node and **record** it.
- **Set** the value of each position component of each path node.

This section contains a procedure for teaching KAREL paths. Refer to Chapter 2 *TURNING ON AND JOGGING THE ROBOT* for information on jogging the robot. Refer to Chapter 6 *PROGRAM ELEMENTS* for information on the components of a position.

KAREL Path

A *path* can be thought of as a list of positions that has one name for the entire list. A *path node* is an individual position in the list. See Figure 3–35 .

Figure 3–35. Path with Four Path Nodes



You refer to an individual path node by specifying its number in the list. The number follows the path variable name and is enclosed in brackets (**[ ]**) . **WELD_PATH[4]**, for example, is the fourth node in the path called **WELD_PATH** . You can have a standard path node, user-defined path node, and a user-defined header.

When you use a path to represent a series of positions, you do not need to know in advance how many positions the path includes. When you teach the path, you can use as many positions as you need. Each position you teach is added to the list of nodes for a path.

In a path you can do the following:

- **Append** or add new path nodes to the end of the path
- **Insert** new path nodes any place in the path except at the end
- **Delete** or remove path nodes from the path

You can also teach associated data that accompanies each path. In addition, if you have specified path header information, you can teach the path header information when teaching other path information.

Standard KAREL Path Node

A standard KAREL path node contains a list of the standard node types. Standard path node types are

- NODE_POS
- GROUP_ASSOC
- COMMON_ASSOC

NODE_POS is the position of the path node. It contains positional components you can teach by recording or by setting values.

GROUP_ASSOC and **COMMON_ASSOC** are *path associated data*. Path associated data defines information specific to the path node. Refer to the *KAREL Reference Manual* for more information about path associated data.

User-Defined KAREL Path Node

A user-defined path node contains any information you defined in the KAREL program. Refer to the *KAREL Reference Manual* for more information about user-defined path nodes.

User-Defined Header

A user-defined *header* contains more information associated with a path. The header is defined in the KAREL program as part of the path, if desired. You can set header information outside of a KAREL program, if necessary. Refer to the *KAREL Reference Manual* for more information about user-defined path headers.

Use Procedure 3-54 to teach a KAREL path.

Procedure 3-54 Teaching KAREL Paths

Conditions

- All personnel and unnecessary equipment are out of the workcell.
- The KAREL program that contains the paths you are teaching has been loaded into controller memory. Refer to Chapter 9 *PROGRAM AND FILE MANIPULATION* .

Steps

1. Select the KAREL program for which you want to modify KAREL variables:
   a. Press SELECT.
   b. Move the cursor to the KAREL program you want and press ENTER.
2. Press DATA.
3. Press F1, [TYPE].
4. Select KAREL Posns. See the following screen for an example of the *top level* KAREL Posns screen. It lists both position and path variables.

```
DATA KAREL Posns
1 POS1          G1  Not Recorded
2 POS2          G2  Not Recorded
3 WPR1          G1  Not Recorded
4 POS_ARR1      G1  [5] of Position
5 WPR_ARR2      G1  [4,5] of XYZWPR
6 VEC1              Vector
7 PTH1              2 nodes
8 PTH2              0 nodes
9 PTH3              0 nodes
10 PTH4             3 nodes
```

Note G1 indicates that the position is in motion group 1.

@, when displayed, indicates that the robot is currently near the position.

Note If KAREL paths are not displayed, make sure the $KAREL_ENB system variable is set to 1.

5. Move the cursor to the path you want to teach and press ENTER. You will see a screen similar to the following.

```
DATA KAREL Posns
PTH2
1   [1]                        *
```

A list of path nodes is displayed. If this is a new path and no nodes exist one node will be created for you automatically. If you do not record or add any nodes during your path editing session and you press PREV, this node will be deleted.

Note * indicates the path node has not been recorded. R, when displayed, indicates the path node has been recorded.

@, when displayed, indicates the path node has been recorded and the robot is near that node.

Note You can now teach each path node by recording it or by setting position components. After you have taught a path node you can test it by moving the robot to it.

6. To teach a path node by recording it:

a. Move the cursor to the path node you want to teach.

b. Press and hold in the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.

c. Jog the robot to the position you want to record.

d. Press and hold the SHIFT key and press F3, RECORD.

Note Recording a node moves the cursor to the next node in the list automatically. If you record the last node, a new node is appended to the end of the list automatically. If you do not record this node and you press PREV, this new node will be deleted.

e. Repeat Step 6a through Step 6d for each path node you want to record.

7. To teach a path node by setting position components:

a. Move the cursor to the path node you want to teach.

b. Press ENTER. See the following screen for an example of a standard path node.

```
DATA KAREL Posns
PTH2[1]
1 NODE_POS            G1 Not Recorded
2 GROUP_DATA             Group Assoc
3 COMMON_DATA            Common Assoc
```

c. Select NODE_POS and press ENTER. You will see a screen similar to the following.

```
DATA KAREL Posns
PTH2[1].NODE_POS IN GROUP[1]
1  C             Not Recorded
2  X             Not Recorded
3  Y             Not Recorded
4  Z             Not Recorded
5  W             Not Recorded
6  P             Not Recorded
7  R             Not Recorded
```

d. Select the position component you want to set:

- **To set configuration, C,** move the cursor to C and press ENTER. Select the configuration settings you want and set them. Press PREV when you are finished setting configuration.

- **To set X, Y, Z, W, P, or R,** move the cursor to the component and use the numeric keys to type the value. Press ENTER.

e. Press PREV until the following screen is displayed (list of node information).

```
DATA KAREL Posns
PTH2[1]
1 NODE_POS            G1 Recorded
2 GROUP_DATA             Group Assoc
3 COMMON_DATA            Common Assoc
```

f. **To display information for the next path node,** press F2, NXNODE.

g. **To display information for the previous path node,** press NEXT,>, and press F2, PRNODE.

h. **To display information for a specific path node,** at the list of path nodes, press NEXT, >, press F3, IXNODE, and then enter the number of the path node you want.

i. Repeat Step 7 Step 7a through Step 7 Step 7d for each node in the path.

8. To teach path associated data:

a. Move the cursor to the path node you want to teach.

b. Press ENTER. See the following screen for an example of a standard path node.

```
DATA KAREL Posns
PTH2[1]
1 NODE_POS             G1 Recorded
2 GROUP_DATA              Group Assoc
3 COMMON_DATA             Common Assoc
```

c. Select the associated data you want to teach (GROUP_DATA, COMMON_DATA, or other user-defined data) and press ENTER. You will see a screen similar to the following.

```
DATA KAREL Posns
PTH2[1].GROUP_DATA
1 SEGRELSPEED           0
2 SEGMOTYPE
3 SEGORIENTYPE
4 SEGBREAK              FALSE
```

d. Select an item and set it to the value you want.

e. When you are finished with this group of data, press PREV.

f. Repeat Step 8a through Step 8d for each group of associated data you want to teach.

9. To teach path header information:

a. Press PREV until the top level KAREL Posns screen is displayed. See the following screen for an example.

Note You can teach path header information only if you have defined it in your KAREL program. If HEADER appears as the function key label for F2 when you select a path, path header information has been defined and you can teach it.

```
DATA KAREL Posns
1 POS1          G1  Not Recorded
2 POS2          G2  Not Recorded
3 WPR1          G1  Not Recorded
4 POS_ARR1      G1  [5] of Position
5 WPR_ARR2      G1  [4,5] of XYZWPR
6 VEC1              Vector
7 PTH1              2 nodes
8 PTH2              4 nodes
9 PTH3              0 nodes
10 PTH4             3 nodes
```

Note G1 indicates that the position is in motion group 1.

@, when displayed, indicates that the robot is currently near the position.

b. Move the cursor to the path for which you want to modify header information.

c. Press F2, HEADER. You will see a screen similar to the following.

```
DATA KAREL Posns
PTH2.PATHHEADER
1 INT1                  *uninit*
2 INT2                  *uninit*
3 S1                    STRUCT1_T
```

d. Select the item you want to teach and enter the necessary information.

e. When you are finished editing the path header information, press PREV to return to the top level KAREL Posns screen.

10. Press PREV until the top level KAREL Posns screen is displayed. See the following screen for an example.

```
DATA KAREL Posns
1 POS1          G1  Not Recorded
2 POS2          G2  Not Recorded
3 WPR1          G1  Not Recorded
4 POS_ARR1      G1  [5] of Position
5 WPR_ARR2      G1  [4,5] of XYZWPR
6 VEC1              Vector
7 PTH1              2 nodes
8 PTH2              4 nodes
9 PTH3              0 nodes
10 PTH4             3 nodes
```

Note G1 indicates that the position is in motion group 1.

@, when displayed, indicates that the robot is currently near the position.

11. **If you want to modify a path (append, delete, or insert path nodes),** select the path you want to modify and press ENTER. You will see a screen similar to the following.

```
DATA KAREL Posns
PTH2
1   [1]                     R
2   [2]                     R
3   [3]                     R
4   [4]                     R
```

Note R indicates that the path node has been recorded.

* indicates that the path node has not been recorded.

@ indicates that the path node has been recorded and the robot is near that node.

12. **To append a path node** to the end of the list:
    a. Move the cursor to any path node.
    b. Press NEXT, >.
    c. Press F2, APPEND. A node is appended to the end of the list. The cursor is moved to the newly appended node.
    d. Teach the path node by recording or setting it.
13. **To delete a path node:**

Note You cannot delete a path node if it is the only one in the list.

a. Move the cursor to the path node you want to delete.

b. Press NEXT, >.

c. Press F3, DELETE.

14. To insert a path node between two path nodes:

a. Decide where you want to insert the path node. Move the cursor to select the path node following that path node. For example, if you want to insert a path node between nodes 5 and 6, place the cursor on path node 6.

b. Press NEXT, >.

c. Press F4, INSERT. A node is inserted into the list. The cursor is moved to the newly inserted node.

d. Teach the path node by recording or setting it.

15. If you want to mode the robot to a taught path node:

a. On the top level KAREL Posns screen, move the cursor to the path you want to use and press ENTER to display the list of path nodes. See the following screen for an example.

```
DATA KAREL Posns
PTH2
1   [1]                      R
2   [2]                      R
3   [3]                      R
4   [4]                      R
```

Note The path node you want to move the robot to must have been previously taught.

b. Press and hold in the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.

c. Set the speed to a low value for safety.



Warning

In the next step, the robot will move. Make sure that personnel and unnecessary equipment are out of the workcell, otherwise, personnel could be injured and equipment damaged.

d. Press the key for the type of move you want the robot to perform:

- **For a linear move,** press and hold in the SHIFT key and press F4, MOVE_LN. When the robot starts moving, you can release F4 but must continue pressing SHIFT.

- **For a joint move,** press and hold in the SHIFT key and press F5, MOVE_JT. When the robot starts moving, you can release F5 but must continue pressing SHIFT.

16. To save the path to a file:

a. From any of the KAREL Posn screens, press FCTN.

b. Select SAVE. All the variables in the selected KAREL program will be saved to the file, *PROGRAM.VR,*on the default device. Refer to Section 9.1.3 for information on setting the device.

3.14.4 KAREL Variables

KAREL variables are created and defined within a KAREL program. You can set variables used in KAREL programs outside of the program, if necessary.

Refer to Table 3–50 for information on the items on the KAREL Vars screen. Use Procedure 3-55 to set KAREL variables outside of a program. Refer to the *KAREL Reference Manual* for more information about KAREL variables.

Table 3–50. DATA KAREL Vars Screen Items

ITEM	DESCRIPTION
Variable name	This item is a listing of the KAREL variables in the selected program.
Variable type	This item is a listing of the data type for each variable. To set the variable, move the cursor to the variable you want to change, press ENTER, and type the new value.

Procedure 3-55 Modifying KAREL Variables

Conditions

- The KAREL program that contains the variables you are modifying has been loaded into controller memory. Refer to Chapter 9 *PROGRAM AND FILE MANIPULATION* .

Steps

1. Select the KAREL program whose variables you want to modify:

a. Press SELECT.

b. Press F1, [TYPE].

c. Select KAREL Progs.

d. Move the cursor to the KAREL program you want and press ENTER.

2. Press DATA.

3. Press F1, [TYPE].
4. Select KAREL Vars. You will see a screen similar to the following.

```
DATA KAREL Vars
1 I                 *uninit*
2 I_ARR1            [10] of Integer
3 R                 *uninit*
4 R_ARR2            [3,4] of Real
5 BOOL              *uninit*
6 STR               *uninit*
7 S1                STRUCT1_T
8 S2                STRUCT2_T
9 S1_ARR1           [3] of STRUCT1_T
10 S2_ARR3           [3,4,5] of STRUCT2_T
```

Note If KAREL variables are not displayed, make sure the $KAREL_ENB system variable is set to 1.

5. Move the cursor to the variable you want to set, press ENTER, and type the necessary information.
6. **If the variable is an array,** a list of array elements is displayed or press PREV to return to the top level KAREL variables screen. Move the cursor to the element or field you want to set and press ENTER. If the variable is a structure, a list of fields is displayed. Type the necessary information.
7. **To save the variables to a file:**
   a. Press MENUS.
   b. Select FILE.
   c. Press F1, [TYPE].
   d. Select File.
   e. Press F5, [UTIL].
   f. Select Set Device.
   g. Move the cursor to the device you want and press ENTER.
   h. From any of the KAREL Posn screens, press FCTN.
   i. Select SAVE. All the variables in the selected KAREL program will be saved to the file, *PROGRAM.VR,* on the default device.

3.15 SYSTEM CONFIGURATION SETUP

The system configuration menu contains items that must be set when the system is installed. Table 3–51 contains the screen items from the System Configuration Setting screen, a detailed description of each item, and any related system variable information.

The Use PPABN signal DETAIL screen allows you to specify if the pneumatic pressure alarm is detected for each motion group.

The Default logical command DETAIL screen allows you to set the default logic instruction function keys and indicate how many lines of default logic instructions will be inserted into a teach pendant program when the function keys are pressed.

The Output when wait on Input DETAIL screen allows you to set up the Output when waiting on Input feature. This feature provides a way to turn on a digital output which indicates that a WAIT instruction in a teach pendant program has been waiting on a specified digital or group input for longer than a specified time.

The Hand Broken DETAIL screen allows you to set Hand Broken for each motion group.

Table 3–52 through Table 3–55 contain DETAIL screen information.

Refer to Section 4.9 for more information on UOP signals. Use Procedure 3-56 to set up system configuration.

For more information about the RESET Chain Failure alarm, refer to Section A.3.5 .

Table 3–51. System Configuration Setup Screen Items

Screen Item	Description	Related System Variable
Use HOT START	When power-failure handling is enabled, hot start is performed when the controller is turned ON.	$SEMIPOWERFL
I/O power fail recovery	This item specifies how the recovery of I/O is performed when hot start is valid. This also specifies how the recovery of simulated I/O is performed when hot start is invalid. There are four modes in hot start: • NOT RECOVER • RECOVER SIM • UNSIMULATE • RECOVER ALL	$PWF_IO

Table 3–51. System Configuration Setup Screen Items (Cont'd)

Screen Item	Description	Related System Variable
NOT RECOVER	Recovery of Input/Output signals cannot be performed regardless of the setting for hot start. All outputs are turned off and simulated mode is released.	$PWF_IO = 1
RECOVER SIM	Simulated mode is recovered regardless of the setting for hot start. However, all real output signals and simulated Input/Output signals are turned off.	$PWF_IO = 2
UNSIMULATE	Output signals are recovered when hot start is valid, but all simulated modes are released. Since output signals are not recovered when hot start is invalid, this causes the same result as NOT RECOVER.	$PWF_IO = 3
RECOVER ALL	Input/Output signals are recovered when hot start is valid. Output signals and simulated mode are in the same state as they were when the controller was turned off. Since output signals are not recovered when hot start is invalid, this causes the same result as RECOVER SIM.	$PWF_IO = 4
Autoexec program for Cold start	This item specifies the program name that is executed automatically when you turn on the controller and HOT START is invalid. The specified program is executed just after turning on the controller.	$PWR_NORMAL
Autoexec program for Hot start	This item specifies the name of the program that is activated automatically when the controller is turned on and when power-failure handling is enabled. The specified program is executed immediately after the power is turned on. If the specified program is not executed after 15 seconds have elapsed, this program is aborted. **Note** Since the program activated automatically when the controller is turned on is executed before the power is applied to the servo motors, that program cannot operate the robot. Therefore, specify only a program designed specifically for system setup or for initializing I/O devices. Also, specify the attributes for the program on the detail program screen, as follows: **Group Mask : [*,*,*,*,*]** **Ignore pause : [TRUE]**	$PWR_SEMI

Table 3–51. System Configuration Setup Screen Items (Cont'd)

Screen Item	Description	Related System Variable
HOT START done signal	This completion signal for power-failure handling specifies the digital output signal (DO), issued when power-failure handling (hot start) has been completed. When power-failure handling is not done, this signal is off. When 0 is specified for this signal, this signal is not output.	$SEMIPWFDO
Restore selected program	This item specifies whether the program that was selected when the controller was turned off after a Cold start, is selected again after the controller is turned back on. When this item is enabled, the program selected when the power was turned off is selected when the power is turned back on. When this item is disabled, no program is selected when the power is turned on again.	$DEFPROG_ENB
Enable UI signals	UOP I/O signals are enabled or disabled. When disabled, peripheral unit input signals (UI[1] to UI[18]) are ignored.	$OPWORK.$uop_disable
START for CONTINUE only	When this external start signal is set to enable (TRUE), the external activation signal (START) activates only the suspended programs.	$SHELL_CFG.$cont_only
CSTOPI for ABORT	When CSTOPI forces a program to terminate, program termination by CSTOPI is enabled, the CSTOPI input immediately terminates the program that is currently being forcibly executed.	$SHELL_CFG.$use_abort

Table 3–51. System Configuration Setup Screen Items (Cont'd)

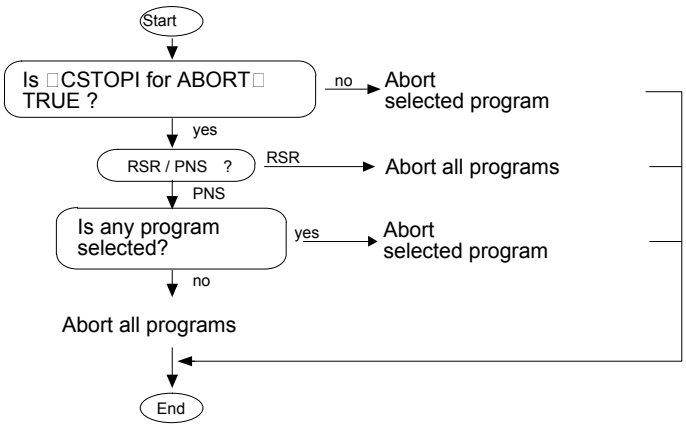
Screen Item	Description	Related System Variable
Abort all programs by CSTOPI	This item selects whether the CSTOPI signal aborts all programs in a multi-tasking environment. When TRUE is specified for this item, the CSTOPI input signal functions as follows: • Aborts all programs if RSR is selected for RSR/PNS. • Aborts the selected program if PNS is selected for RSR/PNS. • If no program is selected, however, all programs are aborted.	$SHELL_CFG.$cstopi_all
		
	When FALSE is specified for "Abort all programs by CSTOPI," the CSTOPI input signal aborts only a selected program (in the same way as the conventional specifications). When using the CSTOPI input signal in the same way as for a conventional system, specify FALSE (default setting).	
PROD_START depend on PNSTROBE	When the PROD_START input (enabled with the confirmation signal) is enabled, the PROD_START input is effective only when the PNSTROBE input is on. Enabling this item prevents the program indicated on the teach pendant, which is not to be activated, from being activated erroneously by noise or an incorrect sequence.	$SHELL_CFG.$prodstartyp
Detect FAULT_RESET signal	The detection of RESET input at a rising or falling edge specifies whether the signal is detected at a rising or falling edge. After this item has been set or modified, turn off the controller, then turn it on again to enable the setting. Cold start is automatically performed for this operation.	$SCR.$resetinvert

Table 3–51. System Configuration Setup Screen Items (Cont'd)

Screen Item	Description	Related System Variable
Use PPABN signal	Abnormal-pressure (*PPABN) detection enable or disable is specified for each motion group. Position the cursor on this item and press the ENTER key to display the screen for setting detection enable or disable for each group. When the *PPABN signal is not used, set this item to disable. After this item has been set or modified, turn off the controller, then turn it on again to enable the setting. Cold start is automatically performed for this operation.	$PARAM_GROUP.$ppabn_enbl
WAIT timeout	This item specifies the time limit for conditional wait instructions, WAIT..., Timeout, LBL[...]	$WAITTMOUT
RECEIVE timeout	This item specifies the time limit for register receive instructions, RCV R[...], Timeout, LBL[...] The time limit can be taught only when the optional sensor interface is specified.	$RCVTMOUT
Return to top of program	After a program has terminated, this item specifies whether the cursor is positioned at the start of the program upon termination of that program. When this item is disabled, the cursor remains positioned at the end of the program (not positioned at the start of the program) upon termination of the program.	$PNS_END_CUR
Original program name (F1-F5)	This item specifies the words displayed for the soft keys on the program creation screen. It is convenient to specify words that are frequently used for program names.	$PGINP_WORD[1] to [5]
Default logical command	This item displays the screen used to set the default logical instruction function keys, or **hot keys** . First you must define the function key name and then you can assign instructions to it. You can define a set of 0-4 instructions on softkeys F2, F3, and F4 of the INST menu. You can also set a label for the softkey. The entries on the screen refer to softkey F2, F3, and F4 on the [INST] menu, which are blank by default. Installing a name such as time in the Name field causes the function key to have that label. The Lines item allows you to set 0-4 lines that will display the user-selected program instructions. With the NAME field highlighted, press ENTER to edit the softkey label. When you are finished, press ENTER again. Highlight the Lines item, and type the number of instructions that you want to display when the softkey is pressed. To add the instructions, select an existing program, or create a new program. The function key items that you set up will be displayed in the editor. Press the newly-defined function key and a popup window with the instructions will be displayed. Initially there will be no instructions available. To insert instructions, press the ED_DEF key. Press the INST key to insert the instructions associated to the Default logical command. When you are finished with installing program instructions, press the F5, DONE key.	

Table 3–51. System Configuration Setup Screen Items (Cont'd)

Screen Item	Description	Related System Variable
Maximum of ACC instruction	This item specifies the maximum for the override specified with an acceleration or deceleration override instruction, ACC.	$ACC_MINLMT
Minimum of ACC instruction	This item specifies the minimum for the override specified with an acceleration or deceleration override instruction, ACC.	
WJNT for default motion	This function adds a Wjnt motion option to the default motion instruction such as linear or circular, or deletes it at the same time.-When F4, [ADD], is pressed, the Wjnt additional instruction is added to every default linear or circular motion instruction. Then, the display on the system configuration screen switches from DELETE (or ******) to ADD. In the prompt line, "Added WJNT to default motion" appears.-When F5, [DELETE], is pressed, the Wjnt additional instruction is deleted from every default linear or circular motion instruction. Then, the display on the system configuration screen switches from ADD (or ******) to DELETE. In the prompt line, "Deleted WJNT from default motion" appears.	
Auto display of alarm menu	This item selects whether the alarm screen is displayed automatically. When TRUE is specified for this item, the alarm screen is displayed automatically. The default setting is FALSE. When this item has been set or modified, turn off the controller, then turn it on again to enable the setting.	$ER_AUTO_ENB
Force Message	When the teach pendant MESSAGE instruction is executed, the USER screen is displayed automatically, by default (Force Message = ENABLE). If you set Force Message to DISABLE, the message is written to the USER screen, but the screen does not change to the USER screen automatically. If you set Force Message to ENBL (TP OFF), then the user page is displayed automatically only if the teach pendant ENABLE (ON/OFF) keyswitch is set to OFF.	
Reset CHAIN FAILURE Detection	This item resets the chain failure detection fault and requires you to press RESET to clear the alarm. This item works the same as pressing F4 "RES_1CH" in the Active Alarm screen.	
Allow Force I/O in AUTO mode	This item can be set to either TRUE or FALSE. When the controller is in AUTO mode, and this item is set to TRUE, you can change the status of any I/O port and simulate or unsimulate I/O at the desired port.	$AUTOMODE_DO
Allow Chg. Ovrd in AUTO mode	If this is set to TRUE, it is possible to change speed override while the controller is in AUTO mode. Otherwise, this is not permitted.	$AUTOMODE_OV

Table 3–51. System Configuration Setup Screen Items (Cont'd)

Screen Item	Description	Related System Variable
Signal to set in AUTO Mode	If this is non-zero, this is the number of a digital output signal which is to be automatically set TRUE when the controller is in AUTO mode. Otherwise, it is set to FALSE.	
Signal to set in T1 mode	If this is non-zero, this is the number of a digital output signal which is to be automatically set TRUE when the controller is in T1 mode. Otherwise, it is set to FALSE.	
Signal to set in T2 mode	If this is non-zero, this is the number of a digital output signal which is to be automatically set TRUE when the controller is in T2 mode. Otherwise, it is set to FALSE.	
Signal to set if E-Stop	If this is non-zero, this is the number of a digital output signal which is to be automatically set TRUE when one of the emergency stop conditions is asserted. Otherwise, it is set to FALSE.	
Set if INPUT SIMULATED	This input allows you to use an output to monitor whether any input is simulated. You must set this item to the index number of the output that will be turned on when the digital, group, robot, or analog input is simulated. You must turn off the controller and turn it on again to activate this output.	$INPT_SIM_DO
Sim. Input Wait Delay	This item is used by the Simulated Input Skip feature. It is the time in seconds that a WAIT instruction will delay before continuing when Simulated Input Skip is enabled. Valid settings are from 0 to 99.99.	$SIMWAITTIM
Set if Sim. Skip Enabled	This item is used by the Simulated Input Skip feature. It is the index of a digital output that will be turned on to indicate that Simulated Input Skip is enabled. Valid numbers for this item are 0 through 32766.	$SIMWAITOUT
Set when prompt displayed	This item provides a digital output signal from the controller that indicates that a prompt box is being displayed on the teach pendant. This feature is used to provide an indicator to an operator that an action must be taken at the user interface. The output signal is turned on when the prompt box is displayed and turned off when the prompt box is acknowledged and removed. Valid numbers for this item are from 0 through 32766. After setting this item, you must cycle controller power for the change to take effect.	None

Table 3–51. System Configuration Setup Screen Items (Cont'd)

Screen Item	Description	Related System Variable
Output when WAIT on input	This item displays the screen used to set up the Output when waiting on Input feature. Output when waiting on Input provides a way to turn on a digital output that indicates that a WAIT instruction in a teach pendant program has been waiting on a specified digital or group input for longer than a specified time. The feature allows you to define a range of inputs to be monitored, a timeout value and a digital output that will be turned on to indicate that a timeout has occurred. Separate settings are provided for digital and group inputs. With this feature enabled, if a teach pendant program is run and that program executes a WAIT instruction, the instruction will be checked for condition clauses that include a digital or group input that has an index within the range to be monitored. If the condition clauses containing those inputs are not satisfied within the timeout period and the overall WAIT has not been satisfied, the specified digital output will turn on. The following condition clauses are valid and may be combined within a single WAIT instruction: DI[]=ON, DI[]=OFF, DI[]=ON+, DI[]=OFF-, GI[]=constant, GI[]<>constant, GI[]<constant, GI[]<=constant, GI[]>constant, GI[]>=constant.	None
Signal if OVERRIDE = 100	If this is non-zero, this is the number of a digital output signal which is to be automatically set TRUE when the OVERRIDE equals 100. Otherwise, it is set to FALSE.	None
Hand Broken	If this is set to ENABLED, a Hand Broken circuit must be connected to the robot. Then, when a Hand Broken signal is received, a Hand Broken error is posted. If this is set to DISABLED, there is no need to connect a Hand Broken circuit to the robot. If a circuit is connected, an error is posted indicating that the setting is incorrect. Hand Broken must be set for each motion group configured on the robot.	$SCR_GRP[ ].$HBK_ENBL

Table 3–51. System Configuration Setup Screen Items (Cont'd)

Screen Item	Description	Related System Variable
Remote/Local Setup	This item specifies the state of the SI[2:Remote] input or designates another signal that controls this input. When SI[2:Remote] is OFF, the Cycle Start switch on the operator's panel can be used to initiate program execution. If SI[2:Remote] is ON, a remote signal such as from UOP can start a program. The possible selections are: Remote: The SI[2:Remote] signal is always ON Local: The SI[2:Remote] signal is always OFF External I/O: The SI[2:Remote] signal is controlled by an external I/O port (defined below)	$REMOTE_CFG.$REMOTE _TYPE
External I/O (ON:Remote)	When Remote/Local Setup is set to External I/O, this setting determines what I/O port controls the Remote/Local input SI[2:Remote]. The possible selections are: DI, DO, RO, RO, UI, UO	$REMOTE_CFG. $REMOTE_IOTYP $REMOTE_CFG. $REMOTE_IOIDX

Table 3–51. System Configuration Setup Screen Items (Cont'd)

Screen Item	Description	Related System Variable
UOP Auto Assignment	The possible settings are None, Full, Full (Slave), Full (CRM79), Simple, Simple (Slave), Simple (CRM79), and Simple (CRM81). Full (Slave), Full (CRM79), Simple (Slave), Simple (CRM79), and Simple (CRM81) are used only on a Mate Controller. None indicates that no UOP signals are being mapped. Full indicates that all UOP signals are being mapped. Simple maps RESET and CSTOPI signals to the same input. Simple also maps the PNSTROBE and Start signals to the same input. PNSTROB occurs on the rising edge and Start occurs on the falling edge of this signal. Simple(CRM79) maps UOP signals to the CRM79 port on the Mate CPU This is the 50 Pin Honda connector. Simple (CRM81) maps UOP signals to the CRM81 port on the Mate CPU. This is the 20 Pin Amp connector with lock tabs. Full(Slave) and Simple(Slave) assign UOP to an I/O link slave (Rack 32, Slot 1) in Mate controllers only. In other robot controllers, "Simple" and "Simple(Slave)" map to the same I/O points. In robot controllers, "Full" and "Full(Slave)" map the same I/O points.	
Multiple-program selection (Multi Program Selection)	This setting is used to switch the program selection method between *single task mode* and *multitask mode*. If this setting is enabled, multitask mode is selected. This means that you can select and view/edit program X while program Y is running or paused. If it is disabled, single task mode is selected. This means that you can only select and view/edit the currently running or paused program. If the R651 standard settings are ordered, the setting is disabled by default, and if the R650 North America only settings are ordered, it is enabled by default. When the setting is changed, the change is immediately applied.	

Table 3–52. Use PPABN Signal DETAIL Screen Items

ITEM	DESCRIPTION
GROUP1 - GROUP5 Values: TRUE, FALSE Default: FALSE	This item allows you to set whether the pneumatic pressure alarm is detected for each motion group.

Table 3–53. Default Logical Command DETAIL Screen Items

ITEM	DESCRIPTION
2-F2 / 2-F3 / 2-F4	These items are the logical command function keys.
Function(page-key)	This item lists the available functions keys. These function keys can be accessed from page two of the function keys while you are modifying a teach pendant program. [POINT] is on page one.
Name	This item specifies the name that has been assigned to the function key. You can change this value.
Lines Range: 0 - 4	This item specifies how many lines of logic instructions should be inserted into a teach pendant program. You can change this value. If this item is set to 0, that function key cannot be used.

Table 3–54. Output when waiting on Input DETAIL Screen Items

ITEM	DESCRIPTION
WAIT for DI range: Min: 0 Max: 32766	This item indicates the starting and ending index of the digital inputs to be monitored. In order to enable this feature, the starting index must be less than or equal to the ending index. If the starting index is greater than the ending index, the feature is disabled for digital inputs. If the starting and ending index are both 0, then all digital inputs are monitored.
WAIT for DI time Min: 0 Max: 99.99	This setting is the timeout period in seconds after which the digital output specified in 'WAIT for DI output' below will turn on.
WAIT for DI output Min: 0 Max: 32766	This setting is the index of a digital output that will be turned on after the timeout period. In order to enable this feature for digital inputs, this setting must be greater than 0.
WAIT for GI range: Min: 0 Max: 32766	This item indicates the starting and ending index of the group inputs to be monitored. In order to enable this feature, the starting index must be less than or equal to the ending index. If the starting index is greater than the ending index, the feature is disabled for group inputs. If the starting and ending index are both 0, then all group inputs are monitored.

Table 3–54. Output when waiting on Input DETAIL Screen Items (Cont'd)

WAIT for GI time Min: 0 Max: 99.99	This setting is the timeout period in seconds after which the digital output specified in 'WAIT for GI output' below will turn on.
WAIT for GI output Min: 0 Max: 32766	This setting is the index of a digital output that will be turned on after the timeout period. In order to enable this feature for group inputs, this setting must be greater than 0.

Table 3–55. Hand Broken DETAIL Screen Items

ITEM	DESCRIPTION
GROUP1 - GROUP2 Values: ENABLE, DISABLE Default: ENABLE	This item allows you to set whether the pneumatic pressure alarm is detected for each motion group.

Procedure 3-56 Setting Up System Configuration

Steps

1. Press MENUS.
2. Select SYSTEM.
3. Press F1, [TYPE].
4. Select Config. You will see a screen similar to the following.

```
System/Config
1 Use HOT START:                 FALSE
2 I/O power fail recovery:RECOVER ALL
3 Autoexec program          [********]
for Cold start:
4 Autoexec program          [ATHSTART]
for Hot start:
5 HOT START done signal:     DO[    0]
6 Restore selected program:     TRUE
7 Enable UI signals:            TRUE
8 START for CONTINUE only:      FALSE
9 CSTOPI for ABORT:             FALSE
10 Abort all programs by CSTOPI: FALSE
11 PROD_START depend on PNSTROBE:FALSE
12 Detect FAULT_RESET signal:    FALL
13 Use PPABN signal:         <*GROUPS*>
14 WAIT timeout:              30.00 sec
15 RECEIVE timeout:           30.00 sec
16 Retun to top of program:      TRUE
17 Original program name(F1): [PRG    ]
18 Original program name(F2): [MAIN   ]
19 Original program name(F3): [SUB    ]
20 Original program name(F4): [TEST   ]
21 Original program name(F5): [*******]
22 Default logical command:  <*DETAIL*>
23 Maximum of ACC instruction:   150
24 Minimum of ACC instruction:     0
25 WJNT for default motion:      ******
26 Auto display of alarm menu:   FALSE
27 Force Message:         ENABLE
28 Reset CHAIN FAILURE detection:FALSE
29 Allow Force I/O in AUTO mode: TRUE
30 Allow chg. ovrd. in AUTO mode:TRUE
31 Signal to set in AUTO mode DO[    0]
32 Signal to set in T1 mode   DO[    0]
33 Signal to set in T2 mode   DO[    0]
34 Signal to set if E-STOP    DO[    0]
35 Set if INPUT SIMULATED     DO[    0]
36 Sim. Input Wait Delay:      0.00 sec
37 Set if Sim. Skip Enabled   DO[    0]
38 Set when prompt displayed  DO[    0]
39 Signal if OVERRIDE = 100   DO[    0]
40 Hand Broken:              <*GROUPS*>
41 Remote/Local Setup:     OP panel key
42   External I/O(ON:Remote):  DI[  0]
43  UOP Auto assignment:  Full
44 Multi Program Selection        TRUE
```

5. Move the cursor to the item you want to set, then enter a value by using the numeric keys or function keys on the teach pendant. Refer to Table 3–51 for detailed information about each field on the screen.

- To type a character string, move the cursor to the item, then press ENTER. You can then type the necessary characters.
- To set abnormal-pressure detection or default logic instructions, move the cursor to <*GROUPS*> or <*DETAIL*> then press ENTER. The screen for setting the corresponding item appears. Press the PREV key to display the System Config screen.

6. If you set or modify an item that requires a Cold start to take effect, the following message appears on the screen "please power on again." You must perform a Cold start. Turn OFF the controller and then turn it on again. See the following screen for an example.

```
System/Config
12: Detect FAULT_RESET signal          FALL
13: Use PPABN signal:              <*GROUPS*>
14: WAIT timeout:                   30.00 sec
Please power on again
```

Chapter 4

INPUT/OUTPUT (I/O) SETUP

Contents

Chapter 4	INPUT/OUTPUT (I/O) SETUP	4–1
4.1	OVERVIEW	4–3
4.1.1	Overview	4–3
4.1.2	Hardware	4–3
4.1.3	Kinds of I/O	4–3
4.2	SETTING UP I/O	4–5
4.2.1	Overview	4–5
4.2.2	Configuring I/O	4–6
4.2.3	Adding Comments About I/O	4–10
4.2.4	Complementary Output Signals and Polarity	4–10
4.2.5	Simulating I/O	4–10
4.2.6	Controlling Outputs	4–10
4.2.7	Memory Image Port Assignment	4–11
4.2.8	Procedure for Setting Up I/O	4–11
4.2.9	Custom I/O	4–27
4.3	DISTRIBUTED (MODEL B) I/O SETUP	4–29
4.3.1	Overview	4–29
4.3.2	Setting the DIP Switches	4–31
4.3.3	Setting Up the Basic Digital I/O Units	4–34
4.3.4	Setting Up User I/O	4–36
4.4	ROBOT I/O SETUP	4–37
4.5	STANDARD OPERATOR PANEL (SOP) I/O SETUP	4–41
4.6	SETTING UP I/O LINK DEVICES	4–45
4.6.1	Overview	4–45
4.6.2	I/O Link Devices	4–45
4.6.3	Process I/O	4–45
4.6.4	Model A I/O	4–46
4.6.5	Model B I/O	4–47
4.6.6	I/O Link Device Screen	4–47

4.6.7 I/O Link Connection 4–50
4.6.8 FANUC I/O Link Connection Unit 4–52
4.6.9 Setting the Number of Ports 4–60
4.7 I/O INTERCONNECT SETUP 4–62
4.8 CONTROLLING I/O 4–68
4.8.1 Overview 4–68
4.8.2 Forcing Outputs 4–68
4.8.3 Simulating Inputs and Outputs 4–69
4.8.4 Output When WAITing on Input 4–70
4.9 USER OPERATOR PANEL (UOP) SIGNAL DEFINITION 4–72
4.9.1 Overview 4–72
4.9.2 UOP Input Signals 4–75
4.9.3 UOP Output Signals 4–80
4.10 HANDLINGTOOL CELL INTERFACE I/O SIGNALS 4–83
4.10.1 Overview 4–83
4.10.2 HandlingTool Cell Interface I/O Signals 4–84

4.1 OVERVIEW

4.1.1 Overview

Note The DeviceNet interface I/O is also available. For further information about this kinds of I/O refer to the *DeviceNet Interface Setup and Operations Manual* .

Inputs and outputs (I/O) are electrical signals that enable the controller to communicate with the robot, end-of-arm tooling, and other external devices, such as sensors and actuators.

To use I/O, you must do the following:

1. Identify the I/O hardware you need.
2. Install the I/O hardware, if necessary.
3. Connect the I/O hardware to the necessary devices.
4. **Configure the I/O using the teach pendant. This informs the controller software of the I/O you installed and connected so it can be used by the robot.**

This chapter contains information about configuring the association between physical devices and I/O signal numbers. Refer to the *Controller Maintenance Manual* for information about installing, connecting, and using I/O hardware.

4.1.2 Hardware

The following kinds of I/O hardware are available:

- Modular (Model A) I/O
- Distributed (Model B) I/O
- Process I/O
- Terminal I/O (ArcTool only)

Refer to the *Controller Maintenance Manual* for illustrations of and detailed information on these kinds of I/O hardware.

4.1.3 Kinds of I/O

The following kinds of I/O are available. The amounts and kinds vary depending on the kind of I/O hardware you use:

- Analog

- Digital
- Group
- User Operator Panel (UOP)
- PLC
- Standard Operator Panel (SOP)
- Robot

Analog

An analog I/O signal (AI or AO) is an input or output voltage that has a value within the range of the I/O board or module used.

Analog input devices convert external analog signals into numeric signals for use by the controller. Analog output devices allow analog signals to be send out to external devices. Typical values of analog inputs and outputs are from -10 volts to 0 volts, or from 0 volts to +10 volts. These values depend on the particular device used in your controller.

For teach pendant and KAREL programs, analog data is converted into digits and is then read by the program as numbers.

Digital

A digital I/O signal (DI or DO) is a control signal sent to or from the controller. Digital signals can have only one of two possible states: ON or OFF. Digital signals provide access to data on a single input or output signal line.

Group

Group I/O signals (GI or GO) are made up of a sequence of digital I/O signals. These are interpreted as a binary integer.

A group signal is a group of up to 16 ones (1) and zeros (0), indicated ON or OFF. Each bit in group input or output is a single digital input or output line. Unused bits are assigned a "0."

If you group your digital I/O signals, you can control or monitor the sequence of I/O signals within your program by setting or reading a single group using the group I/O instructions.

User Operator Panel (UOP) I/O

The User Operator Panel (UOP) provides 18 input signals and 20 or 24 output signals (four are optional), that can be connected to a remote device or a remote operator panel, to control the robot.

Most UOP I/O signals (UI or UO) are active when the robot is in a remote condition. Signals that affect safety are always active.

For systems with a process I/O board, the UOP signals are configured by default to dedicated ports.

Note PaintTool with Standard I/O only uses the first eight input and output signals. However, you can assign the rest of the signals as long as you have I/O available.

Note If you configure UOP input and output signals, the UI and UO physical locations are actually DI/DO on the physical I/O. The physical DI/DO can be double configured as both UI/UO and DI/DO logicals. This allows you to control or monitor UI/UO signals within your program by using the DI/DO instructions.

PLC I/O

PLC I/O (PI or PO) is an optional feature that provides signals that allow a cell controller (PLC) to access the discrete I/O within a controller directly, in addition to transferring I/O signal status information.

The cell controller views the robot I/O interface as a remote I/O rack (RIO). When you use the RIO interface along with the PLC I/O, outputs from the cell controller system are then stored to outputs from the controller output modules, and inputs into the controller input modules can be read as inputs into the cell controller I/O system.

Standard Operator Panel (SOP) I/O

SOP I/O signals (SI or SO) correspond to internal controller software panel digital input and output signals that control the operator panel on the controller. You cannot change SOP I/O assignments, but might find it to be useful to display them during troubleshooting.

Robot I/O

Robot I/O signals (RI or RO) consist of the input and output signals between the controller and the robot. These signals are sent to the EE (End Effector) connector located on the robot. The number of robot input and output signals varies depending on the number of axes in your system.

4.2 SETTING UP I/O

4.2.1 Overview

This section describes how to set up the following kinds of I/O:

- **Analog** inputs and outputs- AI[n] and AO[n]
- **Digital** inputs and outputs - DI[n] and DO[n]
- **Group** inputs and outputs - GI[n] and GO[n]
- **UOP** (User Operator Panel) inputs and outputs - UI[n] and UO[n]

These kinds of I/O signals are attached to physical ports and are accessed from programs. The [n] corresponds to a signal number or group number.

When you set up I/O, you do the following:

- Configure I/O
- Simulate I/O
- Control outputs and display inputs
- Configure polarity and complementary outputs
- Add comments about the I/O

Refer to Section 4.2.8 to set up I/O.

4.2.2 Configuring I/O

When all appropriate I/O hardware has been installed and connections have been made, you must **configure** the I/O. Configuring I/O establishes the correspondence between the signal number or group number and the physical port.

Each signal or sequence of signal numbers is configured to a *rack*, a *slot* in the rack, and the *channel number* or *starting number*, when the software is loaded on the controller. You can change this configuration. You configure I/O differently depending on the kind of I/O you are using. This section describes how to configure the following kinds of I/O: analog, digital, group, and UOP.

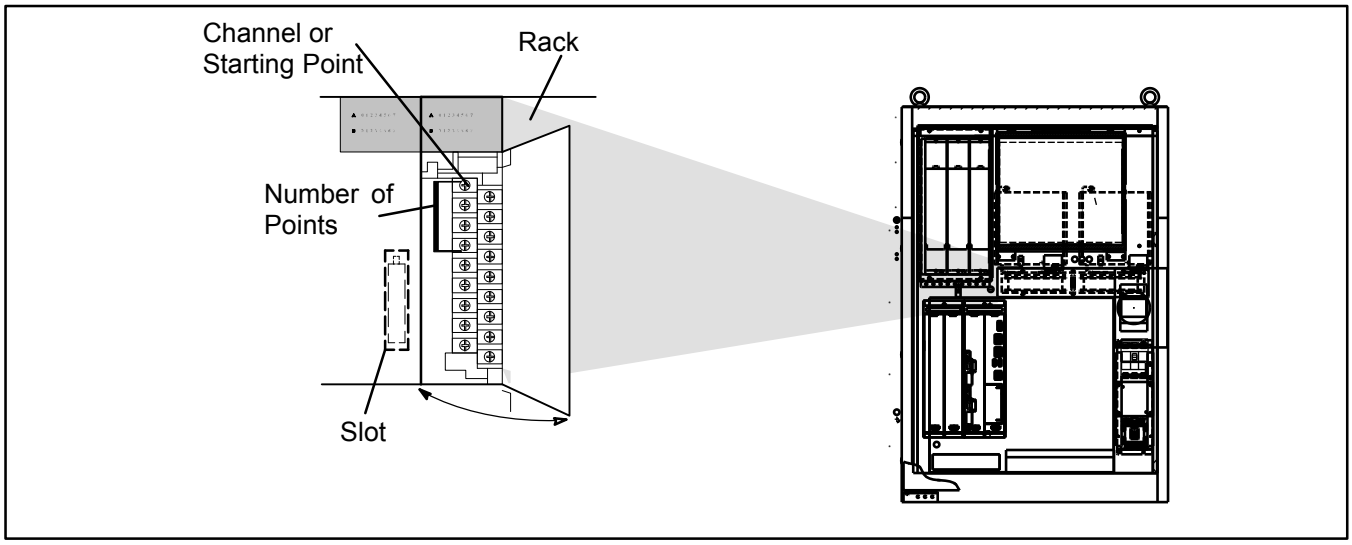
Note If you configure UOP input and output signals, the UI and UO physical locations are actually **digital** ports on the process I/O, modular I/O, distributed I/O, or remote I/O boards. In effect, the physical digital I/O can be double configured as both UOP I/O and digital I/O logicals. This allows you to control or monitor UOP I/O signals within your program using the digital I/O instructions.

When you configure I/O, you must define the following for each signal or range of signals you configure:

- First point of range - for digital and UOP I/O only
- Last point of range - for digital and UOP I/O only
- Rack
- Slot
- Channel - for analog I/O only
- Starting point - for digital, UOP, and group I/O only
- Number of points - for group I/O only

See Figure 4–1 .

Figure 4–1. Rack, Slot, Channel, and Starting Point



Default Digital I/O Configuration for Process I/O Boards

Process I/O boards use the default digital input and output configuration assignments listed in Table 4–1 .

Table 4–1. Process I/O Board Default Digital Input and Output Configuration

Range	Rack	Slot	Start Point
DI (1-22)	0	1	19
DO (1-20)	0	1	21

Default UOP I/O Configuration for Process I/O

Physical ports will be assigned automatically as UOP signals if **all** of the following conditions are satisfied:

- The system variable $IO_AUTO_UOP is set to TRUE.
- There are no existing assignments of UOP signals.
- The default hardware is present.
- Ports on the default hardware have not already been assigned.
- The default hardware has the required number of input and output ports.

The hardware devices listed in Table 4–2 are eligible for default UOP assignment.

Table 4–2. I/O Hardware Eligible for Default UOP Assignment

Hardware Type	Rack	Slot
ME-NET	64	1
Interbus-S	73	1
Field-bus	66 or 67	1
Process I/O	0	1
Modular I/O	1	1
PDI I/O (LR PaintTool only)	48	1

In all cases in Table 4–2 , assignments of UOP inputs and UOP outputs begin at physical port 1. With process I/O boards, the software provides the UOP input and output configuration assignments listed in Table 4–3 and Table 4–4 .

Table 4–3. Default UOP Input Configuration for Process I/O Boards

#	Range	Rack	Slot	Start Point
1	UI (1-8)	0	1	1
2	UI (9-16)	0	1	9
3	UI (17-18)	0	1	17

Table 4–4. Default UOP Output Configuration for Process I/O Boards

#	Range	Rack	Slot	Start Point
1	UO (1-8)	0	1	1
2	UO (9-16)	0	1	9
3	UO (17-20)	0	1	17

Rack

The rack is the physical location on which the input or output process I/O board or modular I/O module is mounted. Your system can contain multiple racks. Refer to Table 4–5 .

Table 4–5. Rack Assignments for Different Kinds of I/O

Kind of I/O	Rack Assignment
Modular (Model A) I/O	Physical location on which the input or output modules are mounted. First on SLC chain is Rack 1. • **When used without distributed (ModelB) I/O,** begins at Rack 1. • **When used with distributed (Model B) I/O,** the distributed system is Rack 1 and the modular rack is Rack 2.
Distributed (Model B) I/O	Rack 1
Process I/O and Terminal I/O (ArcTool only)	Rack 0
DeviceNet Interface	Racks 81 - 84
PDI I/O (LR PaintTool only)	Rack 48

Slot

The slot is the space on the rack where the modular I/O module is connected. The meaning of slot depends on the kind of I/O you are using. Refer to Table 4–6 .

Table 4–6. Slot Assignments for Different Kinds of I/O

Kind of I/O	Slot Assignment
Modular (Model A) I/O	The space on the rack where the I/O module is connected.
Distributed (Model B) I/O	Determined by the DIP switch settings on the unit.
Process I/O and Terminal I/O (ArcTool only)	Begin at Slot 1 for the first unit
DeviceNet Interface	The slot number is the MAC Id for the device.

Channel (for Analog I/O only)

The channel is the physical position of the port on the process I/O board or terminal number on the modular I/O module.

Starting Point (for Digital, Group, and UOP I/O only)

Starting point is the port number within the sequence of ports on the board or module.

- For **modular I/O** , the starting point number refers to the terminal number.
- On a **distributed basic I/O unit** that has both digital inputs and digital outputs, both inputs and outputs start at 1.

- For **expansion I/O units** , mixed inputs and outputs are allowed.
  - — If the base and extension are the same (both digital inputs or digital outputs), the ports on the extension module start at (n+1), where n is the number of ports in the base module.
  - — If the base and extension are different, the ports on each start at 1.

Refer to the *FANUC Robotics Controller Maintenance Manual* for the relationship between starting point numbers and pin numbers on connectors CRM2A and CRM2B on a process I/O board.

4.2.3 Adding Comments About I/O

Adding comments of up to 16 alphanumeric characters about I/O allows you to include text that describes the signal. For example, you can add a comment to indicate that a particular sensor is connected to the physical port configured to the signal. After you have added comments for I/O signals, they will be displayed in teach pendant program instructions that refer to those I/O signals.

4.2.4 Complementary Output Signals and Polarity

You can configure digital output signals to be controlled independently or in **complementary pairs** . If an output signal is controlled independently, a command to turn that output signal on or off controls only that output signal. If an output signal is controlled in a complementary pair, a command to turn that signal ON will also turn the other member of the pair OFF. A command to turn the signal OFF will also turn the other member of the pair ON.

You can configure digital input/output signals with **normal polarity** (active ON) or **inverse polarity** (active OFF).

4.2.5 Simulating I/O

Simulating I/O allows you to test a program that uses I/O. Simulating I/O does not actually send output signals or receive input signals. I/O signals can be simulated individually. If an input signal is simulated it can be set ON or OFF from the teach pendant. Refer to Section 4.8.3 .

4.2.6 Controlling Outputs

Controlling outputs allows you to set the output value and turn it on in a program or to force it on manually. I/O signals can be controlled individually. Refer to Chapter 6 *PROGRAM ELEMENTS* to turn on output signals from a teach pendant program, and Section 4.8.2 to force output signals. Refer to the *FANUC Robotics KAREL Reference Manual* to turn ON output signals from a KAREL program.

4.2.7 Memory Image Port Assignment

There are times when it is useful to assign ports to *memory image* ports. These are locations in controller memory that do not have physical connections. In particular, these ports are useful in testing systems where the actual digital I/O boards or modules are not present.

The controller has 2500 Boolean (ON/OFF) memory image ports. Digital input, digital output, UOP input, and UOP output signals can be assigned to these ports by specifying rack 0, slot 0, and starting port number 1-2500.

Similarly, the controller has 250 numeric memory image ports. Analog input, analog output, group input, and group output signals can be assigned to these ports by specifying rack 0, slot 0, and channel or starting port number 1-250.

4.2.8 Procedure for Setting Up I/O

Table 4–7 describes each Analog I/O Monitor screen item. Table 4–8 describes each Analog I/O Configure screen item.

Table 4–9 describes each Digital I/O Monitor screen item. Table 4–10 describes each Digital I/O Configure screen item. Table 4–11 describes each Digital I/O DETAIL screen item.

Table 4–12 describes each Group I/O Monitor screen item. Table 4–13 describes each Group I/O Configure screen item.

Use Procedure 4-1 to set up analog, digital, group, and UOP I/O.

Table 4–7. I/O Analog In/Out Monitor Screen Items

ITEM	DESCRIPTION
I/O[#]	This item is the signal number of the analog input or output.
SIM	This item indicates whether the analog input or output is simulated or unsimulated. To change this value, 1. Move the cursor to the appropriate line. 2. Move the cursor to the SIM column. 3. Press F4, SIMULATE, or press F5, UNSIM.

Table 4–7. I/O Analog In/Out Monitor Screen Items (Cont'd)

VALUE	This item indicates whether the status of the signal is displayed in decimal or hexadecimal format. To change the value, 1. Move the cursor to the appropriate line. 2. Move the cursor to the VALUE column. 3. Press F4, FORMAT, to change the display.
COMMENT	This item is a comment field into which you can type information pertaining to the signal. To type a comment, 1. Move the cursor to the appropriate line. 2. Press F4, DETAIL. If you do not see DETAIL, press NEXT. 3. Press ENTER. 4. Type the comment. 5. Press ENTER. 6. Press PREV to return to the I/O Analog In/Out Monitor screen.

Table 4–8. I/O Analog In/Out Configure Screen Items

ITEM	DESCRIPTION
I/O #	This item is the signal number of the analog input or output.
RACK	This item is the physical location on which the I/O board or module is mounted. To change the rack value, 1. Move the cursor to the line you want to change. 2. Move the cursor to the RACK column. 3. Type the new value. 4. Press ENTER.

Table 4–8. I/O Analog In/Out Configure Screen Items (Cont'd)

SLOT	This item is the space on the rack where the I/O module is connected. To change the slot value, 1. Move the cursor to the line you want to change. 2. Move the cursor to the SLOT column. 3. Type the new value. 4. Press ENTER.
CHANNEL	This item is the physical position of the port on the process I/O board or terminal number on the modular I/O module. To change the channel value, 1. Move the cursor to the line you want to change. 2. Move the cursor to the CHANNEL column. 3. Type the new value. 4. Press ENTER.

Table 4–9. I/O Digital In/Out Monitor Screen Items

ITEM	DESCRIPTION
I/O[#]	This item is the signal number of the digital input or output.
SIM	This item indicates whether the digital input or output is simulated or unsimulated. To change this value, 1. Move the cursor to the appropriate line. 2. Move the cursor to the SIM column. 3. Press F4, SIMULATE, or press F5, UNSIM.

Table 4–9. I/O Digital In/Out Monitor Screen Items (Cont'd)

STATUS	This item indicates the current status of the selected signal. To change this value, 1. Move the cursor to the appropriate line. 2. Move the cursor to the STATUS column. 3. Press F4, ON, or press F5, OFF.
COMMENT	This item is a comment field into which you can type information pertaining to the signal. To type a comment, 1. Move the cursor to the appropriate line. 2. Press F4, DETAIL. If you do not see DETAIL, press NEXT. 3. Move the cursor to the Comment field. 4. Press ENTER. 5. Type the comment. 6. Press ENTER. 7. Press PREV to return to the I/O Digital In/Out Monitor screen.

Table 4–10. I/O Digital In/Out Configure Screen Items

ITEM	DESCRIPTION
I/O #	This item is the number of each signal range.
RANGE	This item is a range, or sequence, of signals. To change the range, 1. Move the cursor to the line you want to change. 2. Move the cursor to either the start range or the end range value. 3. Type the new value. 4. Press ENTER.

Table 4–10. I/O Digital In/Out Configure Screen Items (Cont'd)

RACK	This item is the physical location on which the I/O board or module is mounted. To change the rack value, 1. Move the cursor to the line you want to change. 2. Move the cursor to the RACK column. 3. Type the new value. 4. Press ENTER.
SLOT	This item is the space on the rack where the I/O module is connected. To change the slot value, 1. Move the cursor to the line you want to change. 2. Move the cursor to the SLOT column. 3. Type the new value. 4. Press ENTER.
START	This item is the port number within the sequence of ports on the board or module. To change the start value, 1. Move the cursor to the line you want to change. 2. Move the cursor to the START column. 3. Type the new value. 4. Press ENTER.
STAT	This item indicates the status of the digital signal. Status can be one of the following: • ACTIV - The assignment was valid at start up and is in effect. • UNASG - The assignment has not been made. • INVAL - The assignment is invalid based on the digital I/O hardware present when the controller was turned on. • PEND - The assignment is valid but was made since the last time the controller was turned on and is therefore not active. You must turn off the controller and then turn it back on to make the change take effect.

Table 4–11. I/O Digital In/Out DETAIL Screen Items

ITEM	DESCRIPTION
I/O [#]	This item is the signal number of the digital input or output.
Comment	This item is a comment field into which you can enter information pertaining to the signal. To type a comment, 1. Move the cursor to the Comment field. 2. Press ENTER. 3. Type the comment. 4. Press ENTER. 5. Press PREV to return to the I/O Analog In/Out Monitor screen.
Polarity	This item is indicates whether signals are of NORMAL or INVERSE polarity. To set the polarity, 1. Move the cursor to the Polarity field. 2. Press F4, INVERSE, or F5, NORMAL.
Complementary	This item indicates whether signals are controlled as complementary pairs. To set complementary pairs, 1. Move the cursor to the Complementary field. 2. Press F4, TRUE, or F5, FALSE.

Table 4–12. I/O Group In/Out Monitor Screen Items

ITEM	DESCRIPTION
I/O[#]	This item is the signal number of the group input or output.
SIM	This item indicates whether the group input or output is simulated or unsimulated. To change this value, 1. Move the cursor to the appropriate line. 2. Move the cursor to the SIM column. 3. Press F4, SIMULATE, or press F5, UNSIM.

Table 4–12. I/O Group In/Out Monitor Screen Items (Cont'd)

VALUE	This item indicates whether the status of the signal is displayed in decimal or hexadecimal format. To change the value, 1. Move the cursor to the appropriate line. 2. Move the cursor to the VALUE column. 3. Press F4, FORMAT, to change the display value.
COMMENT	This item is a comment field into which you can type information pertaining to the signal. To type a comment, 1. Move the cursor to the appropriate line. 2. Press F4, DETAIL. If you do not see DETAIL, press NEXT. 3. Press ENTER. 4. Type the comment. 5. Press ENTER. 6. Press PREV to return to the I/O Analog In/Out Monitor screen.

Table 4–13. I/O Group In/Out Configure Screen Items

ITEM	DESCRIPTION
I/O #	This item is the signal number of the group input or output.
RACK	This item is the physical location on which the I/O board or module is mounted. To change the rack value, 1. Move the cursor to the line you want to change. 2. Move the cursor to the RACK column. 3. Type the new value. 4. Press ENTER.

Table 4–13. I/O Group In/Out Configure Screen Items (Cont'd)

SLOT	This item is the space on the rack where the I/O module is connected. To change the slot value, 1. Move the cursor to the line you want to change. 2. Move the cursor to the SLOT column. 3. Type the new value. 4. Press ENTER.
START PT	This item is the port number within the sequence of ports on the board or module.
NUM PTS	This item is the number of points.



Warning

The software preconfigures some or all of the I/O. Make certain that you do not assign or try to simulate I/O that already has been preconfigured; otherwise, you could injure personnel or damage equipment.

Procedure 4-1 Setting Up I/O

Note Analog and digital I/O is configured automatically by the system at Cold start when the system variable $IO_AUTO_CFG is set to TRUE. Use this procedure only if you want to change the configuration.

Steps

1. Press MENUS.
2. Select I/O.
3. Press F1, [TYPE].
4. Select the kind of I/O you want to set up: analog, digital, or group. You will see either an input or output screen. Go to the appropriate step:
   - For analog I/O, go to Step 5 .
   - For digital I/O, go to Step 10 .
   - For group I/O, go to Step 16 .
   - For UOP I/O, go to Step 21 .

   Note If you are using SpotTool+ and if you have previously defined a signal in the Cell, Weld, or Equipment I/O screens, the name of the signal as defined on the Cell, Weld, or Equipment I/O screens will be displayed in the comment field of the corresponding I/O signal on these screens.

5. For **analog I/O**, you will see a screen similar to the following.

```
I/O Analog Out
      #    SIM    VALUE
 AO [   1]  U        0   [              ]
 AO [   2]  U        0   [              ]
 AO [   3]  *        *   [              ]
 AO [   4]  *        *   [              ]
 AO [   5]  *        *   [              ]
 AO [   6]  *        *   [              ]
 AO [   7]  *        *   [              ]
 AO [   8]  *        *   [              ]
 AO [   9]  *        *   [              ]
 AO [  10]  *        *   [              ]
```

Note By default, analog inputs and outputs can have up to 50 I/O points that can be configured. These configurations do not affect the amount of digital I/O that can be configured.

To change between the display of the input and output screens, press F3, IN/OUT.

To move quickly through the information, press and hold the SHIFT key and press the down or up arrow keys.

6. Move the cursor to the I/O signal you want to configure.

7. Press F2, CONFIG. You will see a screen similar to the following.

```
I/O Analog Out
 AO #  RACK   SLOT  CHANNEL
 1       1      1       1
 2       0      1       2
 3       0      0       0
 4       0      0       0
 5       0      0       0
 6       0      0       0
 7       0      0       0
 8       0      0       0
 9       0      0       0
```

8. Configure the I/O:

a. Move the cursor to RACK, type the value, and press ENTER.

b. Move the cursor to SLOT, type the value, and press ENTER.

c. Move the cursor to CHANNEL, type the value, and press ENTER.

Note After you configure the I/O, be sure to verify that the assignment is valid for the present physical I/O configuration. Go to Step 26 to verify the I/O assignments.

9. To add a comment:

a. If you are not already at the MONITOR screen, press F2, MONITOR.

b. Move the cursor to the input or output for which you want to set a comment.

c. Press NEXT, >, and then press F4, DETAIL. You will see a screen similar to the following.

```
I/o Analog Out
Port Detail
 Analog Output:                [  1]
1   Comment:  [port-1 comment           ]
```

d. Press ENTER, press the appropriate function keys to type the comment, and press ENTER.

e. **To set comments** for the previous signal, press F2, PRV-PT, and for the next signal, press F3, NXT-PT.

10. For **digital I/O**, you will see a screen similar to the following.

```
I/O Digital In
       #   SIM   STATUS
 DI [   1]   U    OFF  [                ]
 DI [   2]   U    OFF  [                ]
 DI [   3]   U    OFF  [                ]
 DI [   4]   U    OFF  [                ]
 DI [   5]   U    OFF  [                ]
 DI [   6]   U    OFF  [                ]
 DI [   7]   U    OFF  [                ]
 DI [   8]   U    OFF  [                ]
 DI [   9]   U    OFF  [                ]
 DI [  10]   U    OFF  [                ]
Sorted by number
```

Note By default, digital inputs and outputs can have up to 512 I/O points that can be configured.

To change between the display of the input and output screens, press F3, IN/OUT.

To move quickly through the information, press and hold the SHIFT key and press the down or up arrow keys.

Note By default, signals are displayed in signal number order. Alternatively, they can be displayed in alphabetical order of comment.

11. To sort the signals, do the following:

- To sort by number, press NEXT, >, and then F2, NUM_SRT.
- To sort by comment, press NEXT, >, and then F3, CMT_SRT.

12. To configure digital signals, move the cursor to the signals you want to configure and press F2, CONFIG. You will see a screen similar to the following. Refer to Table 4–14 .

```
I/O Digital Out
 #    RANGE        RACK  SLOT START STAT.
 1 DO [   1-  16]     1    1    1  INVAL
 2 DO [  17-  19]     1    2    6  ACTIV
 3 DO [  20-  24]     0    0    0  UNASG
 4 DO [  25-  28]     1    2    1  ACTIV
 5 DO [  29- 100]     0    0    0  UNASG
 6 DO [ 101- 356]    16    1    1  PEND
 7 DO [ 357- 390]     0    0    0  UNASG
 8 DO [ 391- 398]     1    3    0  INVAL
 9 DO [ 399- 400]     0    0    0  UNASG
Power OFF, then ON to enable changes
```

Table 4–14. CONFIG Screen I/O Status

Status	Description
ACTIV (Active)	The assignment is valid and currently active.
INVAL (Invalid)	The assignment is invalid, based on the digital I/O hardware present when the controller was turned on.
PEND (Pending)	The assignment is valid but was made since the last time the controller was turned on and is therefore not active. You must turn off the controller and then turn it back on to make the change take effect.
UNASG (Unassigned)	An assignment has not been made.

a. Move the cursor to RANGE and set the range. Refer to Table 4–15 for restrictions on setting the range.

Table 4–15. Range Setting Limitations

DI/DO[FFF, LLL]
The first port number (FFF) must be greater than the last port number (LLL) of the preceding line if it is not UNASG.
The first port number (FFF) must be greater than or equal to the FFF of the preceding line if it is UNASG.
The first port number (FFF) must be greater than zero if there is no preceding line.
The last port number (LLL) must be less than the first port number (FFF) in the next line if it is not UNASG.
The last port number (LLL) must be less than or equal to the LLL of the next line, if it is UNASG.
The last port number (LLL) must not be greater than the current value if there is no next line.

b. Move the cursor to RACK, type the value, and press ENTER.

c. Move the cursor to SLOT, type the value, and press ENTER.

d. Move the cursor to START, type the value, and press ENTER.

e. **To delete the assignment for a range of ports**, either set the RACK, SLOT, and START values to zero, or press F4, DELETE, and then confirm the deletion. After a deletion, the range of ports will be merged with the UNASG lines immediately above or below it automatically.

Note After you configure the I/O, be sure to verify that the assignment is valid for the present physical I/O configuration. Go to Step 26 to verify the I/O assignments.

13. To set polarity or complementary pairs for digital signal from the DETAIL screen,

a. If you are not already at the MONITOR screen, press F2, MONITOR.

b. Move the cursor to the input or output port you want to configure.

c. Press NEXT, >, and then press F4, DETAIL. You will see a screen similar to the following.

```
I/O Digital Out
Port Detail

 Digital Output:                [  1]
  1   Comment:  [port-1 comment           ]
  2          Polarity:  INVERSE
  3   Complementary[  1  , 2]:  TRUE
```

d. **To set polarity**, move the cursor to Polarity and press F4 for INVERSE, or F5 for NORMAL.

Note Complementary will be displayed only for odd-numbered digital output signals.

e. **To set complementary pairs**, move the cursor to Complementary and press F4 for TRUE, or F5 for FALSE.

14. **To add a comment for digital I/O signals**,

a. From the Port Detail screen for the signals you want, move the cursor to Comment and press ENTER.

b. Press the appropriate function keys to type the comment and press ENTER.

15. **To set comments, complementary pairs, or polarity** for the previous signal, press F2, PRV-PT, and for the next signal, press F3, NXT-PT.

16. For **group I/O**, you will see a screen similar to the following.

```
I/O Group Out
     #       SIM     VALUE
 GO [   1]    *          0   [               ]
 GO [   2]    *          0   [               ]
 GO [   3]    *          0   [               ]
 GO [   4]    *          0   [               ]
 GO [   5]    *          0   [               ]
 GO [   6]    *          0   [               ]
 GO [   7]    *          0   [               ]
 GO [   8]    *          0   [               ]
 GO [   9]    *          0   [               ]
 GO [  10]    *          0   [               ]
```

To change between the display of the input and output screens, press F3, IN/OUT.

To move quickly through the information, press and hold the SHIFT key and press the down or up arrow keys.

17. Move the cursor to the I/O signal you want to configure.

18. Press F2, CONFIG. You will see a screen similar to the following.

```
I/O Group Out
  GO #  RACK   SLOT   START PT  NUM PTS
   1       0      0         0       0
   2       0      0         0       0
   3       0      0         0       0
   4       0      0         0       0
   5       0      0         0       0
   6       0      0         0       0
   7       0      0         0       0
   8       0      0         0       0
   9       0      0         0       0
```

19. Configure the I/O:

a. Move the cursor to RACK, type the value, and press ENTER.

b. Move the cursor to SLOT, type the value, and press ENTER.

c. Move the cursor to START PT, type the value, and press ENTER. The starting point can be any number up to and including 999.

d. Move the cursor to NUM PTS, type the value, and press ENTER. The number of points can be from 1 up to and including 16.

Note After you configure the I/O, be sure to verify that the assignment is valid for the present physical I/O configuration. Go to Step 26 to verify the I/O assignments.

20. To add a comment:

a. If you are not already at the MONITOR screen, press F2, MONITOR.

b. Move the cursor to the input or output group for which you want to set a comment.

c. Press NEXT, >, and then press F4, DETAIL. You will see a screen similar to the following.

```
I/O Group Out
Port Detail

 Group Output:                  [  1]
  1   Comment:  [port-1 comment           ]
```

d. Press ENTER, press the appropriate function keys to type the comment, and press ENTER.

e. **To set comments** for the previous signal, press F2, PRV-PT, and for the next signal, press F3, NXT-PT.

21. For **UOP I/O**, you will see a screen similar to the following.

```
I/O UOP Out
      #       STATUS
UO [  1]     OFF  [Cmd enabled       ]
UO [  2]     OFF  [System ready      ]
UO [  3]     OFF  [Prg running       ]
UO [  4]     OFF  [Prg paused        ]
UO [  5]     OFF  [Motion held       ]
UO [  6]     OFF  [Fault             ]
UO [  7]     OFF  [At Perch          ]
UO [  8]     OFF  [TP enabled        ]
UO [  9]     OFF  [Batt alarm        ]
UO [ 10]     OFF  [Busy              ]
UO [ 11]     OFF  [ACK1/SNO1         ]
UO [ 12]     OFF  [ACK2/SNO2         ]
UO [ 13]     OFF  [ACK3/SNO3         ]
UO [ 14]     OFF  [ACK4/SNO4         ]
UO [ 15]     OFF  [SNO5              ]
UO [ 16]     OFF  [SNO6              ]
UO [ 17]     OFF  [SNO7              ]
UO [ 18]     OFF  [SNO8              ]
UO [ 19]     OFF  [SNACK             ]
UO [ 20]     OFF  [Reserved          ]
```

To change between the display of the input and output screens, press F3, IN/OUT.

To move quickly through the information, press and hold the SHIFT key and press the down or up arrow keys.

Note By default, signals are displayed in signal number order. Alternatively, they can be displayed in alphabetical order of comment.

22. **To sort the signals**, do the following:

    - To sort by number, press NEXT, >, and then F2, NUM_SRT.
    - To sort by comment, press NEXT, >, and then F3, CMT_SRT.

23. **To configure UOP signals**, move the cursor to the signals you want to configure and press F2, CONFIG. You will see a screen similar to the following. Refer to Table 4–14 .

```
I/O UOP Out
 #      RANGE       RACK  SLOT START STAT
 1 UO [  1-  8]       0     1     1  ACTIV
 2 UO [  9- 16]       0     1     9  ACTIV
 3 UO [ 17- 20]       0     1    17  ACTIV
```

24. Configure the I/O:

a. Move the cursor to RANGE and set the range. Refer to Table 4–15 for restrictions on setting the range.

b. Move the cursor to RACK, type the value, and press ENTER.

c. Move the cursor to SLOT, type the value, and press ENTER.

d. Move the cursor to START, type the value, and press ENTER.

e. **To delete an assignment**, either set the RACK, SLOT, and START values to zero, or press F4, DELETE, and then confirm the deletion. After a deletion, the range of ports will be merged with the UNASG lines immediately above or below it automatically.

Note After you configure the I/O, be sure to verify that the assignment is valid for the present physical I/O configuration. Go to Step 26 to verify the I/O assignments.

25. **To add a comment**,

a. If you are not already at the MONITOR screen, press F2, MONITOR.

b. Move the cursor to the input or output group for which you want to set a comment.

c. Press NEXT, >, and then press F4, DETAIL. You will see a screen similar to the following.

```
I/O UOP Out
Port Detail

 UOP Output:                    [  1]
  1   Comment:  [port-1 comment           ]
```

d. Move the cursor to Comment, press ENTER, press the appropriate function keys to type the comment, and press ENTER.

e. **To set comments** for the previous signal, press F2, PRV-PT, and for the next signal, press F3, NXT-PT.

26. **To determine if the assignment is valid (verify it),** press NEXT, >, and then press F2, VERIFY.

- If the assignment is valid, the message, "Port assignment is valid," is displayed.
- If the assignment is not valid, the message, "Port assignment is invalid," is displayed.



Caution

When all I/O is configured, save the information to the default device so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

27. To save the information (when all I/O is configured):

a. Press MENUS.

b. Select FILE.

c. Press F1, [TYPE].

d. Select File.

e. Press F5, [UTIL].

f. Select Set Device.

g. Move the cursor to the device you want and press ENTER.

h. Press MENUS.

i. Select I/O.

j. Press FCTN.

k. Select SAVE. The file will be saved to the DIOCFGSV.IO file on the default device.



Warning

You must turn off the controller and then turn it back on to use the new information; otherwise, you could injure personnel or damage equipment.

28. When you are finished configuring I/O, turn off the controller. Then, turn on the controller so it can use the new information.

4.2.9 Custom I/O

Custom I/O allows you to display a customized I/O screen with up to 20 entries for any kind of I/O. It might or might not be displayed, depending on your configuration.

Table 4–16 shows the Custom I/O Setup items. Use Procedure 4-2 to set up Custom I/O.

Table 4–16. Custom I/O Setup Items

Custom Item Type	Custom Item Index	Description
DI, DO, RO, RI, GO, GI, UO, UI, AO, AI, SO, SI	Any valid I/O Index	Each kind of I/O can be configured in the Custom I/O screen. All comments, simulation status, and state of the I/O will mimic the System I/O screens. All simulation and states can be changed in the Custom I/O screen. I/O comments are inserted based on the currently defined I/O comments and cannot be changed from the Custom I/O menu. I/O comments must be changed in the appropriate I/O screen.

Procedure 4-2 Setting Up Custom I/O

1. Press MENUS.
2. Select I/O.
3. Press F1, [TYPE] and select Custom. If the screen has not been configured, you will see a blank screen that you can configure as desired. The following screens are examples of custom I/O screens.

```
CUSTOM I/O
       DESCRIPTION     TYPE #   S STAT
Prg running            UO[    3]U OFF
FAULT                  UO[    6]U OFF
Fault reset            UI[    5]U OFF
Style Req. / Echo      GO[    1]U    0
Style Select           GI[    1]U    0
Style Ack              DO[    1]U OFF
Lv Pounce(ok to)       DI[    1]U OFF
Fault Led              SO[    3]U OFF
Hold                   SI[    3]U OFF
```

```
CUSTOM I/O           JOINT 100 %
                                  1/4
       DESCRIPTION       TYPE #   S STAT
Box orientation          DO[   11]U OFF
Vac 1 ON                 RO[    1]U OFF
Box Request DO           DO[   10]U OFF
Part pres case 1         RI[    1]U OFF
```

4. To delete an item on the Custom I/O screen, move the cursor to the line that you want to delete; press NEXT, then press F2, [DELETE].
5. To insert an item on the Custom I/O screen, press NEXT, then press F1, [INSERT]. This will display the configuration screen to configure the kind and index of I/O that you want on your Custom I/O Screen. The following screen is an example:

```
CUSTOM I/O CONFIG
      I/O Signal Details

      Signal name: Fault reset
      I/O type/no: UI[    5]
```

6. To configure an item on the Custom I/O screen, move the cursor to the line you want to configure, and press F3, [CONFIG], to display the configuration screen to configure the kind and index of I/O you want on your Custom I/O Screen.
7. Move the cursor to the index, and type the index number.
8. Move the cursor to the kind of I/O, and press F4, [CHOICE], to select the kind of I/O.
9. Press F5, [VERIFY]. This will indicate whether the I/O is valid and will retrieve the I/O comment associated with the I/O point.
10. When you have completed the configuration, press PREV. This will insert your I/O point below the cursor on the main page.

    After you have set up Custom I/O you will be able to set I/O simulation and states by moving the cursor with the arrow keys and using F4, [SIM/ON], or F5, [UNSIM/OFF] keys.

4.3 DISTRIBUTED (MODEL B) I/O SETUP

4.3.1 Overview

This section contains information on how to set up I/O for distributed (Model B) I/O modules.

Refer to the *FANUC Robotics Controller Maintenance Manual* for more detailed information about the Model B I/O hardware.

You must do the following to use distributed I/O:

1. Configure the distributed I/O DIP switches. Refer to Section 4.3.2 .

2. Set up each basic and extension digital I/O module. Refer to Section 4.3.3 .
3. Set up user I/O signals. Refer to Section 4.3.4 .

The following example describes a typical distributed I/O setup.

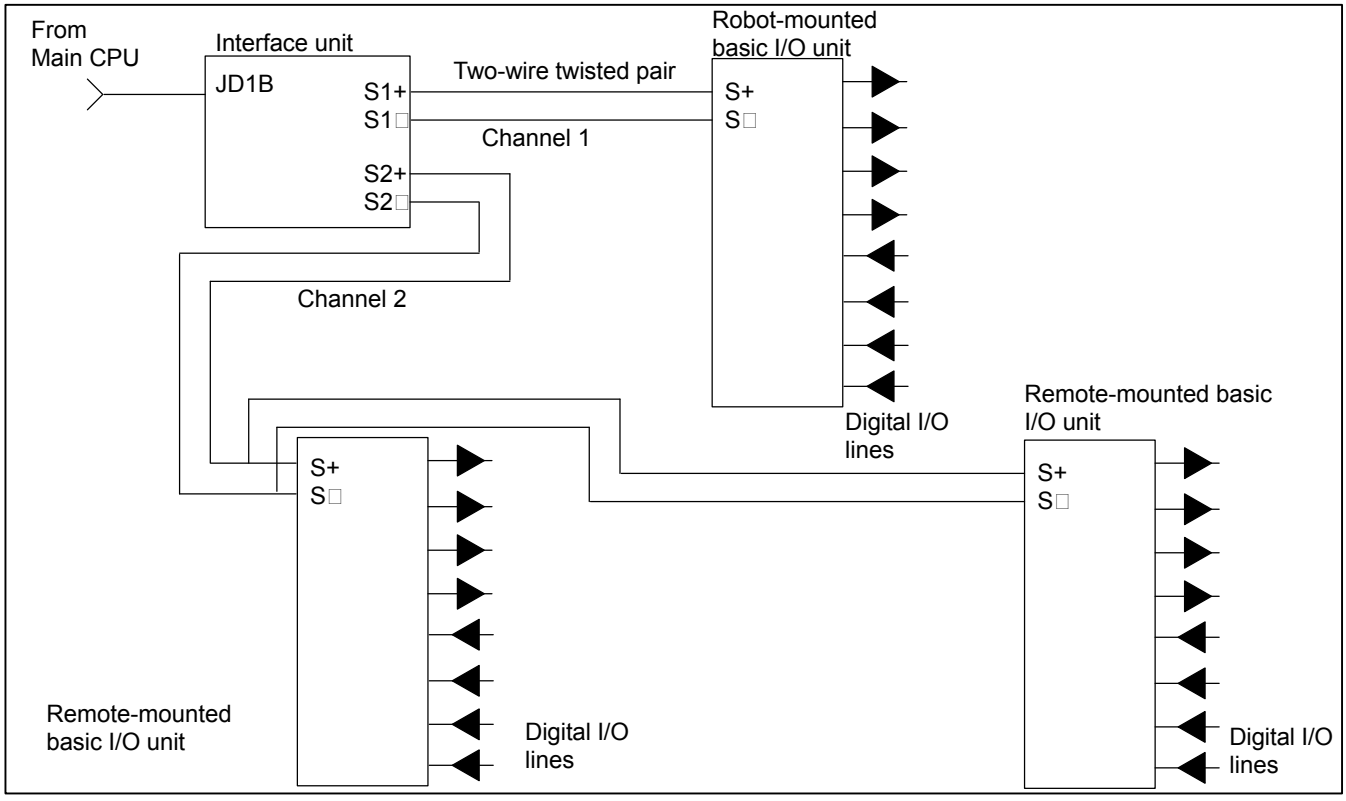
Distributed (Model B) I/O Example Setup

The examples in this section assume that you are setting up an installation with the distributed I/O interface unit mounted in the robot controller and three basic digital I/O units which can be mounted in various remote locations, such as:

- Robot arm (basic unit 1, connected to channel 1)
- Inside the operator box (basic unit 2, connected to channel 2)
- Inside a peripheral device (basic unit 3, connected to channel 2)

See Figure 4–2 for an illustration of this example setup.

Figure 4–2. Example Distributed I/O Setup Block Diagram



4.3.2 Setting the DIP Switches

You must set the following distributed I/O DIP switches:

- On the interface unit, refer to Procedure 4-3
- On each basic digital I/O unit, refer to Procedure 4-4

Procedure 4-3 Setting the DIP Switches on the Interface Unit

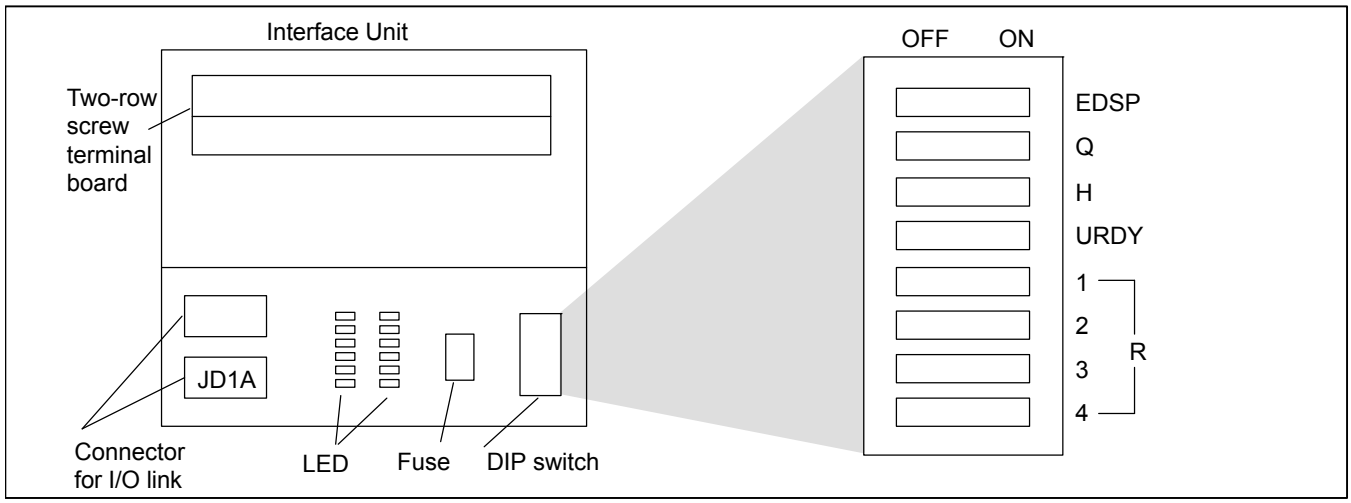
Conditions

- The I/O modules are installed and wired properly.

Steps

1. Locate the DIP switches on the interface unit. An eight-switch DIP switch package is mounted at the lower right corner of the interface module. See Figure 4–3 .

Figure 4–3. Interface Unit DIP Switches



2. Set the EDSP switch to the ON position.
3. Set the communication speed using switches Q and H.

   The I/O system can communicate at the following data rates: 1.2 Mbps, 600 Kbps, 300 Kbps.

   Normally, you will use 1.2 Mbps (1.2 million bits per second). However, when the total length of the communication lines exceeds 100 meters, a slower speed must be used.

   Use the information in Table 4–17 to set switches Q and H.

Table 4–17. Communication Speed Settings for Switches Q and H

Q	H	Communication Speed
OFF	OFF	1.2 Mbps
OFF	ON	600 Kbps
ON	OFF	300 Kbps

4. Write down the positions of switches Q and H. You will need this information when you configure the basic digital I/O units in Procedure 4-4 .
5. Set URDY to the OFF position.
6. Set the termination resistors, represented by switches R1 through R4.
   a. Examine the terminals for channel 1 (S1+ and S1-) and set switch R1 as follows:
      - **If one twisted-pair cable** is attached to these terminals, set the switch to ON.
      - **If more than one twisted-pair cable** is attached to these terminals, set the switch to OFF.
   b. Examine the terminals for channel 2 (S2+ and S2-) and set switch R2 the same way you set switch R1 in Step 6a
   c. Examine the terminals for channel 3 (S3+ and S3-) and set switch R3 the same way you set switch R1 in Step 6a .
   d. Examine the terminals for channel 4 (S4+ and S4-) and set switch R4 the same way you set switch R1 in Step 6a .

Procedure 4-4 Setting the DIP Switches on a Basic Digital I/O Unit

Note You must set the DIP switches for each basic digital I/O unit in your system.

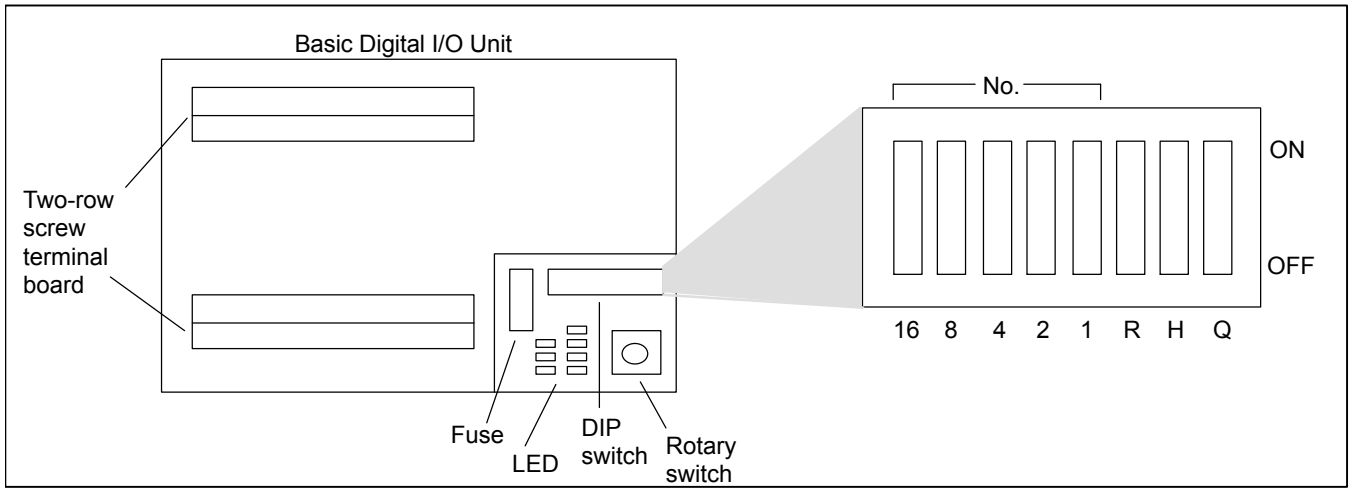
Conditions

- The DIP switches on the interface unit have been set properly. (Procedure 4-3)

Steps

1. Locate the DIP switches on the basic digital I/O unit. An eight-switch DIP switch package is mounted at the lower right corner of each basic digital I/O module. See Figure 4–4 .

Figure 4–4. Basic Digital I/O Module DIP Switches



2. Set the unit number using switches 16, 8, 4, 2, and 1. These switches are set to show the number of the unit in binary notation. The unit number is the slot number supplied in the configuration screens. Refer to Table 4–18 .

Table 4–18. Unit Number Settings of Switches 16, 8, 4, 2, and 1

Unit Number	Settings				
	16	**8**	**4**	**2**	**1**
1	OFF	OFF	OFF	OFF	ON
2	OFF	OFF	OFF	ON	OFF
3	OFF	OFF	OFF	ON	ON
4	OFF	OFF	ON	OFF	OFF
5	OFF	OFF	ON	OFF	ON
6	OFF	OFF	ON	ON	OFF
7	OFF	OFF	ON	ON	ON

3. Set the termination resistor, represented by switch R. Examine the terminals for S+ and S- and set switch R as follows:

- **If one twisted-pair cable** is attached to these terminals, set switch R to ON.
- **If more than one twisted-pair cable** is attached to these terminals, set switch R to OFF.
- **If no wires** are attached to these terminals, R can be set to either ON or OFF.

Note The positions of switches Q and H on the basic digital I/O module are reversed from the positions on the interface module. Be sure to set them properly.

4. Set the communication speed using switches Q and H. Use the same switch settings you used for the interface module in Procedure 4-3 , Step 3 .

4.3.3 Setting Up the Basic Digital I/O Units

You must set up each basic digital I/O unit you use. You do this from the I/O Link screen. Use Procedure 4-5 to set up Model B I/O basic digital I/O units.

Procedure 4-5 Setting Up Model B Basic Digital I/O Units

Conditions

- An I/O unit model B interface is connected.

Steps

1. Press MENUS.
2. Select I/O.
3. Press F1, [TYPE].
4. Select I/O Link. You will see a screen similar to the following.

```
I/O Link Device
   Device Name      Comment     RackSlot
  1 PrcI/O AA [                ] 0   1
  2 Model B   [                ] 1   0
  3 Model A   [                ] 2   0
  4 Model A   [                ] 3   0
```

5. Move the cursor to Model B and press F3, DETAIL. You will see a screen similar to the following.

```
I/O Link Device
 Model B                  Rack 1
Slot Base      Exp.       Comment
  1 ******* ******* [                ]
  2 ******* ******* [                ]
  3 ******* ******* [                ]
 30 ******* ******* [                ]
```

6. Select the **base unit product name** as follows:

a. Move the cursor to Base.

b. Press F4, [CHOICE]. You will see a screen similar to the following.

```
 1 *******          5 BOA12A1
 2 BID16A1          6 BIA16P1
 3 BOD16A1          7 BMD88Q1
 4 BMD88A1          8
```

c. Select the appropriate base unit name.

Note "*******" choice indicates no unit.

You will see a screen similar to the following.

```
I/O Link Device
 Model B                  Rack 1
Slot Base      Exp.       Comment
  1 B0D16A1 ******* [                ]
  2 ******* ******* [                ]
  3 ******* ******* [                ]
 30 ******* ******* [                ]
```

7. Specify the appropriate **expansion units** (which have a "P" in the Product Name) as follows:

a. Move the cursor to Exp.

b. Press F4, [CHOICE]. If the base column is not filled in, the message "No base unit" is displayed.

8. If you want to enter a **comment** , move the cursor to Comment and press ENTER. Press the appropriate function keys to type the comment and press ENTER.



Caution

When all I/O is configured, save the information to the default device so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

9. **To save the I/O settings** ,

a. Press MENUS.

b. Select FILE.

c. Press F1, [TYPE].

d. Select File.

e. Press F5, [UTIL].

f. Select Set Device.

g. Move the cursor to the device you want and press ENTER.

h. Press MENUS.

i. Select I/O.

j. Press FCTN.

k. Select SAVE. The file will be saved to the DIOCFGSV.IO file on the default device.

> **Caution**
>
> CLR_ASG clears assignments of all ports on all units, including process I/O, model A, model B, and PLC devices. The next time the controller is turned on, ports for these devices will be given default assignments.

10. If you want to clear all assignments ,

a. Press F5, CLR_ASG.

The following message is displayed.

```
Clear all assignments?
```

b. Press the appropriate key:

- Press F4, YES to clear all I/O assignments.
- Press F5, NO not to clear all I/O assignments.

11. After you set up detail information, you must turn OFF the controller. Then turn it back ON for the new information to take effect.

4.3.4 Setting Up User I/O

After you have set up the DIP switches and have set up each basic I/O unit, you can set up user I/O. This defines the I/O you will use in your system. You can set up the following kinds of user I/O:

- **Digital** - DI[n] and DO[n]

- **Group** - GI[n] and GO[n]
- **UOP** - UI[n] and UO[n]
- **PLC** - PI[n] and PO[n]

Refer to Section 4.2 for information on how to set up I/O.

4.4 ROBOT I/O SETUP

Robot I/O consists of the input and output signals between the controller and the robot. These signals are sent to the EE (End Effector) connector located on the robot. You can change the status of outputs on the robot I/O screen.

The number of robot input and output signals (RI and RO) varies depending on the number of axes in your system.

You can do the following with robot signals:

- Force robot output signals ON or OFF
- Configure complementary RO signals
- Configure the polarity of RI and RO signals
- Simulate I/O— refer to Section 4.8.3
- Set comments
- Display robot input and output signals

Complementary Output Signals

You can configure robot output signals to be controlled independently or in complementary pairs. If an output signal is controlled independently, a command to turn that output signal ON or OFF controls only that output signal. If an output signal is controlled in a complementary pair, a command to turn that signal ON will also turn its pair OFF. A command to turn the signal OFF will also turn its pair ON.

Polarity

You can configure robot input/output signals with normal polarity (active ON) or inverse polarity (active OFF).

You can set up and configure robot I/O from the I/O Robot In/Out screen.

The I/O Robot In screen displays the items in Table 4–19 .

Table 4–19. I/O Robot In/Out Screen Items

ITEM	DESCRIPTION
I/O [#]	This item is the signal number of the robot input or output.
STATUS	This item indicates the current status of the selected signal. To change this value, 1. Move the cursor to the appropriate line. 2. Move the cursor to the STATUS column. 3. Press F4, ON, or press F5, OFF.
COMMENT	This item is a comment field into which you can type information pertaining to the signal. To type a comment, 1. Move the cursor to the appropriate line. 2. Press F4, DETAIL. If you do not see DETAIL, press NEXT. 3. Move the cursor to the Comment field. 4. Press ENTER. 5. Type the comment. 6. Press ENTER. 7. Press PREV to return to the I/O Robot In/Out screen.

You can type comments, and set polarity and complementary values from the I/O Robot In/Out DETAIL screen.

Table 4–20. I/O Robot In/Out DETAIL Screen Items

ITEM	DESCRIPTION
I/O [#]	This item is the signal number of the robot input or output.
Comment	This item is a comment field into which you can type information pertaining to the signal. To type a comment, 1. Move the cursor to the Comment field. 2. Press ENTER. 3. Type the comment. 4. Press ENTER. 5. Press PREV to return to the I/O Robot In/Out screen.

Table 4–20. I/O Robot In/Out DETAIL Screen Items (Cont'd)

Polarity	This item is indicates whether signals are of NORMAL or INVERSE polarity. To set the polarity, 1. Move the cursor to the Polarity field. 2. Press F4, INVERSE, or F5, NORMAL.
Complementary	This item indicates whether signals are controlled as complementary pairs. To set complementary pairs, 1. Move the cursor to the Complementary field. 2. Press F4, TRUE, or F5, FALSE.

Use Procedure 4-6 to set up robot I/O.

Procedure 4-6 Setting Up Robot I/O

Steps

1. Press MENUS.
2. Select I/O.
3. Press F1, [TYPE].
4. Select Robot. You will see either the robot input or robot output screens. See the following screen for an example.

   Note If you have previously defined a signal in the Equipment I/O screen, the name of the signal as defined on the Equipment I/O screen will be displayed in the comment field of the corresponding I/O signal on these screens.

```
I/O Robot Out
      #    SIM  STATUS
  RO[   1]  U    OFF    [                ]
  RO[   2]  U    OFF    [                ]
  RO[   3]  U    OFF    [                ]
  RO[   4]  U    OFF    [                ]
  RO[   5]  U    OFF    [                ]
  RO[   6]  U    OFF    [                ]
  RO[   7]  U    OFF    [                ]
  RO[   8]  U    OFF    [                ]
```

To change between the display of the input and output screens, press F3, IN/OUT.

To move quickly through the information, press and hold the SHIFT key and press the down or up arrow keys.

5. **To force an output signal,** move the cursor to the output you want to change and press

   - F4, ON, to turn on an output signal.
   - F5, OFF, to turn off an output signal.

6. **To set port comments, polarity, or complementary pairs** , press NEXT, >, and then press F4, DETAIL. You will see a screen similar to the following.

```
I/O Robot Out
Port Detail

 Robot Output:                [  1]
  1   Comment:  [port-1 comment           ]
  2         Polarity:  INVERSE
  3   Complementary[  1   , 2]:  TRUE
```

   - **To add a comment,** move the cursor to comment, press ENTER, press the appropriate function keys to type the comments, and press ENTER.
   - **To set polarity** , move the cursor to Polarity and press F4 for INVERSE, or F5 for NORMAL.

     Note Complementary will be displayed only for odd-numbered robot output signals.

   - **To set complementary pairs** , move the cursor to Complementary and press F4 for TRUE, or F5 for FALSE.
   - **To set comments, complementary pairs, or polarity** for the previous signal, press F2, PRV-PT, and for the next signal, press F3, NXT-PT.



Caution

When all I/O is configured, save the information to the default device so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

7. **To save the information** (when all I/O is configured):
   a. Press MENUS.
   b. Select FILE.
   c. Press F1, [TYPE].
   d. Select File.

e. Press F5, [UTIL].

f. Select Set Device.

g. Move the cursor to the device you want and press ENTER.

h. Press MENUS.

i. Select I/O.

j. Press FCTN.

k. Select SAVE. The file will be saved to the DIOCFGSV.IO file on the default device.



Warning

You must turn off the controller and then turn it back on to use the new information; otherwise, you could injure personnel or damage equipment.

8. When you are finished configuring I/O, turn off the controller. Then, turn on the controller so it can use the new information.

4.5 STANDARD OPERATOR PANEL (SOP) I/O SETUP

The Standard Operator Panel (SOP) I/O screen indicates the status of the standard operator panel signals. SOP input (SI) signals and SOP output (SO) signals correspond to internal controller software Panel Digital Input (PDI) signals and Panel Digital Output (PDO) signals.

Refer to Table 4–21 , Table 4–22 and Table 4–23 .

Table 4–21. I/O SOP In/Out Screen Items

ITEM	DESCRIPTION
I/O [#]	This item is the signal number of the SOP input or output.

Table 4–21. I/O SOP In/Out Screen Items (Cont'd)

STATUS	This item indicates the current status of the selected signal. To change this value, 1. Move the cursor to the appropriate line. 2. Move the cursor to the STATUS column. 3. Press F4, ON, or press F5, OFF.
COMMENT	This item is a comment field into which you can type information pertaining to the signal. To type a comment, 1. Move the cursor to the appropriate line. 2. Press F4, DETAIL. If you do not see DETAIL, press NEXT. 3. Press ENTER. 4. Type the comment. 5. Press ENTER. 6. Press PREV to return to the I/O SOP In/Out Monitor screen.

Table 4–22. Standard Operator Panel Input Signals

SI	PDI	Function	Description
0	1	Not used	This is open for additional PDIs.
1	2	Fault Reset	This input signal is normally turned OFF, indicating that the FAULT RESET button is not being pressed.
2	3	Remote	This input signal is turned OFF, indicating that the controller is not set to remote. This item is not available on the Operator Panel.
3	4	Hold	This input signal is normally turned ON, indicating that the HOLD push button is not being pressed. This item is not available on the Operator Panel.
4	5	User PB#1	This signal is USR PB#1, which is normally turned OFF, indicating that USER PB#1 is not being pressed. This item is not available on the Operator Panel.
5	6	User PB#2	This signal is USR PB#2, which is normally turned OFF, indicating that USER PB#2 is not being pressed. This item is not available on the Operator Panel.
6	7	Cycle Start	This input signal is normally turned OFF, indicating that the CYCLE START push button is not being pressed.
7	8	Not used	This is open for additional PDIs.

Table 4–22. Standard Operator Panel Input Signals (Cont'd)

SI	PDI	Function	Description
8	9	CE/CR SELECT b0	This signal is used for the MODE SELECT switch.
9	10	CE/CR SELECT b1	This signal is used for the MODE SELECT switch.
10-15	11-16	NOT USED	This is open for additional PDIs.

Table 4–23. Standard Operator Panel Output Signals

SO	PDO	Function	Description
0	1	Remote LED	This output signal indicates the controller is set to remote.
1	2	Cycle Start	This output signal indicates the CYCLE START button has been pressed or a program is running.
2	3	Hold	This output signal indicates a HOLD button has been pressed. This item is not available on the Operator Panel.
3	4	Fault LED	This output signal indicates a fault has occurred and has not been reset.
4	5	Batt Alarm	This output signal indicates the voltage in the CPU or r obot batteries are low.
5	6	User LED #1	For SpotTool+, this output signal echoes the Process Complete output signal. For HandlingTool, this signal is not used. For other applications, this output signal is user-definable. This item is not available on the Operator Panel.
6	7	User LED #2	For SpotTool+, this output signal echoes the UOP[ATPERCH] output signal. For HandlingTool, this signal is not used. For other applications, this output signal is user-definable. This item is not available on the Operator Panel.
7	8	TP enabled	This output signal indicates the teach pendant is enabled.
8-15	9-16	not used	This is open for additional PDOs.

Use Procedure 4-7 to display and force SOP I/O.

Procedure 4-7 Displaying and Forcing SOP I/O

Steps

1. Press MENUS.
2. Select I/O.
3. Press F1, [TYPE].
4. Select SOP. If outputs are selected, you will see a screen similar to the following.

```
I/O SOP Out
      #    STATUS
  SO[ 0]    OFF   [Remote LED      ]
  SO[ 1]    OFF   [Cycle start     ]
  SO[ 2]    OFF   [Hold            ]
  SO[ 3]    OFF   [Fault LED       ]
  SO[ 4]    OFF   [Batt alarm      ]
  SO[ 5]    OFF   [USER PB#1       ]
  SO[ 6]    OFF   [USER PB#2       ]
  SO[ 7]    OFF   [TP enabled      ]
  SO[ 8]    OFF   [                ]
  SO[ 9]    OFF   [                ]
```

To change between the display of the input and output screens, press F3, IN/OUT.

To move quickly through the information, press and hold the SHIFT key and press the down or up arrow keys.

Note You can only view the status of input signals. SOP input signals cannot be forced.

5. **To force an output signal**, move the cursor to the output you want to change:
   - **To turn on an output signal**, press F4, ON.
   - **To turn off an output signal**, press F5, OFF.

   Note Most forced signals will be overwritten by values determined by their function.

4.6 SETTING UP I/O LINK DEVICES

4.6.1 Overview

An I/O link device is a device that is connected to the controller through the I/O Link (I/O-LK) connector on the Main CPU PCB. You use the I/O link screen to set up Model B I/O basic digital I/O units and to view the configuration of other I/O link devices.

Setting up I/O link devices can involve the following tasks:

- Setting up basic digital I/O units - Section 4.3.3
- Setting the number of ports - Section 4.6.9

4.6.2 I/O Link Devices

The basic limitation on I/O Link Devices for the controller is 512 bits (64 bytes) of input data and 512 bits (64 bytes) of output data. This is a limitation of the SLC-2 protocol used over the JD1A connector. Each Digital I/O uses 1 bit and each Analog Channel requires 16 bits (2 bytes) of data.

Note The system detects the number of I/O Link Devices when the controller is turned on. These are displayed on the I/O Link Device screen in the I/O Menus.

4.6.3 Process I/O

Each Process I/O module regardless of type requires 16 bytes of I/O. Therefore, the maximum number of Process I/O Modules that can be used in the system is 4 (taking up the entire I/O Link Address). Typically systems use just 1 Process I/O module.

Process I/O is assigned to Rack 0. Multiple boards are assigned via consecutive Slots starting at 1. A system with 4 Process I/O modules would be assigned as Rack 0 Slots 1-4 and take up all 64 bytes of I/O.

Refer to Table 4–24 for Process I/O assignments.

Refer to Table 4–25 for a list of the kinds of Process I/O available on HandlingTool.

Table 4–24. Process I/O Assignments

	Device	Name (Process I/O Board)	Rack	Slot
1	PrcI/O	AA	0	1
2	PrcI/O	BA	0	2
3	PrcI/O	CA	0	3
4	PrcI/O	DA	0	4

Table 4–25. Kinds of Process I/O available on HandlingTool

	Device and Name	Description
1.	PrcI/O AA	Process I/O board AA
2.	PrcI/O AB	Process I/O board AB
3.	PrcI/O BA	Process I/O board BA
4.	PrcI/O BB	Process I/O board BB
5.	PrcI/O CA	Process I/O board CA
6.	PrcI/O CB	Process I/O board CB
7.	PrcI/O DA	Process I/O board DA
8.	Laser	Laser I/O

4.6.4 Model A I/O

A maximum of four Model A racks and Model B Interface Modules can be connected to the I/O Link. If only Model A racks are used then the limit is the maximum allowed on the I/O Link of 512 bits (64 bytes) of I/O.

Racks and Interface Modules are assigned starting at Rack 1. A system with four Model A racks would be displayed on the I/O Link Screen as shown in Table 4–26 .

Table 4–26. Model A I/O Assignments

	Device	Name	Rack	Slot
1	Model	A	1	0
2	Model	A	2	0
3	Model	A	3	0
4	Model	A	4	0

4.6.5 Model B I/O

A maximum of four Model B Interface Modules and Model A racks can be connected to the I/O Link. Each Model B Interface Module has four communication channels (designated S1-S4) over which a maximum of 30 I/O units can be connected. The maximum amount of I/O which an Interface Module can support is 256 I/O. However, 32 bits (4 bytes) of Input are reserved by the system to report Module status. This limits the user accessible I/O to 224I/256O. A system with four Model B Interface Modules would be displayed on the I/O Link Screen as shown in Table 4–27 .

Table 4–27. Model B I/O Assignments

	Device	Name	Rack	Slot
1	Model	B	1	0
2	Model	B	2	0
3	Model	B	3	0
4	Model	B	4	0

Separate Interface Modules and Cables exist for both A-size and B-size Cabinets. Each of the 4 communication channels (S1-S4) can support two cables. Therefore, each Interface Module can support up to eight (8) communication cables. The total length of each cable must not exceed 100m). A total of 30 DI/DO Units can be distributed over the communication cables. If all 30 DI/DO Units are connected on one cable then none can be connected to the other seven cables.

4.6.6 I/O Link Device Screen

This screen lists all process I/O boards, model A I/O racks, model B interface units, Terminal I/O modules (ArcTool only), and PLC like devices connected to the controller through the I/O-LINK connector on the MAIN CPU printed circuit board.

Table 4–28 lists the I/O Link Device List Screen Items.

Table 4–28. I/O Link Device List Screen Items

ITEM	DESCRIPTION
Device Name	This item is a listing of the different kinds of I/O devices in your system.

Table 4–28. I/O Link Device List Screen Items (Cont'd)

Comment	This item is a comment field into which you can type information pertaining to the signal. To type a comment, 1. Move the cursor to the appropriate line. 2. Press ENTER. 3. Type the comment. 4. Press ENTER.
Rack	This item is the physical location on which the I/O board or module is mounted.
Slot	This item is the space on the rack where the I/O module is connected.

Figure 4–5 is an example of the I/O link device screen when

- Process I/O board CB is connected to JD1A of the controller
- One I/O unit model B interface is connected
- Two I/O unit Model A racks are connected

Figure 4–5. I/O Link Device Screen

```
I/O Link Device
   Device Name      Comment     RackSlot
  1 PrcI/O AA [                ] 0   1
  2 Model B   [                ] 1   0
  3 Model A   [                ] 2   0
  4 Model A   [                ] 3   0
```

This menu is displayed when you press I/O, F1, [TYPE] and then select Link Device.

Table 4–29 contains descriptions of the device names displayed on the I/O Link Device screen.

Table 4–29. Device Names

Device name displayed	Device Description
PrcI/O AA	Process I/O board AA
PrcI/O AB	Process I/O board AB
PrcI/O BA	Process I/O board BA
PrcI/O BB	Process I/O board BB
PrcI/O CA	Process I/O board CA
PrcI/O CB	Process I/O board CB
PrcI/O DA	Process I/O board DA
PrcI/O EA	Process I/O board EA
PrcI/O EB	Process I/O board EB
PrcI/O GA	Process I/O board GA
Laser	Laser I/O
MODEL A	FANUC I/O UNIT MODEL A
MODEL B	FANUC I/O UNIT MODEL B
90-30 PLC	GEFanuc 90-30 PLC slave mode interface unit
I/O adptr	I/O Link adapter
JEMA PC	JEMA PC
R-J2 Mate	R-J2 Mate slave mode
Weld I/F	Weld I/F board
Unknown	Terminal I/O (ArcTool only)

The slot value of Model A and Model B on this screen is 0.

For devices whose number of ports cannot be decided automatically, you can use the DETAIL screen to set the number of ports manually. See Procedure 4-8 .

The devices that have access to the DETAIL screen are listed in Table 4–30 .

Table 4–30. Devices that have Access to the DETAIL Screen

Device name displayed	Device Description
MODEL B	MODEL B unit setting
90-30 PLC	Number of ports setting
I/O adptr	Number of ports setting

Table 4–30. Devices that have Access to the DETAIL Screen (Cont'd)

Device name displayed	Device Description
JEMA PC	Number of ports setting
R-J2 Mate	Number of ports setting
Unknown	Terminal I/O setting

You can add a comment for every device. Comment data is linked to rack, slot and device type. After hardware configuration is changed, if rack, slot and device type are matched, the comment of this device is displayed. If rack, slot or device type are not matched, the comment of this device is not displayed.



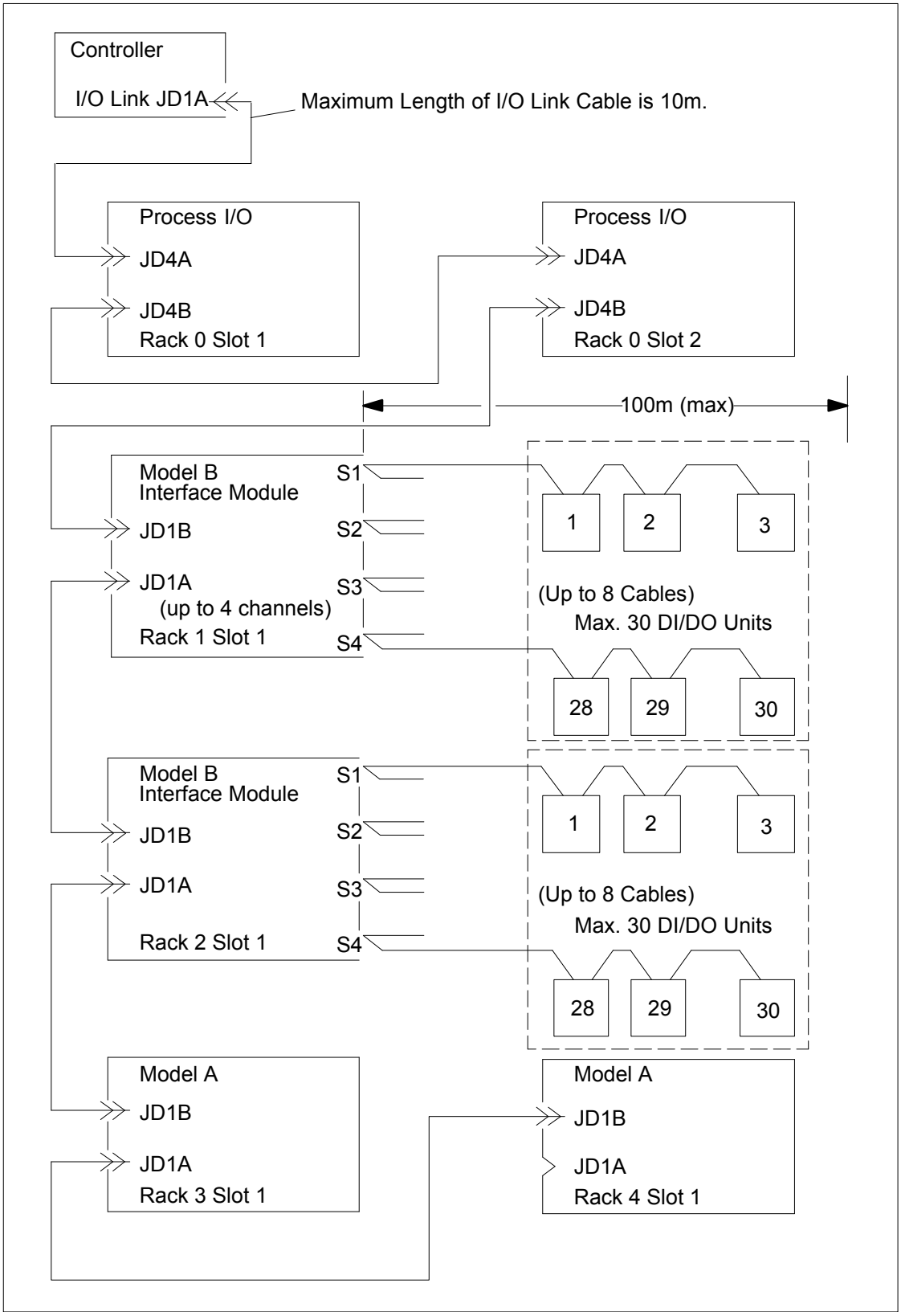
Caution

CLR_ASG clears assignments of all ports on all units, including process I/O, model A, model B, Terminal I/O (ArcTool only), and PLC devices. The next time the controller is turned on, ports for these devices will be given default assignments, if the system variables $IO_AUTO_CFG and $IO_AUTO_UOP are set to TRUE.

4.6.7 I/O Link Connection

Figure 4–6 shows a controller with two Process I/O Modules, two Model B Interface Modules and two Model A racks. The Process I/O Module is defined at Rack 0 Slot 1 and Rack 0 Slot 2. Some Process I/O modules get their power from the I/O Link Cable. This Power is not passed on in the Model A and Model B I/O Link Cable. The two Model B Interface Modules are defined as Rack 1/Slot 1 and Rack 2/Slot 1 respectively, and the two Model A racks are defined as Rack 3/Slot 1 and Rack 4/Slot 1 respectively.

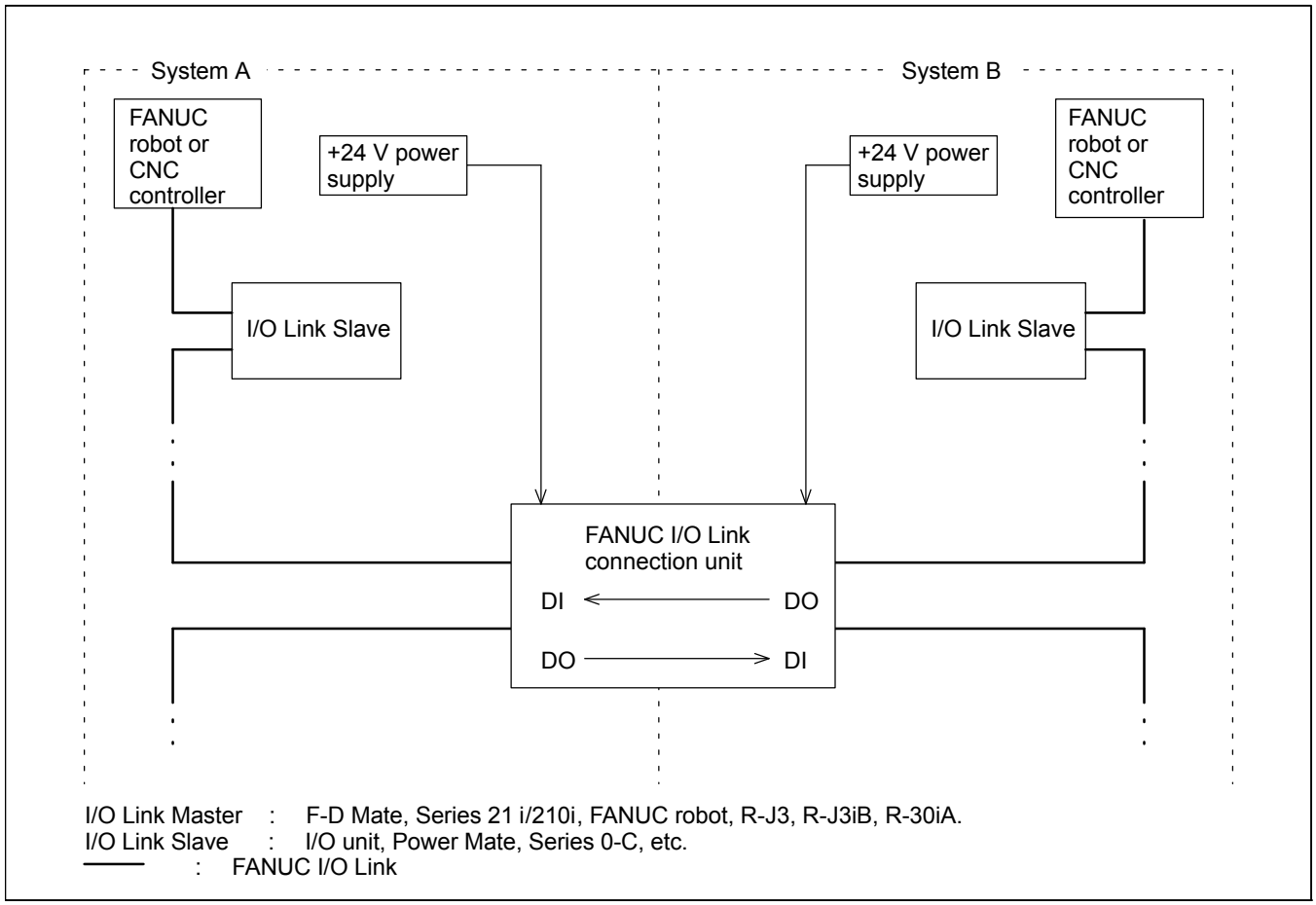
Figure 4–6. I/O Link Diagram



4.6.8 FANUC I/O Link Connection Unit

This unit connects FANUC I/O Link master devices such as the CNC and robot, via an I/O Link to enable the transfer of DI/DO signals. See Figure 4–7 .

Figure 4–7. System that Uses FANUC I/O Link Connection Units



Note This system enables I/O data transfer between two independent FANUC I/O Link master devices. When the system is adjusted and maintained, the FANUC I/O Link can be operated with the system power for one of the FANUC I/O Link lines switched off, that is, the link operation is stopped. In this case, DI data sent from a system at rest consists entirely of zeros. If one of the links is stopped, either abnormally or normally, it takes up to several hundred milliseconds for this function to take effect. During this period, that data which exists immediately before the link stops is sent out. Take this into account when designing your system.

Specifications

Table 4–31 lists the FANUC I/O Link specifications.

Table 4–31. FANUC I/O Link Specifications

Item	Specification
I/O Link function	Provided with two slave mode I/O Link interface channels, between which DI/DO data can be transferred. [Interface types] One of the following combinations is selected: Electrical - optical Electrical - electrical Optical - optical
Number of DI/DO data items	DI: Up to 256, DO: Up to 256 The number of data items actually used varies depending on the amount of data assigned in the host.
Power supply	Each I/O Link interface must be independently supplied with +24 VDC. Voltage: +24 VDC +10%, -15% Current: 0.2 A (excluding surge) If a master unit does not have sufficient capacity to supply power to each unit (0.2 A per slot), use an external power supply unit. The power supply must be switched on, either simultaneously with or before, the I/O Link master. The two systems can be switched on and off independently of each other. Data from a system to which no power is supplied appears as zeros when viewed from the other system. The data becomes 0 within 200 ms of the power being switched off.
External dimensions	180 mm (wide) 150 mm (high) about 50 mm (deep). Figure 4–9 is an outline drawing of the unit.
Installation	The unit, which is a stand-alone type, is installed in the power magnetics cabinet. Figure 4–10 shows how to mount the unit.
Operating environment	Temperature : 0 to 60°C Humidity : 5 to 75% RH (non-condensing) Vibration : 0.5 G or less

Ordering Information

Table 4–32 lists FANUC I/O Link ordering information.

Table 4–32. FANUC I/O Link Ordering Information

Interface type	Specification
Electrical-optical interface	A20B-2000-0410
Electrical-electrical interface	A20B-2000-0411
Optical-optical interface	A20B-2000-0412

LED Indicators

Figure 4–8 shows the locations of the LEDs on the FANUC I/O Link. Table 4–33 shows LED status information. Figure 4–9 and Figure 4–10 show an outline drawing and mounting location drawing.

Figure 4–8. LED Locations

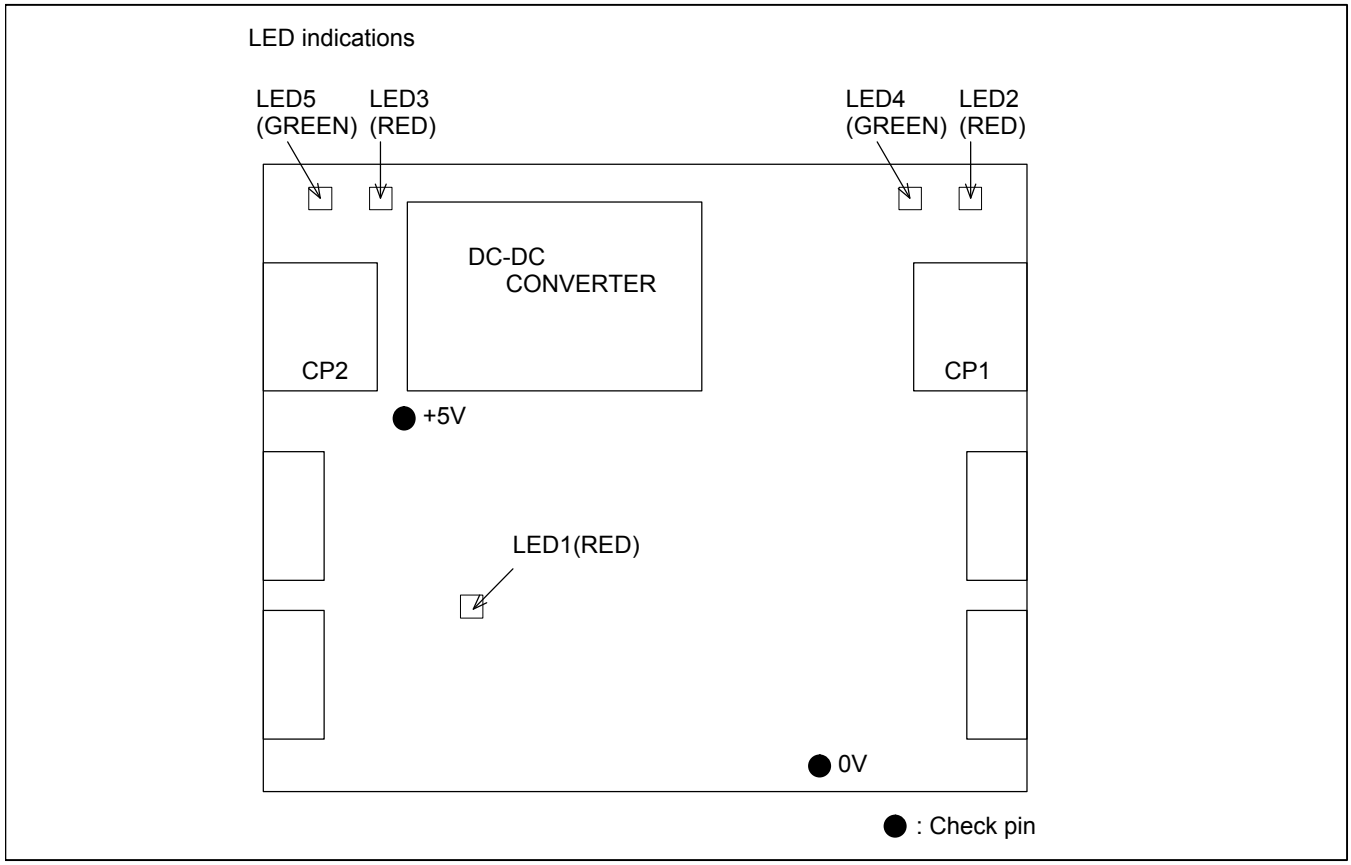


Table 4–33. LED Status Descriptions

	LED status	Description
1	LED1□	Normal
	LED1■	A RAM parity error occurred because of a hardware failure.
2	LED4■ LED2 □	CP1 is supplied with the specified voltage. (Pilot lamp)
	LED4□ LED2 ■	CP1 is supplied with a voltage that is lower than specified or zero.
	LED4■ LED2 ■	A communication error occurred in a channel of CP1.

Table 4–33. LED Status Descriptions (Cont'd)

	LED status	Description
3	LED5■ LED3 □	CP2 is supplied with the specified voltage. (Pilot lamp)
	LED5□ LED3 ■	CP2 is supplied with a voltage that is lower than specified or zero.
	LED5■ LED3 ■	A communication error occurred in a channel of CP2.

■: On □: Off

Figure 4–9. Outline Drawing

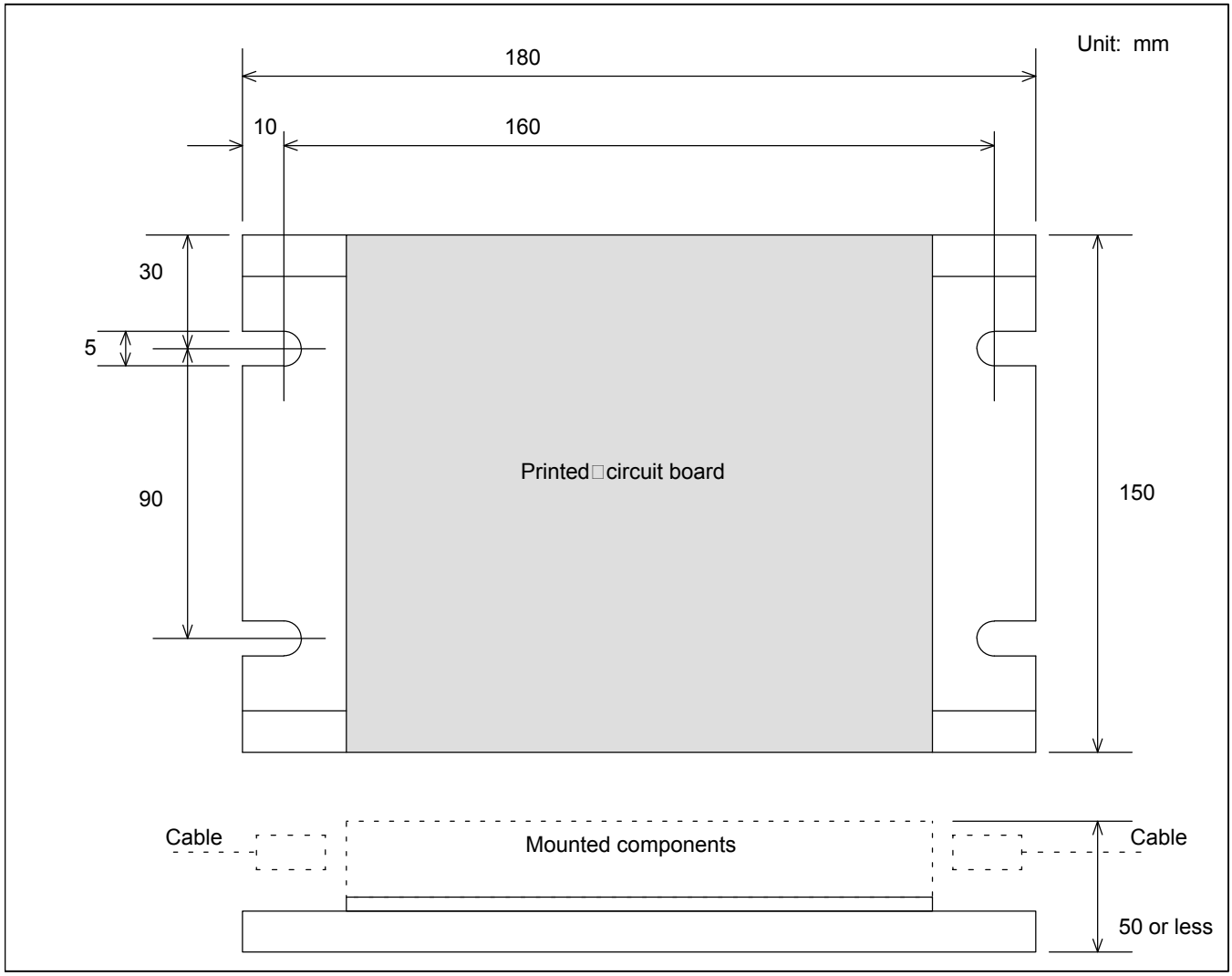
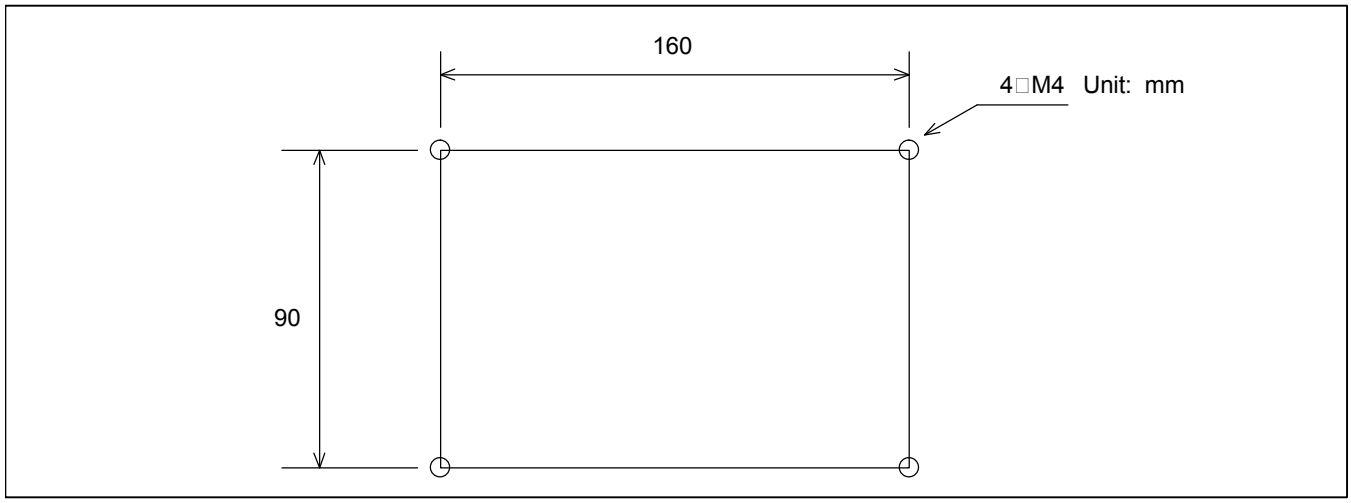


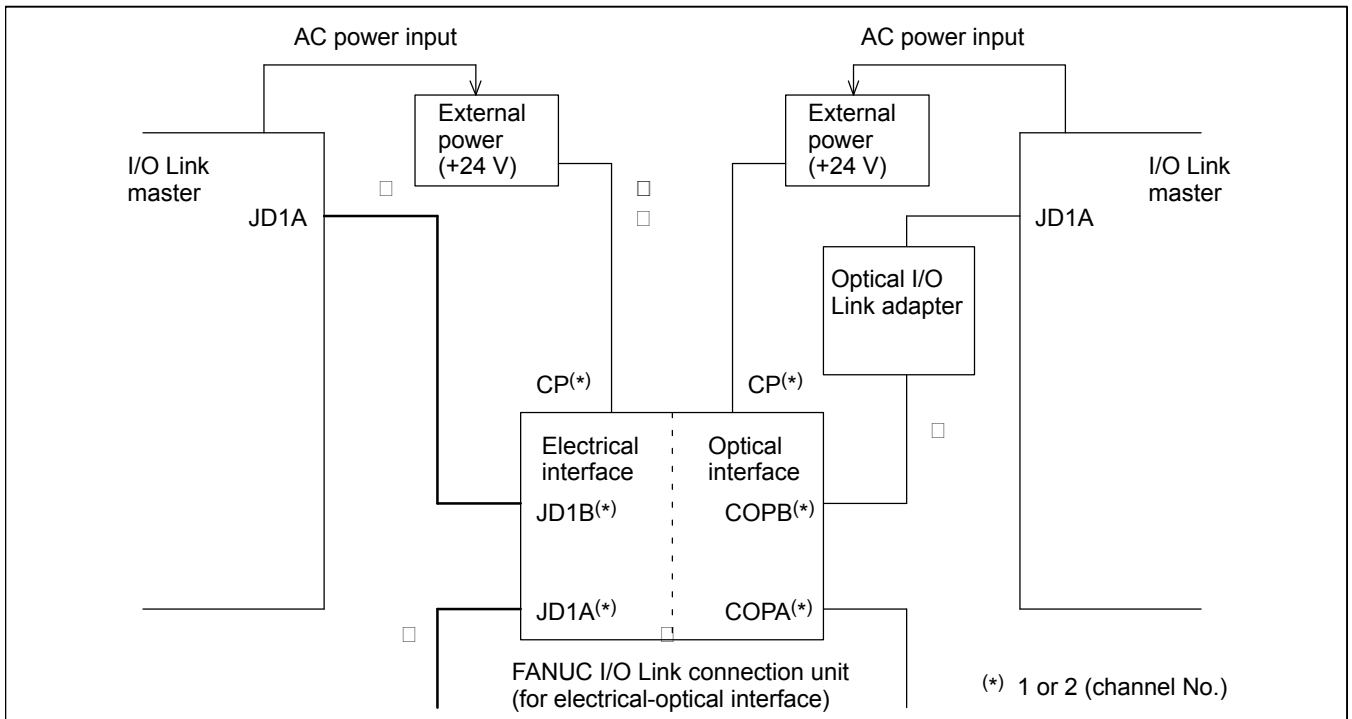
Figure 4–10. Mounting Location



Connection for I/O Link Interface

An example connection diagram is shown in Figure 4–11 .

Figure 4–11. Example Connection Diagram



[Name of I/O Link connection unit connectors]

Electrical-optical

Connector name I/O Link interface	
Channel 1	Channel 2
JD1A1	COPA2
JD1B1	COPB2
CP1	CP2

Electrical-electrical

Connector name I/O Link interface	
Channel 1	Channel 2
JD1A1	JD1A2
JD1B1	JD1B2
CP1	CP2

Optical-optical

Connector name I/O Link interface	
Channel 1	Channel 2
COPA1	COPA2
COPB1	COPB2
CP1	CP2

The electrical signal cable connectors are shown in Figure 4–12 .

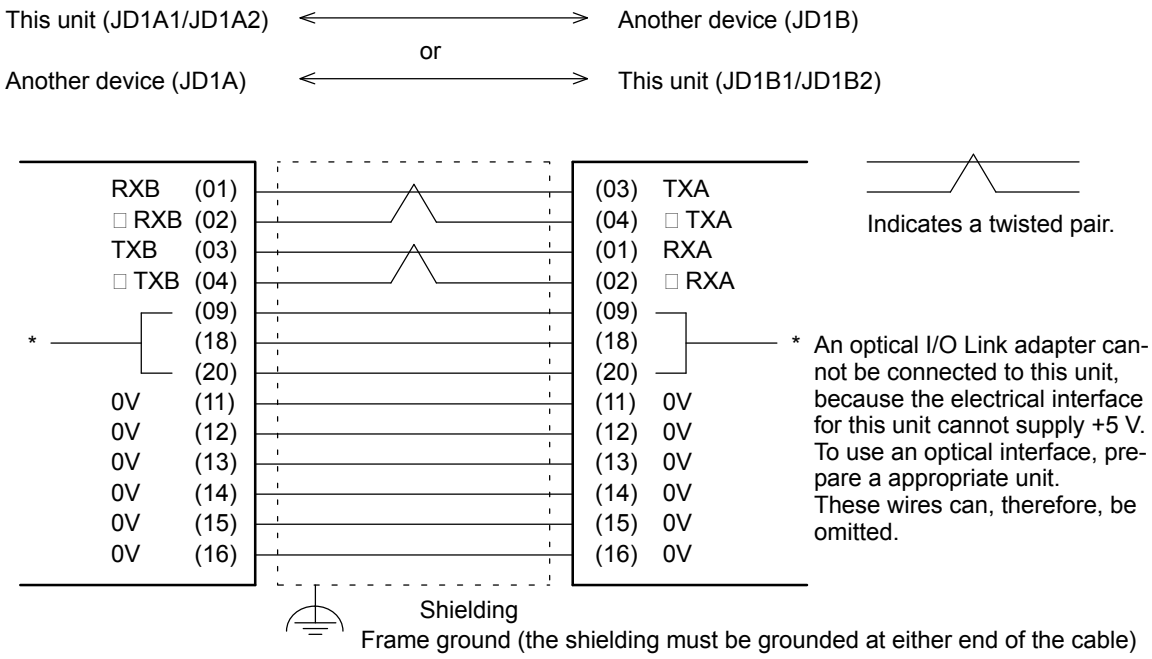
Figure 4–12. Electrical Signal Cable Connectors

JD1A1/JD1A2

11	0V	1	RXB
12	0V	2	□ RXB
13	0V	3	TXB
14	0V	4	□ TXB
15	0V	5	
16	0V	6	
17		7	
18	-	8	
19		9	-
20	-	10	

JD1B1/JD1B2

11	0V	1	RXA
12	0V	2	□ RXA
13	0V	3	TXA
14	0V	4	□ TXA
15	0V	5	
16	0V	6	
17		7	
18	-	8	
19		9	-
20	-	10	



□ Cable-side connector specificatio n : PCR-E20F A (manufactured by Honda Tsushin)
□ Cable specification : A66L-0001-0284#10P or equivalent
□ Cable length : 10 m (maximum)

The signal cable (optical) specifications are as follows:

- Optical cable specification: A66L-6001-0009#XXXX (where XXXX is a cable length specification)

  Cable specification examples:

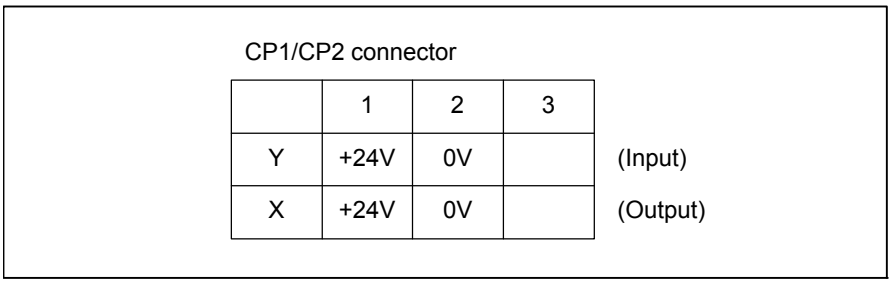
  — 10 m - L10R03

  — 100 m - L100R3

- Cable length: 200 m (maximum)

The power supply cable connector is shown in Figure 4–13 and its specifications follow.

Figure 4–13. Power Supply Cable Connector



- 24 VDC is supplied via a Y-connector. **If the power supply has sufficient capacity,** power can be supplied to another device with the X-side as output.
- Power must be supplied to both CP1 and CP2.
- Cable-side connector specification
- Y-connector: A63L-0001-0460#3LKY (AMP Japan, 2-178288-3)

  X-connector: A63L-0001-0460#3LKX (AMP Japan, 1-178288-3)

  Contact: A63L-0001-0456#BS (AMP Japan, 175218-5)

  Ordering information: Y + 3 contacts: A02B-0120-K323X + 3 contacts: A02B-0120-K324
- Cable material: Vinyl-insulated electrical wire AWG20-16
- Cable length: Determine the length of the cable such that the supplied voltage at the receiving end satisfies the requirements, because the voltage may fluctuate and drop as a result of the resistance of the cable conductor.

For frame grounding, ground the frame of the unit using a wire having a cross section of at least 5.5 m^2 (class 3 or higher). An M4 frame ground terminal is provided.

4.6.9 Setting the Number of Ports

Use Procedure 4-8 to set the number of ports you want to use.

Procedure 4-8 Setting the Number of Ports

Steps

1. Press MENUS.
2. Select I/O.
3. Press F1, [TYPE].
4. Select I/O Link. You will see a screen similar to the following.

```
I/O Link Device
   Device Name      Comment     RackSlot
  1 PrcI/O AA [                ] 0   1
  2 Model B   [                ] 1   0
  3 Model A   [                ] 2   0
  4 Model A   [                ] 3   0
```

5. Move the cursor to the line of 90-30 PLC, I/O Connect, JEMA PC, R-J Mate, R-J3 Mate, or Unknown in the I/O link device screen.
6. Press F3, DETAIL. You will see a screen similar to the following.

```
I/O Link Device
 90-30 PLC      Rack 1  Slot 1

      Port Name                    Points
  1 Digital Input                      0
  2 Digital Output                     0
```

7. Type the number of ports needed for your device and press ENTER.

Caution

CLR_ASG clears assignments of all ports on all units, including process I/O, model A, model B, Terminal I/O (ArcTool only), and PLC devices. The next time the controller is turned on, ports for these devices will be given default assignments, if the system variables $IO_AUTO_CFG and $IO_AUTO_UOP are set to TRUE.

8. **If you want to clear all assignments**,

   a. Press F5, CLR_ASG.

   The following message is displayed.

   ```
   Clear all assignments?
   ```

   b. Press the appropriate key:

- Press F4, YES to clear all I/O assignments.
- Press F5, NO not to clear all I/O assignments.

9. **To save the I/O settings** :

Note If you save DIOCFGSV.IO from a model A I/O menu or from the FILE [BACKUP] menu, you must also save the Model B I/O Setup data and comments.

   a. Press MENUS.
   b. Select FILE.
   c. Press F1, [TYPE].
   d. Select File.
   e. Press F5, [UTIL].
   f. Select Set Device.
   g. Move the cursor to the device you want and press ENTER.
   h. Press MENUS.
   i. Select I/O.
   j. Press FCTN.
   k. Select SAVE. The file will be saved to the DIOCFGSV.IO file on the default device.

10. After you set up detail information, you must turn OFF the controller. Then turn it back ON for the new information to take effect.

4.7 I/O INTERCONNECT SETUP

The I/O interconnect feature allows you to output the states of robot digital input (RI), digital input (DI), Standard Operator Panel (SI), and Emergency Stop (ES) signals to digital output (DO) signals and (DI) robot digital output (RO) signals to notify external devices of the input states of the signals.

With I/O InterConnect, you can do the following:

- Redirect the status of a RI signal to a DO signal

  RI[m] -> DO[n], where

  — m: RI signal number

  — n: 1-32766

- Redirect the status of a DI signal to a RO signal

  DI[i] -> RO[j], where

  — i: 1-32766

  — j: RO signal number

- Redirect the status of a DI signal to a DO signal

  DI[k] -> DO[l], where

  — k: 1-32766

  — l: 1-32766

- Redirect the status of an SI signal to a DO signal

  SI[q] -> DO[r], where

  — q: SI signal number

  — r: 1-32766

- Redirect the status of an emergency stop (ES) signal to a DO signal

  ES -> DO[t], where

  — ES: emergency stop signal

  — t: 1-32766

You use the I/O Interconnect screen to connect signals and enable and disable the connections. For example, when "ENABLE DI[2]->RO[3]" is set, the state of DI[2] is output to RO[3].

Note I/O interconnection changes take effect immediately. It is *NOT* necessary to turn the controller off then on for these changes to take effect.

Restrictions

You have the following restrictions when you use I/O Interconnect:

- When the redirection of DI[i] to DO[j] is enabled, the state of DI[i] is periodically output to DO[j]. In this case, if DO[j] is changed, it will be overwritten.
- The redirection of each signal can be enabled or disabled only from the relevant page of the I/O Interconnect screen.
- If two or more input signals are redirected to an output signal, the state of each input signal is redirected to the output signal.

```
1 ENABLE     RI[  1] -> DO[  1]
2 ENABLE     RI[  2] -> DO[  1]
```

For example, if the signals are redirected as above, the output of DO[1] will be unpredictable when RI[1] is turned on and RI[2] is turned off (actually, DO[1] is repeatedly turned on and off).

- You can configure the number of inputs connected to outputs using the Controlled Start Program Limits Setup screen. Refer to the *Software Installation Manual* for more information.

Table 4–34 lists and describes each item on the I/O InterConnect menu. The I/O InterConnect screens are shown in Procedure 4-9 .

Table 4–34. I/O InterConnect Screen Items

ITEM	DESCRIPTION
No.	This item displays the line number of the interconnect. The ITEM key can be used to select a particular line.
Enb/Disabl	This item specifies whether or not to redirect the signal. If set to **ENABLE** the signal will be redirected. If set to **DISABLED**, the signal will not be redirected. If the signal number of the DO or the DI is 0, then the signal will not be redirected.
Input	This item displays the RI, DI, SI, or ES signal that will be redirected. RI, SI, and ES signal numbers cannot be modified.
Output	This item displays the RO or DO signal that will receive the status for the input signal. The RO signal number cannot be modified.

Use Procedure 4-9 to use I/O InterConnect.

Procedure 4-9 Setting Up I/O InterConnect

Steps

1. Press MENUS.
2. Select I/O.
3. Press F1, [TYPE].
4. Select Interconnect. You will see a screen similar to the following.

```
INTERCONNECT
 No. Enb/Disabl  INPUT         OUTPUT
  1  ENABLE    RI[    1] ->  DO[    0]
  2  DISABLE   RI[    2] ->  DO[    0]
  3  DISABLE   RI[    3] ->  DO[    0]
  4  DISABLE   RI[    4] ->  DO[    0]
  5  DISABLE   RI[    5] ->  DO[    0]
  6  DISABLE   RI[    6] ->  DO[    0]
  7  DISABLE   RI[    7] ->  DO[    0]
  8  DISABLE   RI[    8] ->  DO[    0]
```

5. Press F3, [SELECT]. If RI -> DO had been selected previously, you will see a screen similar to the following.

```
INTERCONNECT

 No. Enb/Disabl  INPUT         OUTPUT
  1  DISABLE   RI[    1] ->  DO[    0]
  2  DISABLE   RI[    2] ->  DO[    0]
  3  DISABLE   RI[    3] ->  DO[    0]
  4  DISABL____________________O [    0]
  5  DISABL| 1  RI-> DO      |O [    0]
  6  DISABL| 2  DI-> RO      |O [    0]
  7  DISABL| 3  DI-> DO      |O [    0]
  8  DISABL| 4  SI-> DO      |O [    0]
           | 5  ES-> DO      |
           -----+       +-----
[ TYPE ]        |SELECT| ENABLE DISABLE
```

6. Select the kind of redirection you want:

- **To redirect RI to DO**, select 1, RI->DO.
- **To redirect DI to RO**, select 2, DI->RO.
- **To redirect DI to DO**, select 3, DI->DO.
- **To redirect SI to DO**, select 4, SI->DO.
- **To redirect ES to DO**, select 5, ES->DO.

Note RI, RO, SI, and ES signal numbers cannot be changed.

If you select RI -> DO, you will see a screen similar to the following.

```
INTERCONNECT
 No. Enb/Disabl  INPUT         OUTPUT
  1  DISABLE   RI[    1] ->  DO[    0]
  2  DISABLE   RI[    2] ->  DO[    0]
  3  DISABLE   RI[    3] ->  DO[    0]
  4  DISABLE   RI[    4] ->  DO[    0]
  5  DISABLE   RI[    5] ->  DO[    0]
  6  DISABLE   RI[    6] ->  DO[    0]
  7  DISABLE   RI[    7] ->  DO[    0]
  8  DISABLE   RI[    8] ->  DO[    0]
```

If you select DI -> RO, you will see a screen similar to the following.

```
INTERCONNECT
 No. Enb/Disabl    INPUT        OUTPUT
  1  ENABLE     DI[    0] ->  RO[    1]
  2  DISABLE    DI[    0] ->  RO[    2]
  3  DISABLE    DI[    0] ->  RO[    3]
  4  DISABLE    DI[    0] ->  RO[    4]
  5  DISABLE    DI[    0] ->  RO[    5]
  6  DISABLE    DI[    0] ->  RO[    6]
  7  DISABLE    DI[    0] ->  RO[    7]
  8  DISABLE    DI[    0] ->  RO[    8]
```

If you select DI -> DO, you will see a screen similar to the following.

```
INTERCONNECT
 No. Enb/Disabl   INPUT         OUTPUT
  1  ENABLE     DI[    0] ->  DO[    0]
  2  DISABLE    DI[    0] ->  DO[    0]
  3  DISABLE    DI[    0] ->  DO[    0]
  4  DISABLE    DI[    0] ->  DO[    0]
  5  DISABLE    DI[    0] ->  DO[    0]
  6  DISABLE    DI[    0] ->  DO[    0]
 32  DISABLE    DI[    0] ->  DO[    0]
```

If you select SI -> DO, you will see a screen similar to the following.

```
INTERCONNECT
 No. Enb/Disabl   INPUT          OUTPUT
  1  ENABLE     SI[    0] ->  DO[    0]
  2  DISABLE    SI[    1] ->  DO[    0] <--RESET
  3  DISABLE    SI[    2] ->  DO[    0]
  4  DISABLE    SI[    3] ->  DO[    0]
  5  DISABLE    SI[    4] ->  DO[    0]
  6  DISABLE    SI[    5] ->  DO[    0]
  7  DISABLE    SI[    6] ->  DO[    0] <--START
  8  DISABLE    SI[    7] ->  DO[    0]
  9  DISABLE    SI[    8] ->  DO[    0] <--CE-1
 10  DISABLE    SI[    9] ->  DO[    0] <--CE-2

 16  DISABLE    SI[   16] ->  DO[    0]
```

Note The default DO numbers for RESET, CE-1 (MODE SELECT switch 1), CE-2 (MODE SELECT switch 2), and START are specified automatically. You can change these numbers if desired.

The relationship between the MODE SELECT switch signals and the modes of operation is shown in Table 4–35 .

Table 4–35. Relationship Between the MODE SELECT Switch Signals and Modes of Operation

Signal	Mode of Operation		
	T2	T1	AUTO
CE-1	0	1	1
CE-2	0	0	1

If you select ES -> DO , you will see a screen similar to the following.

```
INTERCONNECT
 No. Enb/Disabl   INPUT        OUTPUT
  1  ENABLE   [EMGOP    ] -> DO[    0] <--SOP E-STOP
  2  DISABLE  [EMGTP    ] -> DO[    0] <--TP E-STOP
  3  DISABLE  [DEADMAN  ] -> DO[    0] <--DEADMAN
  4  DISABLE  [FENCE    ] -> DO[    0] <--FENCE OPEN
  5  DISABLE  [ROT      ] -> DO[    0]
  6  DISABLE  [HBK      ] -> DO[    0]
  7  DISABLE  [EMGEX    ] -> DO[    0] <--EXTERNAL
  8  DISABLE  [PPABN    ] -> DO[    0]    E-STOP
  9  DISABLE  [BELTBREAK] -> DO[    0]
 10  DISABLE  [FALM     ] -> DO[    0]
 11  DISABLE  [SVON     ] -> DO[    0]
 12  DISABLE  [IMSTP    ] -> DO[    0] <--UOP IMSTOP
 13  DISABLE  [BRKHLD   ] -> DO[    0]
 14  DISABLE  [USRALM   ] -> DO[    0]
 15  DISABLE  [SRVDSCNCT] -> DO[    0]
 16  DISABLE  [NTED     ] -> DO[    0]
```

Note Refer to the *FANUC Robotics Controller Maintenance Manual* for more information on emergency stop signals.

7. For each signal you want to redirect, type the signal number of the DI or DO and press ENTER.

Note If the signal number of the DO or the DI is 0, **the signal will not be redirected.**

8. For each signal you want to redirect, enable or disable the redirection of the signal:

- **To enable the redirection**, press F4, ENABLE.
- **To disable the redirection**, press F5, DISABLE.

Note I/O interconnection changes take effect immediately. It is *NOT* necessary to turn the controller on then off for these changes to take effect.

Note The response time to update a signal is from 64ms.

4.8 CONTROLLING I/O

4.8.1 Overview

Controlling I/O allows you to test the I/O in your system for proper function during testing operations. Controlling I/O includes:

- Forcing outputs
- Simulating inputs and outputs

4.8.2 Forcing Outputs

Forcing outputs is turning output signals on or off. Outputs can also be forced within a program using I/O instructions. Refer to the Input/Output Instructions section of this manual, or to the *KAREL Reference Manual*. Use Procedure 4-10 to force outputs outside of a program.

Note RO[1] and RO[2] control the same signals as HAND 1, and RO[3] and RO[4] control the same signals as HAND 2.

Procedure 4-10 Forcing Outputs

Conditions

- The outputs you are forcing have been configured. Refer to Section 4.2.2 .

Steps

1. Press MENUS.
2. Select I/O.
3. Press F1, [TYPE].
4. Select the kind of output you want to force: digital, analog, group, robot, UOP, or SOP.



Warning

Forcing digital outputs causes connected devices to function. Make certain you know what the digital output is connected to and how it will function before you force it; otherwise, you could injure personnel or damage equipment.

For digital outputs for example, you will see a screen similar to the following.

```
I/O Digital Out
      #   SIM STATUS
  DO[   1] U   OFF      [               ]
  DO[   2] U   ON       [               ]
  DO[   3] U   OFF      [               ]
  DO[   4] U   OFF      [               ]
  DO[   5] U   OFF      [               ]
  DO[   6] U   ON       [               ]
  DO[   7] U   OFF      [               ]
  DO[   8] U   OFF      [               ]
  DO[   9] U   OFF      [               ]
  DO[  10] U   OFF      [               ]
```

5. Move the cursor to the STATUS of the output you want to force.
6. Press the function key that corresponds to the value you want.

For digital, robot, UOP, and SOP outputs, press

- F4 for ON
- F5 for OFF

For analog and group outputs, move the cursor to value, and use the numeric keys to type the value. Value entry is always in decimal format. To change the displayed value from decimal to hexadecimal, press F4, FORMAT. Hexadecimal numbers are followed by an " H" on the screen.

4.8.3 Simulating Inputs and Outputs

Simulating inputs and outputs is forcing inputs and outputs without signals entering or leaving the controller. This can be used to test program logic and motion when I/O devices and signals are not set up. You can simulate digital, analog, group, and robot I/O only; you cannot simulate UOP or SOP I/O. When you are finished simulating a signal you can reset, or *unsimulate* , it. Use Procedure 4-11 to simulate and unsimulate I/O.

Procedure 4-11 Simulating and Unsimulating Inputs and Outputs

Conditions

- The input or output has been configured and is currently valid. Refer to Section 4.2.2 .

Steps

1. Press MENUS.
2. Select I/O.

3. Press F1, [TYPE].
4. Select the type of input or output you want to simulate: digital, analog, or group.

   For digital inputs for example, you will see a screen similar to the following.

```
I/O Digital Input
      #   SIM STATUS
  DI[   1] U   OFF       [              ]
  DI[   2] U   ON        [              ]
  DI[   3] U   OFF       [              ]
  DI[   4] U   OFF       [              ]
  DI[   5] U   OFF       [              ]
  DI[   6] U   ON        [              ]
  DI[   7] U   OFF       [              ]
  DI[   8] S   OFF       [              ]
  DI[   9] U   OFF       [              ]
  DI[  10] U   OFF       [              ]
```

5. **If you simulate a signal,** you can force the status by setting it to a value. When the signal is unsimulated, its actual status is displayed.
6. Move the cursor to the SIM column of the signal you want to simulate.
   - **U** means the signal is not simulated or unsimulated.
   - **S** means the signal is simulated.
7. Simulate or unsimulate the signal.
   - **To simulate**, press F4, SIMULATE.
   - **To unsimulate**, press F5, UNSIM.
8. **To unsimulate all simulated signals,** press FCTN and then select UNSIM ALL I/O.

   Note If you disable Digital/Analog I/O from the TEST CYCLE SETUP screen, all I/O becomes simulated. When you re-enable Digital/Analog I/O from the TEST CYCLE SETUP screen, ports that were simulated using the SIMULATE function key are still simulated.

4.8.4 Output When WAITing on Input

Output when WAITing on Input provides a way to turn on a digital output that indicates that a WAIT instruction in a teach pendant program has been waiting on a specified digital input for longer than a specified time.

The feature allows you to define a range of inputs to be monitored, a timeout value and a digital output that will be turned on to indicate that a timeout has occurred.

With this feature enabled, if a teach pendant program is run and that program executes a WAIT instruction, the instruction will be checked for certain condition clauses that include a digital input that has an index within the range to be monitored. If the condition clauses containing those inputs are not satisfied within the timeout period and the overall WAIT has not been satisfied, the specified digital output will turn on.

For example, if the following setup conditions are set in the SYSTEM Configuration menu:

- Range of inputs to be monitored – 1 through 3
- Timeout value – 30 seconds
- Digital output to turn on – DO[1]

If the following teach pendant instruction is executed:

```
WAIT DI[1]=ON AND DI[2]=OFF AND DI[3]=ON+ AND DI[4]=ON AND DI[5]=ON
```

Then the following will occur:

The overall WAIT will not be satisfied and program execution will not continue until all the conditions in the instruction are satisfied (this is not changed by this feature). In addition to that, if the conditions for DI[1], DI[2] and DI[3] are not all satisfied within the timeout period of 30 seconds, the output DO[1] will turn on.

Refer to Section 3.15 to set up the Output when WAIT for Input items.

After Output when WAITing on Input has been set up, the inputs that are designated to be monitored can be used in a WAIT instruction and if the condition clauses they are used in are not satisfied within the timeout period, the designated output will turn on.

Only the following conditions will be monitored:

- DI[x] = ON
- DI[x] = OFF
- DI[x] = ON+
- DI[x] = OFF-

These conditions can be used in a regular WAIT statement or in a Mixed Logic WAIT statement. For example:

```
1:  WAIT DI[1]=ON AND DI[2]=ON AND
  :  DI[3]=ON

1:  WAIT DI[1]=ON OR DI[2]=ON OR
  :  DI[3]=ON

1:  WAIT ((DI[1]=ON OR (DI[2]=ON AND
  :  DI[3]=ON)) AND DI[4]=ON+)
```

Only the conditions listed above are supported. Other conditions are not supported, such as:

- DI[1] <> ON
- DI[1] = R[1]
- DI[1] = DI[2]
- DI[1] = DI[R[1]}
- DI[1] = AR[1]

The WAIT instruction can contain other condition clauses such as:

```
1:  WAIT DI[1]=ON AND R[2]= 10
```

If the WAIT instruction is satisfied before the timeout period and all the monitored digital inputs have not been satisfied, the specified digital output will not turn on when the timeout period expires.

4.9 USER OPERATOR PANEL (UOP) SIGNAL DEFINITION

4.9.1 Overview

This section contains information on the signal definitions of User Operator Panel (UOP) signals. Refer to Section 4.2 for information on setting up UOP signals.

You can set up and configure UOP I/O from the I/O UOP In/Out Monitor screen.

Table 4–36. I/O UOP In/Out Monitor Screen Items

ITEM	DESCRIPTION
I/O [#]	This item is the signal number of the UOP input or output.

Table 4–36. I/O UOP In/Out Monitor Screen Items (Cont'd)

STATUS	This item indicates the current status of the selected signal. To change this value, 1. Move the cursor to the appropriate line. 2. Move the cursor to the STATUS column. 3. Press F4, ON, or press F5, OFF.
COMMENT	This item is a comment field into which you can type information pertaining to the signal. To type a comment, 1. Move the cursor to the appropriate line. 2. Press F4, DETAIL. If you do not see DETAIL, press NEXT. 3. Press ENTER. 4. Type the comment. 5. Press ENTER. 6. Press PREV to return to the I/O UOP In/Out Monitor screen.

You can configure the range, rack, slot, and start values from the I/O UOP In/Out Configure screen.

Table 4–37. I/O UOP In/Out Configure Screen Items

ITEM	DESCRIPTION
I/O #	This item is the number of each signal range.
RANGE	This item is a range, or sequence, of signals. To change the range, 1. Move the cursor to the line you want to change. 2. Move the cursor to either the start range or the end range value. 3. Type the new value. 4. Press ENTER.

Table 4–37. I/O UOP In/Out Configure Screen Items (Cont'd)

RACK	This item is the physical location on which the I/O board or module is mounted. To change the rack value, 1. Move the cursor to the line you want to change. 2. Move the cursor to the RACK column. 3. Type the new value. 4. Press ENTER.
SLOT	This item is the space on the rack where the I/O module is connected. To change the slot value, 1. Move the cursor to the line you want to change. 2. Move the cursor to the SLOT column. 3. Type the new value. 4. Press ENTER.
START	This item is the port number within the sequence of ports on the board or module. To change the start value, 1. Move the cursor to the line you want to change. 2. Move the cursor to the START column. 3. Type the new value. 4. Press ENTER.
STAT	This item indicates the status of the UOP signal. Status can be one of the following: • ACTIV - assignment was valid at start up and is in effect. • UNASG - assignment has not been made. • INVAL - assignment is invalid based on the digital I/O hardware present when the controller was turned on. • PEND - assignment is valid but was made since the last time the controller was turned on and is therefore not active. You must turn off the controller and then turn it back on to make the change take effect.

4.9.2 UOP Input Signals

For systems with a Process I/O board, the UOP signals are already configured and assigned to dedicated ports. Table 4–38 lists the correspondence between UOP input signal names, UI signals, and the HONDA connector pin number. The UOP input signals are listed and described in Table 4–39 .

Note By default, if a process I/O board is connected to the controller, the UOP signals are automatically assigned as indicated in Table 4–38 . If you do not want to use the UOP signals, you must do the following:

- Use Table 4–38 to set RACK, SLOT, and START PT to zero for all UOP inputs.
- Use Table 4–38 to assign at least the first 8 DIN and DOUT ports of the process I/O board (RACK 0, SLOT 1) to some DIN and DOUT signals. Or you can set $IO_UOP_AUTO = FALSE.

Table 4–38. UOP UI to Process I/O Board DI

UOP Input Signals	Process I/OUOP UI	Honda Connector Pinout CRM2A
*IMSTP	UI 1	01
*HOLD	UI 2	02
*SFSPD	UI 3	03
CSTOPI	UI 4	04
FAULT RESET	UI 5	05
START	UI 6	06
HOME	UI 7	07
ENBL	UI 8	08
RSR1/PNS1	UI 9	09
RSR2/PNS2	UI 10	10
RSR3/PNS3	UI 11	11
RSR4/PNS4	UI 12	12
PNS5/RSR5†	UI 13	13
PNS6/RSR6†	UI 14	14
PNS7/RSR7†	UI 15	15
PNS8/RSR8†	UI 16	16
PNSTROBE	UI 17	29
PROD_START	UI 18	30

*A normally OFF signal held ON. When it is set to OFF, certain conditions will result. Refer to the UOP signal definitions.

†Used for motion group 2 in a multiple motion group system.

Table 4–39. UOP Input Signals

UOP Input Signal	Description
*IMSTP Always active UI[1]	This input is the immediate stop software signal. *IMSTP is a normally OFF signal held ON. When it is set to OFF, it • Pauses a program if one is running • Immediately stops the robot and applies robot brakes • Shuts off power to the servos Error code SRVO-037 *IMSTP Input (Group) will be displayed when this signal is lost. This signal is always active. **Warning** ***IMSTP is a software controlled input and cannot be used for safety purposes. Use *IMSTP with EMG1, EMG2, and EMGCOM to use this signal with a hardware controlled emergency stop. Refer to the maintenance manual for connection information of EMG1, EMG2, and EGMCOM.**
*HOLD Always active UI[2]	This input is the external hold signal. *Hold is a normally OFF signal, held ON. When it is set to OFF, it will do the following: • Pause program execution • Slow motion to a controlled stop and hold • Optional Brake on Hold shuts off servo power after the robot stops
*SFSPD Always active UI[3]	This input is the safety speed input signal. This signal is usually connected to the safety fence. *SFSPD is a normally OFF signal held ON. When it is set to OFF it will do the following: • Pause program execution • Reduce the speed override value to that defined in a system variable. This value cannot be increased while *SFSPD is OFF. • Display error code message SYST009. • Not allow a REMOTE start condition. Start inputs from UOP or SOP are disabled when SFSPD is set to OFF and only the teach pendant has motion control with the speed clamped.

Table 4–39. UOP Input Signals (Cont'd)

UOP Input Signal	Description
CSTOPI Always active UI[4]	This input is the cycle stop input. The function of this signal depends on the system variable $SHELL_CFG.$USE_ABORT. It the system variable $SHELL_CFG.$USE_ABORT is set to FALSE, the CSTOPI input • Clears the queue of programs to be executed that were sent by RSR signals **Warning** **When $SHELL_CFG.$USE_ABORT is set to FALSE, CSTOPI does not immediately stop automatic program execution.** • Automatic execution will be stopped after the current program has finished executing. If the system variable $SHELL_CFG.$USE_ABORT is set to TRUE, the CSTOPI input • Clears the queue of programs to be executed that were sent by RSR signals. • Immediately aborts the currently executing program(s) for programs that were sent to be executed by either RSR or PNS.
FAULT_RESET Always active UI[5]	This input is the external fault reset signal. When this signal is received the following will happen: • Error status is cleared • Servo power is turned on • The paused program **will not be resumed**
START Active when the robot is in a remote condition (CMDENBL = ON) UI[6]	This input is the remote start input. The function of this signal depends on the system variable $SHELL_CFG.$CONT_ONLY. **If the system variable $SHELL_CFG.$CONT_ONLY is set to FALSE** the START input signal • Resumes a paused program • If a program is aborted, the currently selected program starts from the position of the cursor. **If the system variable $SHELL_CFG.$CONT_ONLY is set to TRUE** the START input signal • Resumes a paused program only. The PROD_START input must be used to start a program from the beginning.

Table 4–39. UOP Input Signals (Cont'd)

UOP Input Signal	Description
HOME Active when the robot is in a remote condition (CMDENBL = ON) UI[7]	This input is the home input. When this signal is received the robot moves to the defined home position. You configure the system to do this by setting up a macro program to run when UI[7] is received.
ENBL Always active UI[8]	This input is the enable input. This signal must be ON to have motion control ability. When this signal is OFF, robot motion cannot be done. When ENBL is ON and the REMOTE switch on the operator panel is in the REMOTE position, the robot is in a remote operating condition.
RSR 1-4 Active when the robot is in a remote condition (CMDENBL = ON) UI[9-12]	These items are the robot service request input signals. When one of these signals is received, the corresponding RSR program is executed or, if a program is running currently, stored in a queue for later execution. RSR signals are used for production operation and can be received while an ACK output is being pulsed. See Figure 4–14 .
PNS 1-8 Active when the robot is in a remote condition (CMDENBL = ON) UI[9] - UI[16]	These items are program number select input signals. PNS selects programs for execution, **but does not execute programs.** Programs that are selected by PNS are executed using the START input or the PROD_START input depending on the value of the system variable $SHELL_CFG.$CONT_ONLY. The PNS number is output by using the SNO signal (selected number output) and the SNACK signal (selected number acknowledge) will be pulsed.
PNSTROBE Active when the robot is in a remote condition (CMDENBL = ON) UI[17]	This item is the program number select strobe input signal. See Figure 4–15 .
PROD_START Active when the robot is in a remote condition (CMDENBL = ON) UI[18]	This item is the Production Start Input when used with PNS will initiate execution of the selected program from the PNS lines. When used without PNS, PROD_START executes the selected program from the current cursor position. Coordinate with CYCLE START. See Figure 4–15 .

Figure 4–14 and Figure 4–15 provide information about the timing of the signals used with RSR and PNS.

Figure 4–14. RSR Timing Diagram

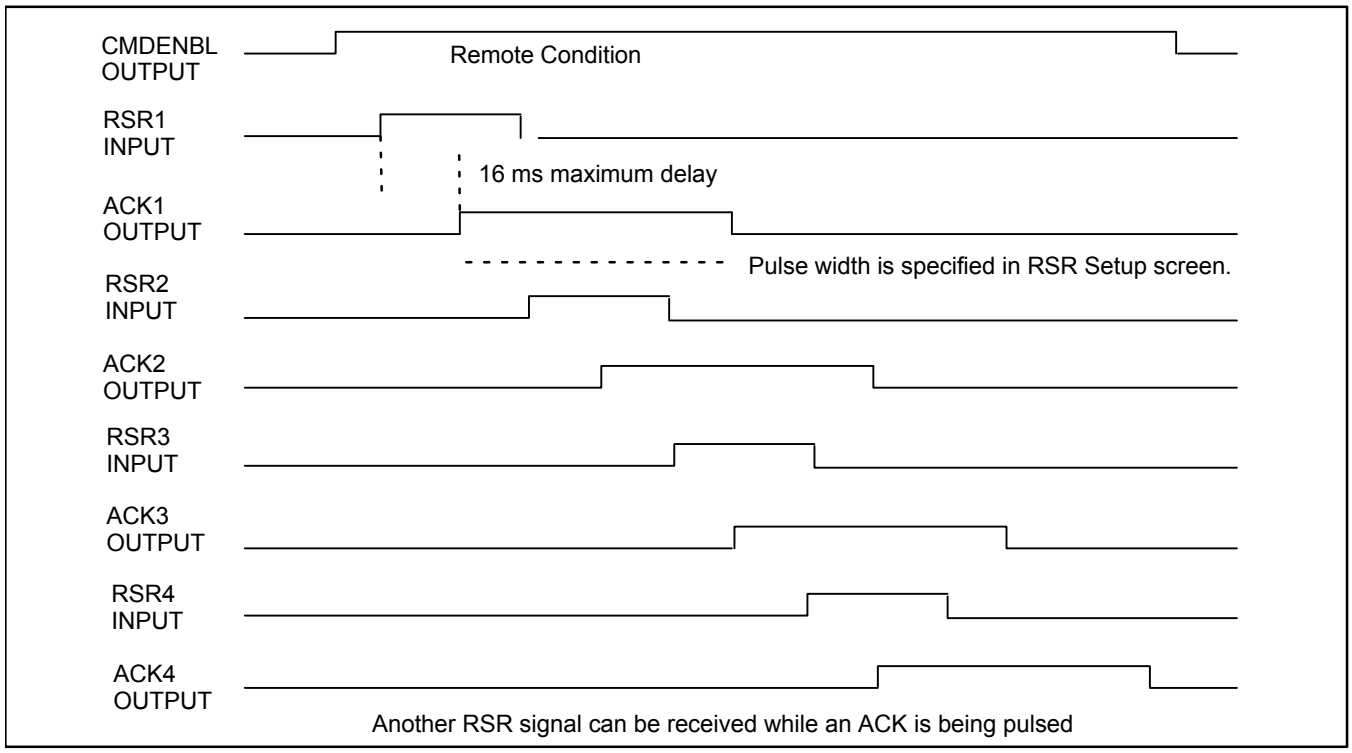
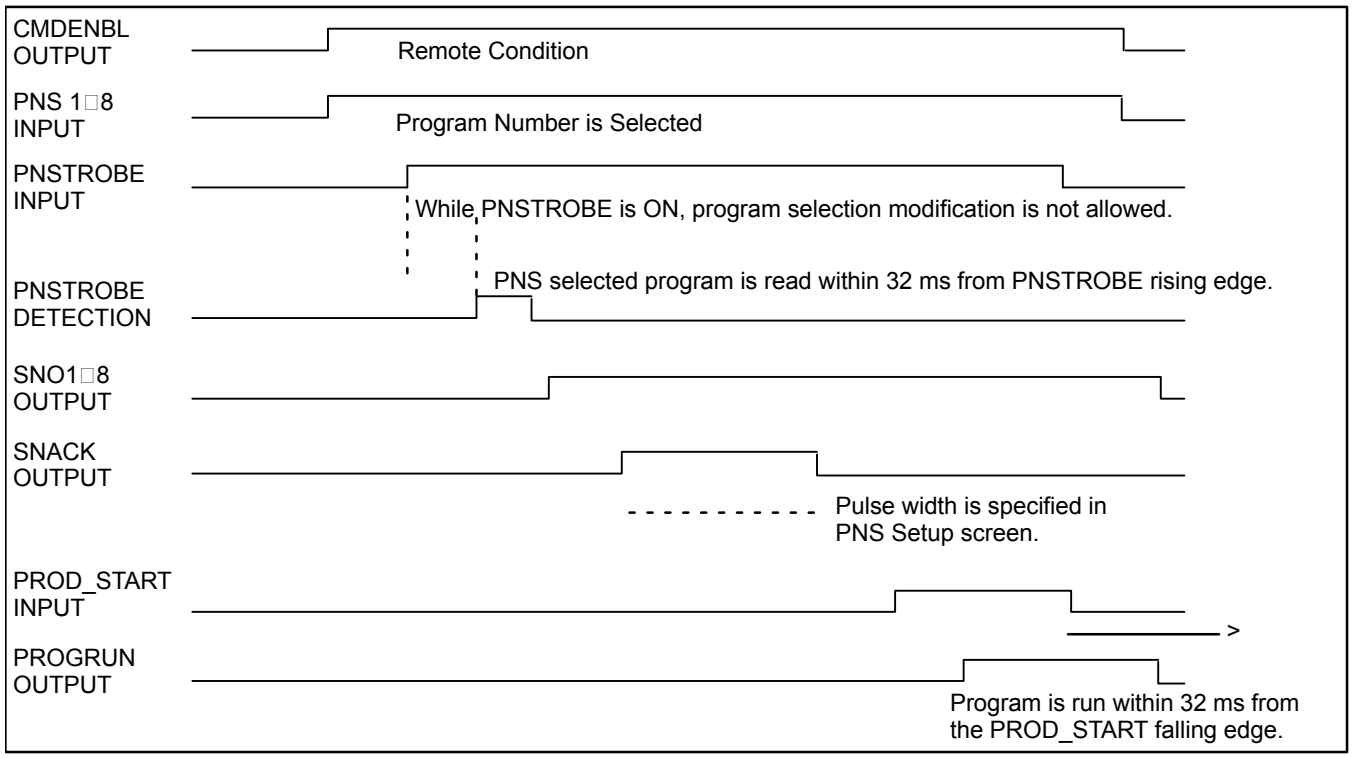


Figure 4–15. PNS Timing Diagram



4.9.3 UOP Output Signals

For systems with a process I/O board, the UOP signals are already configured and assigned to dedicated ports. Table 4–40 lists the correspondence between UOP input signals names, UO signals, and the HONDA connector pin number. The UOP has the output signals that are listed and described in Table 4–41 .

Note By default, if a process I/O board is connected to the controller, the UOP signals are automatically assigned as indicated in Table 4–40 . If you do not want to use the UOP signals, you must do the following:

- Use Table 4–40 to set RACK, SLOT, and START PT to zero for all UOP outputs.
- Use Table 4–40 to assign at least the first 8 DIN and DOUT ports of the process I/O board (RACK 0, SLOT 1) to some DIN and DOUT signals. Or you can set $IO_UOP_AUTO = FALSE.

Table 4–40. UOP Outputs to Process I/O Board DO

UOP Output Signals	Process I/O UOP UO	Honda Connector Pinout CRM2A
CMDENBL	UO 1	33
SYSRDY	UO 2	34
PROGRUN	UO 3	35
PAUSED	UO 4	36
HELD	UO 5	38
FAULT	UO 6	39
ATPERCH	UO 7	40
TPENBL	UO 8	41
BATALM	UO 9	43
BUSY	UO 10	44
ACK1/SNO1	UO 11	45
ACK2/SNO2	UO 12	46
ACK3/SNO3	UO 13	19
ACK4/SNO4	UO 14	20
SNO5/ACK5†	UO 15	21
SNO6/ACK6†	UO 16	22
SNO7/ACK7†	UO 17	24
SNO8/ACK8†	UO 18	25
SNACK	UO 19	26
RESERVED	UO 20	27

†Used for motion group 2 in a multiple motion group system.

Table 4–41. UOP Output Signals

UOP Output Signal	Description
CMDENBL UO[1]	This item is the command enable output. This output indicates that the robot is in a remote condition. This signal goes on when the remote switch is turned to ON. This output only stays on when the robot is not in a fault condition. When SYSRDY is OFF, CMDENBL is OFF. See Figure 4–14 and Figure 4–15 . This signal goes on when the following conditions are **all satisfied**. • Teach pendant disabled • Remote switch is ON • SFSPD input is ON • ENBL input is ON • $RMT_MASTER system variable is 0 • Not in single step mode • Mode selection switch is set to AUTO (when mode select switch is installed)
SYSRDY UO[2]	This output is the system ready output. This output indicates that the servo motors are turned on.
PROGRUN UO[3]	This item is the program run output. This output turns on when a program is running.
PAUSED UO[4]	This output is the paused program output. This output turns on when a program is paused.
HELD UO[5]	This output is the hold output. This output turns on when the SOP HOLD button has been pressed, or the UOP *HOLD input is OFF.
FAULT UO[6]	This output is the error output. This output turns on when a program is in an error condition.
ATPERCH UO[7]	This output is the at perch output. This output turns on when the robot reaches the predefined perch position. When $SHELL_WRK.$KAREL_UOP=FALSE, then the system sets $ATPERCH. The **ATPERCH position = Reference position #1** .
TPENBL UO[8]	This output is the teach pendant enable output. This output turns on when the teach pendant is on.
BATALM UO[9]	This output is the battery alarm output. This output turns on when the CMOS RAM battery voltage goes below 2.6 volts or if robot battery voltages are low.
BUSY UO[10]	This output is the processor busy output. This signal turns on when the robot is executing a program or when the processor is busy.
ACK 1-4 UO[11-14]	These items are the acknowledge signals output 1 through 4. These signals turn on when the corresponding RSR signal is received. See Figure 4–14 .

Table 4–41. UOP Output Signals (Cont'd)

UOP Output Signal	Description
SNO 1-8 UO[11] - UO[18]	These items are the signal number outputs. These signals carry the 8 bit representation of the corresponding PNS selected program number. If the program cannot be represented by an 8 bit number, the signal is set to all zeros or off.
SNACK UO[19]	This output is the signal number acknowledge output. This output is pulsed if the program is selected by PNS input. See Figure 4–15 .

4.10 HANDLINGTOOL CELL INTERFACE I/O SIGNALS

4.10.1 Overview

Cell Interface I/O signals are used for communicating between the robot controller and the cell controller (typically, a PLC). For instance, a cell controller can send a style number to the robot to run, get a signal to know if the robot override is at 100%, monitor if any of the digital inputs are simulated, and so forth.

The cell I/O menu contains two types of items. One type is the standard system-provided items. The second type is items that the user can define and use to communicate with the PLC or in his TP programs.

For example, in the tables below, Heartbeat is a system-provided digital output. The user can only assign a digital output to the heartbeat item. The robot controller will turn the assigned output on and off in an alternate fashion every 1 second (by default) when the robot is alive. An example of a user-defined item is "User out 1". In this case, you can assign it to an output. Then in the TP program, you can turn the digital output on or off as needed by using the system variable $CELLIO.$DO_UCFGN_I where n = 1 to 10. Users might find it beneficial when they are doing many similar projects where the I/O assignments are different but the TP program is the same. The TP program code might look like this:

```
1:  R[1]=$CELLIO.$DO_UCFG1_I ;
2:  DO[R[1]]=ON ;
```

The cell I/O menu should be set up after the user has completed the setup of the "Prog Select" menu under Setup. Depending upon the selections in the Prog Select menu, the items in the cell I/O menu can vary. For example, you will see the digital outputs for Style Ack only if you select STYLE as the Program Select mode in the "Prog Select" menu.

If you selected RSR or PNS as the Program select mode, refer to Section 3.2 for further details on assigning I/O.

This menu also allows you to view the status of the various signals and also to simulate them.

4.10.2 HandlingTool Cell Interface I/O Signals

4.10.2.1 Overview

This section contains input and output signal information that is specific to the HandlingTool application.

4.10.2.2 HandlingTool Cell Interface Input Signals

You can use the cell interface input screen to

- View the status of the input signals
- Simulate input signals
- Assign input signals

Table 4–42 lists and describes each cell interface input signal.

Use Procedure 4-12 to set up cell interface I/O.

Table 4–42. Cell Interface Input Signals

Input Signal	Description	SYSTEM VARIABLES
Tryout Mode Digital input	Enable this signal to signify that the robot will run in tryout mode. Then in the user TP programs, check the status of this signal to bypass part presence checking or skip signals you do not wish the program to process. (When used with the "MH Gripper option" or in SpotTool+, this input provides a way for the robot to execute material handling commands in a tryout, or dry run, mode. When this input is turned ON, the robot goes into tryout mode and operates all of material handling outputs but ignores all part present inputs. When this input is turned OFF, the robot is not in tryout mode. The cell output signal, MH Tryout, indicates the tryout status.) Refer to the “Material Handling Gripper Option” Chapter of the *HandlingTool Setup and Operations Manual* for detailed information. You can also put the robot in tryout mode by setting the system variable $CELL_SETUP.$DI_TRYOUT_I to TRUE. You can put the robot into tryout mode at the TEST CYCLE Spot Weld menu after you set up I/O.	$CELL_SETUP.$DI_TRYOUT_I

4.10.2.3 HandlingTool Cell Interface Output Signals

You can use the cell interface output screen to

- View the status of the output signals
- Simulate output signals
- Force output signals
- Assign output signals

Table 4–43 lists and describes each cell interface output signal.

Use Procedure 4-12 to set up cell interface I/O.

Table 4–43. Cell Interface Output Signals

Output Signal	Description	System Variables
Style Ack (Only for Style)	This output is used to tell the cell controller when to read the Style Request/Echo output.	$CELL_SETUP.$DO_STYSTR_I
Style Req./Echo (Only for Style)	This output indicates to the robot which group output is used to tell the cell controller which index into the style table was received.	$CELL_SETUP.$GO_STYSEL_I
Input Simulated	If any of the digital inputs are simulated, this output will automatically turn on.	$INPT_SIM_DO
OVERRIDE = 100	If the robot general override is 100%, this output turns on, else off.	$GENOVRD_DO
Robot ready	When the controller is ready to run a program, clear of all faults such as TP enable or machine lock, this output will turn on. This output is used to notify the PLC whether the CMDENBL, SYSRDY, and other conditions (whether each group operation is enabled, whether welding is enabled, and user-specified DI/O and RI/O conditions) are acceptable for a production start. The conditions contained in the signal must be previously set at the STATUS Robot Ready screen. You can check the status of each condition at this status screen.	
Tryout Status	When the Tryout mode input signal (See Table 4–42) is on, this output turns on. It can be used to signify a tryout or test run mode of operation. The tryout mode signal can come from a PLC or it can be simulated. Users can check the status of tryout in their TP programs by checking the value of the system variable in the right. When used with the "MH Gripper" option, the tryout status ON means that the gripper operations are going to be performed without part presence checking or vacuum feedback.	$CELL_SETUP.$DO_TRYOUT_I
Heartbeat Digital output	This signal alternates between the ON and OFF state every Heartbeat timing. The signal is intended to verify the link between the robot and cell controller.	$CELL_SETUP.$DO_HEARTB_I

Table 4–43. Cell Interface Output Signals (Cont'd)

Output Signal	Description	System Variables
MH Fault	Reserved for used with the "MH Gripper" option and in SpotTool+ with the MH plug-in. Refer to Section 13.5.3 for further details. With the MH gripper option, this output is turned on when a vacuum feedback check fails or part presence is not detected during pickup, and so forth.	$CELL_SETUP.$DO_HFAULT_I
MH Alert	Reserved for use with the "MH RS4 compliance" option. Refer to Section 13.5.3 for further details. With the MH gripper or the MH RS4 compliance option, it is possible to disable a fault for 20 or fewer cycles. During the time that a fault is disabled, this output turns on	$CELL_SETUP.$DO_HALERT_I
Refpos1 **n** = 1 to 3 Digital output	This indicates which output is used to indicate when the robot arm is at motion group 1's reference position ***n*** (if the reference position is enabled.)	$REFPOS1[***n***].$DOUT_IN DX ***n*** = 1 to 3
User out ***n*** Digital output	These outputs are intended for use in user teach pendant or KAREL programs and are in the Cell I/O screen for the convenience of setting and monitoring. The index numbers of these outputs, set in the CELL I/O screen, can be accessed in the system variables.	$CELLIO.$DO_UCFGn_I **n** = 1 to 5
User GOUT ***n*** Group output	These outputs are intended for use $CELLIO.$GO_UCFGn_Iin user teach pendant or KAREL programs and are in the Cell I/O screen for the convenience of setting and monitoring. The index numbers of these outputs, set in the CELL I/O screen, can be accessed in the system variables.	$CELLIO.$GO_UCFGn_I **n** = 1 to 10

4.10.2.4 Setting Up HandlingTool Cell Interface I/O Signals

Procedure 4-12 Setting Up Cell Interface I/O

Note This procedure contains information about setting up the cell interface only. For information about how to configure, force, verify and simulate digital signals, refer to Section 4.2 .

Steps

1. Press MENUS.
2. Press I/O.
3. Press F1, [TYPE].
4. Select Cell Intface. You will see either the cell input or output screen.

   Note When you enter a signal number for a signal in these I/O screens, the comment for the corresponding signal in the I/O Digital or I/O Group screen will be updated with the name of the signal displayed here.

   To change between the display of the input and output screens, press F3, IN/OUT.

   To move quickly through the information, press and hold the SHIFT key and press the down or up arrow keys.
5. Move the cursor to the input or output you want to change.
6. Type the number and press ENTER.
7. **To simulate or unsimulate I/O**, place the cursor in the SIM column next to the I/O that you want to change.

   - **To simulate the signal**, press F4, SIM. The signal will be simulated.
   - **To unsimulate the signal**, press F5, UNSIM. The signal will be unsimulated.

   Note You must first simulate any spot welding input that you want to force.

   Warning

   Any spot welding output that is forced on and is not simulated actually turns equipment ON or OFF. Make sure all personnel and unnecessary equipment are out of the workcell and that all safeguards are in place before you force an output; otherwise, you could injure personnel or damage equipment.
8. **To force an output on or off**, place the cursor in the STATUS column next to the I/O that you want to change.

   - **To force the I/O signal on**, press F4, ON.
   - **To force the I/O signal off**, press F5, OFF.
9. **To specify detail information about a signal**, press F2, CONFIG. You will see a screen similar to the following.

```
I/O Cell Outputs
 Output Signal Details
  1 Signal name: USER OUT 2
  2 Output type/no: DOUT [  0]
```

- **To add a comment**, move the cursor to the signal name, press the appropriate function keys to type the comment, and press ENTER.
- **To change the port type**, move the cursor to Input (or Output) type/no., press F4, [CHOICE], select the type of I/O you want, and press ENTER.
- **To change the port number**for any type of I/O except UOP, move the cursor to Input (or Output) type/no., type the port index number, and press ENTER.
- **To check whether the assignment is valid**, press F5, VERIFY.
  - — If the assignment is valid, the message, "Port assignment is valid," is displayed.
  - — If the assignment is not valid, the message, "Port assignment is invalid," is displayed.
- **To display detail information for the previous I/O signal**, press F2, PREV_IO.
- **To display detail information for the next I/O signal**, press F3, NEXT_IO.

4.10.2.5 Robot Ready (Option)

The cell output signal, **Robot Ready,** notifies the cell controller whether the robot is in a production start ready status.

The Robot ready output signal does not turn ON unless the status of the various check items listed in Table 4–44 is OK. The check items are performed with Robot ready on the status screen. It is necessary to set monitor items in advance in accordance with the actual system.

Table 4–44. Check Items with Robot Ready

Check item	Explanation
CMENABLE	Indicates whether the CMENABLE-related items are OK or NG. NG appears if UI allocation is not correct.
SYSRDY	Indicates whether the SYSRDY-related items are OK or NG. NG appears if UI allocation is not correct.

Table 4–44. Check Items with Robot Ready (Cont'd)

Check item	Explanation
General	Indicates whether the robot operation, welding enable/disable, pressurization enable/disable, and other items are OK or NG. To set the items and check whether they are enabled or disabled, press F2 [CONFIG] to display the display screen.
User	Indicates the status of the I/O signals necessary for a production start, such as DI/DO and RI/RO. NG appears if UI allocation is not correct. To set the items and check whether they are enabled or disabled, press F2 [CONFIG] to display the display screen.

Procedure 4-13 Displaying the Robot Ready Status Check Screen

1. Press MENUS.
2. Select STATUS.
3. Press F1, TYPE, and select Robot Ready. You will see a screen similar to the following:

```
Output: Robot ready DO[0]
1 CMDENBL TP disabled NG
2 CMDENBL SI[2] = ON OK
3 CMDENBL SFSPD = ON NG
4 CMDENBL ENBL = ON OK
5 CMDENBL $RMT_MASTER=0 OK
6 CMDENBL SYSRDY=ON NG
7 CMDENBL No active alarms NG
8 CMDENBL Not in single step OK
9 SYSRDY ENBL = ON NG
10 SYSRDY GRP1 Servo ready OK
11 SYSRDY GRP2 Servo ready OK
12 GRP1 Motion enabled NG
13 GRP2 Motion enabled NG
14 DO[0: ] --
15 DO[0: ] --
16 DO[0: ]
```

4. If any settings have been changed, press NEXT, then press F1, REDO, to check the current status.
5. **To change a monitor item** (CMDENBL and SYSRDY conditions cannot be changed)
   a. Move the cursor to the item you want to change and press F2, CONFIG.
   b. To monitor the item, select YES. Otherwise, select NO.
   c. To return to the list, press F2 LIST.

6. To add monitor signals

a. Move the cursor to a DO[0:] field and press F2, CONFIG.

b. Change the SIGNAL type, number, OK condition (ON, OFF), as desired.

c. To monitor the item, select YES. Otherwise, select NO.

d. To return to the list, press F2, LIST.

Chapter 5

PLANNING AND CREATING A PROGRAM

Contents

Chapter 5	PLANNING AND CREATING A PROGRAM	5–1
5.1	OVERVIEW	5–3
5.2	PLANNING A PROGRAM	5–3
5.2.1	Overview	5–3
5.2.2	Motion	5–3
5.2.3	Predefined Positions	5–7
5.3	WRITING AND MODIFYING A PROGRAM	5–15
5.3.1	Overview	5–15
5.3.2	Writing a New Program	5–17
5.3.3	Creating and Writing a New HandlingTool Program	5–22
5.3.4	Modifying a Program	5–28
5.3.5	Setting Up Predefined Positions and Using Them in a Program	5–45
5.3.6	Creating a Macro Program	5–47
5.4	MODIFYING A PROGRAM IN THE BACKGROUND (BACKGROUND EDITING)	5–48
5.4.1	Overview	5–48
5.4.2	Background Edit Process Flow	5–49
5.4.3	Background Editing Using the iPendant	5–51
5.4.4	Modifying a Program in the Background	5–52
5.4.5	Troubleshooting Background Edit	5–54
5.5	SETTING PROGRAM STORAGE CONFIGURATION	5–57
5.5.1	User Interface	5–60
5.5.2	Maximum Program Size	5–62
5.5.3	KAREL Control of STORAGE	5–63
5.5.4	File Formats	5–63
5.5.5	Storage When Loading from a File	5–64
5.5.6	Backup and restore	5–64
5.5.7	Issues relating to SHADOW programs	5–65
5.6	ICON EDITOR	5–65

5.6.1 Using the Icon Editor 5–66
5.6.2 Configuring the Custom Toolbar 5–71

5.1 OVERVIEW

A *FANUC Robotics application program* includes a series of commands, called *instructions*, that tell the robot and other equipment how to move and perform a task. For example, a program directs the robot and controller to:

- Move the robot in an appropriate way to required locations in the workcell.
- Perform an application-specific operation
- Send output signals to other equipment in the workcell.
- Recognize and respond to input signals from other equipment in the workcell.
- Keep track of time, part count, or job number.

5.2 PLANNING A PROGRAM

5.2.1 Overview

This section provides hints that will help you program specific tasks more efficiently. It contains hints for programming

- Motion
- Predefined positions
- Application program guidelines

Note The hints in this section are for programming, not jogging. World frame or user frame are usually best for jogging and recording positions.

5.2.2 Motion

5.2.2.1 Overview

Use the guidelines in this section to help you program certain kinds of robot motion.

Joint Motion

Use joint motion type for the fastest moves and the shortest cycle time.

Linear motion type results in slower moves. Use linear motion type when it is the only way to move to a certain position.

Arc welding is performed using linear and circular motion instructions. Moves between weld paths are generally joint moves.

When running DispenseTool, use linear motion type when the gun is dispensing or moving in a confined area.

When running PaintTool, joint motions are not allowed for 6-axis tracking applications.

FINE Motion

When running ArcTool or SpotTool+, use FINE termination type for the beginning and end of a weld position. FINE termination type positions the robot at the precise point where welding must be done. If you use continuous, the welds will not start or finish exactly at the programmed positions.

When running PaintTool, use fine termination type only at the end of a job. Fine stops the robot precisely at the programmed point.

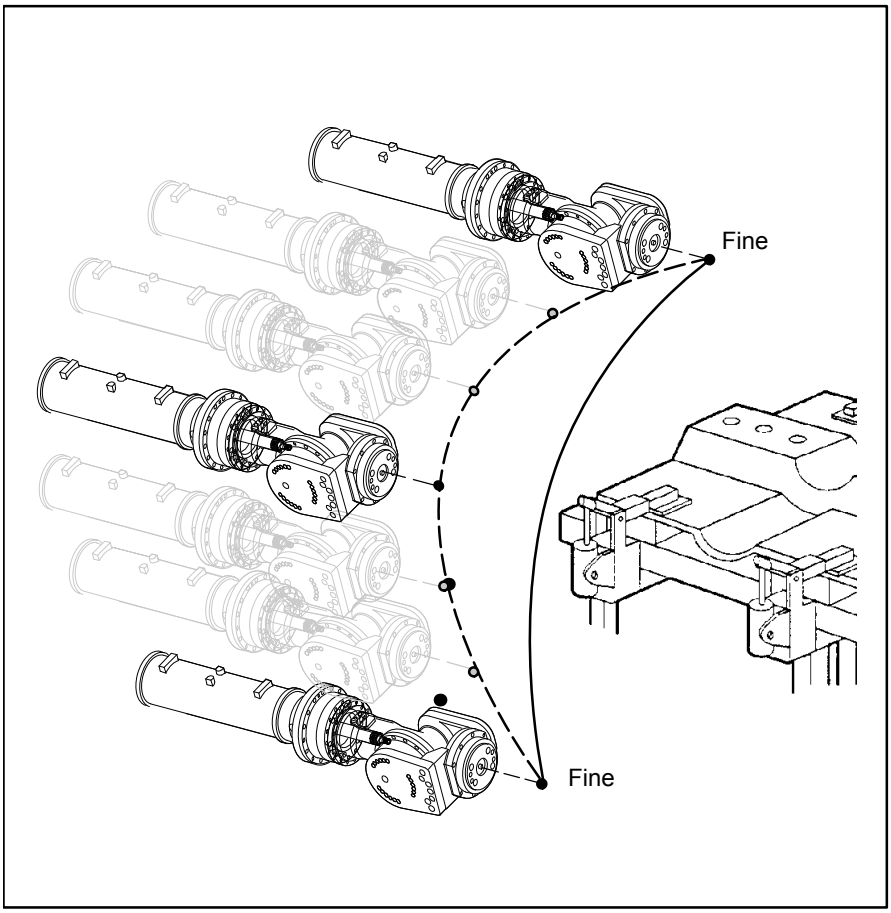
Continuous Motion

Use continuous termination type for the most efficient movement around obstacles. Program the position near the obstacle and then adjust the termination type value as needed.

Use continuous termination type to blend arc welding motions smoothly or to achieve the most efficient dispensing or painting process.

See Figure 5–1 .

Figure 5–1. Continuous Termination Type for Movement Around Obstacles



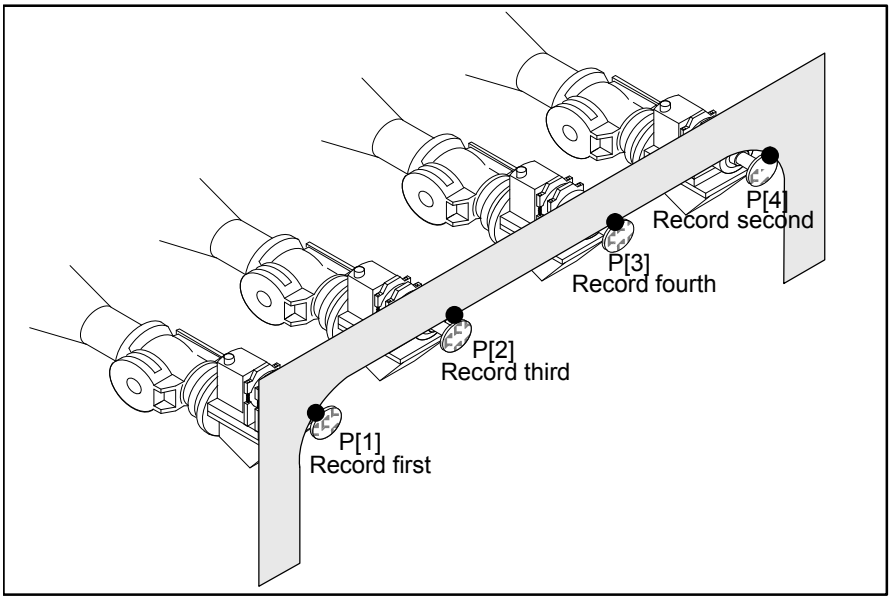
5.2.2.2 Minimize Changes in Wrist Orientation

Abrupt changes in wrist orientation can sometimes increase cycle time. Smooth, gradual changes are faster and more efficient. Plan motions to minimize wrist orientation changes when possible.

When wrist orientation changes are necessary, distribute the rotation of the wrist smoothly throughout several moves. Do not force the robot to change the entire wrist orientation at the end point of a series of moves.

See Figure 5–2 .

Figure 5–2. Minimizing Wrist Orientation Changes



To achieve maximum smoothness when you change wrist orientation,

1. Move the robot to the first position (P[1]). Be sure that the orientation of the wrist is correct for that position.
2. Add a motion instruction to record the position of the robot.
3. Jog the robot to the last position that will be recorded. Be sure the robot is in the correct orientation for this position.
4. Add a motion instruction to record the position of the robot.
5. Record the number of motion instructions you think are necessary between the first position and the last position.
6. At a slow speed, step forward toward the P[1] position. Refer to the "Testing a Program and Running Production" chapter in this manual for more information on single step motion.
7. At a slow speed, step forward toward the second position, or P[2].
8. **Before the robot reaches P[2],** release the SHIFT or FWD key. Record this new position between P[1] and P[2].
9. Repeat 7 and 8 for all the positions between the first position and the last position.

Caution

If, while you smooth the motion, axes 4, 5, and 6 attempt to reach zero degrees and axis 5 flips orientation, change the jog coordinate system to JOINT and continue jogging.

5.2.3 Predefined Positions

5.2.3.1 Overview

You can use *predefined positions* in a program. A predefined position is a position you define that can be used several times in a program or in other programs. For example, predefined positions might include a position for maintenance, a perch position, or a "safe" position.

An output signal can also be used to indicate to the system that the robot has reached the predefined position.

You can define a predefined position in the following ways:

- Macros - These use position registers that contain the predefined position.
- Programs - These also use position registers that contain the predefined position.
- Position registers
- Reference Positions - Refer to Chapter 10 *ADVANCED FUNCTIONS* for information about reference positions.

5.2.3.2 Macros

Macros are programs that can be executed from:

- Specific teach pendant keys
- The MACROS screen
- Specific operator panel buttons
- The MANUAL FCTNS macro screen
- Within a program using the MACRO COMMAND instruction

Macros allow you flexibility to create your own teach pendant instruction. For example, you could specify a macro from the MACROS screen to move the robot to the home position automatically.

Macros also allow you flexibility as to how and when the robot moves to the predefined position. For example, you could specify that when DI[1] = ON, the robot moves to the home position. You could turn DI[1] ON at the start of each production cycle or shift by attaching it to an external pushbutton so the robot begins production at a known position.

Macro positions are defined in the specific program and can be adjusted only if they are changed in the program using TOUCHUP.

Refer to Chapter 3 *GENERAL SETUP* for information about setting up and executing macros. Refer to Section 6.17 for more information about the macro command instruction.

5.2.3.3 Programs

You can write a program that moves the robot to a predefined position. You can use the macro command or CALL program instruction to branch to the macro or program that moves the robot to the predefined position.

Refer to Section 6.9.3 for more information. You can also include the position register used as a "predefined position" within the program.

With ArcTool and HandlingTool, program positions are defined in the specific program and can be adjusted only if they are changed in the program using TOUCHUP.

In DispenseTool, you must teach the predefined positions in special programs named MOV_. You then access the positions you defined using the CALL PROGRAM instruction in your program. Section 5.2.3.4 details how to define and use predefined positions. Refer to Chapter 7 *TESTING A PROGRAM AND RUNNING PRODUCTION* for information on using the Move Menu to move to predefined positions.

The predefined programs used in DispenseTool are:

- MOV_HOME
- MOV_REPR
- MOV_PURG

5.2.3.4 Position Registers

Position registers can be used as predefined positions. Each position register can only contain one robot position.

Refer to Section 6.6 for more information about position registers.



Caution

Position registers and reference positions are global positions. Any changes to position registers that are used as predefined positions in a program will change the predefined position location. Make sure you do not change position registers used as predefined positions; otherwise, unexpected results might occur.

To set up a position register as a predefined position,

1. Press DATA.
2. Press F1, [TYPE].
3. Select Position Reg.
4. Move the cursor to the position register you want to set up.
5. Jog the robot to the position you want to define.

   Note In DispenseTool, you will be jogging the robot to the position you want to predefine.

6. Hold down the SHIFT key and press F3, RECORD.

To use a position register as a predefined position in a program, include the position register you set up in a motion instruction. For example, if you selected PR[2] in Step 4, you could use the following motion instruction in your program.

```
L

PR[2]


50mm/sec FINE
```

See Table 5–1 for a list of which predefined positions are used by which application.

Table 5–1. Predefined Positions Used by Each Tool

	ArcTool	DispenseTool	HandlingTool	SpotTool+
Home or Perch	O	O	O	O
Repair	O	O	O	O
Purge		O		
Pounce		O		O
Safe			O	

Caution

Recorded positions and position registers are affected by UFRAME, and UFRAME has an affect during playback. If you change UFRAME, any recorded positions and position registers will also change.

Note If the position register is to be shared between two programs, both programs must have the same tool frame and user frame in order to move the robot to the same position in space.

Home Position

The *home position* is a position away from the workpiece transfer area. Program the robot to move to home before the first position, between cycles, and any time the robot must be away from workcell activity.

In DispenseTool, the home position is also called the perch position. The program MOV_HOME is used to move the robot to the home position. The home position is defined as the last position recorded in the MOV_HOME program at the last cold start. The AT PERCH UOP output turns on when the robot is near the last position recorded in the MOV_HOME program. You must define the MOV_HOME program before you perform any calibration procedures.

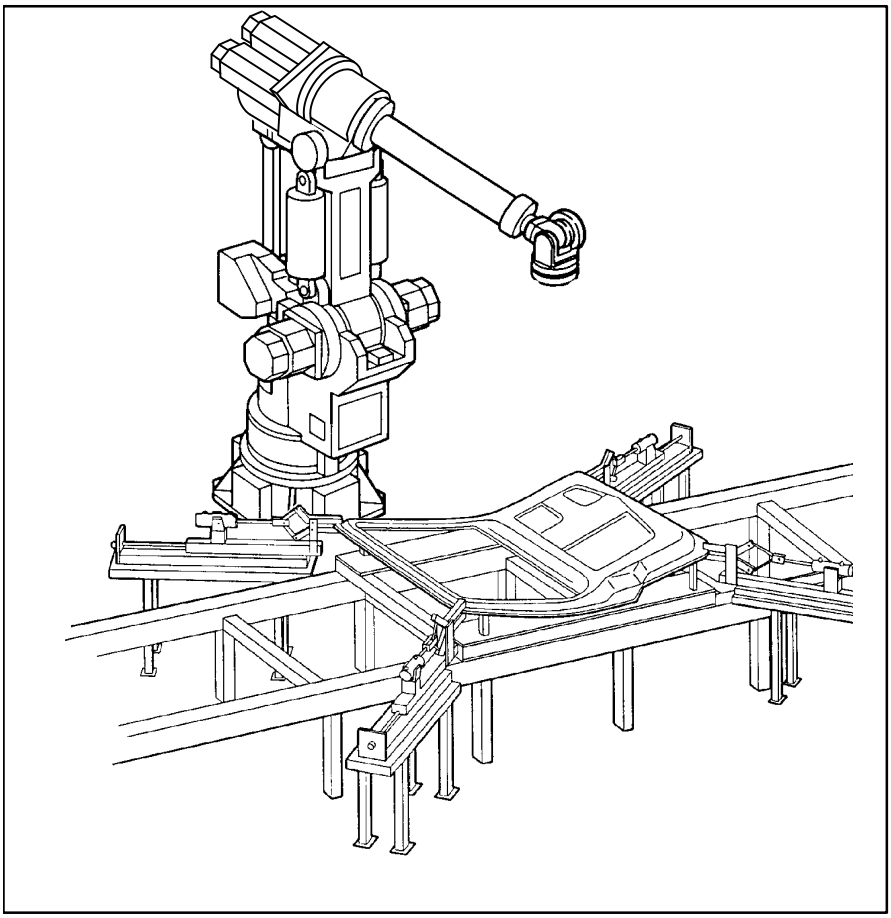
For fastest cycle time, record the home position close to the first position in the path, in line with the first position in the path, and with the same wrist orientation as the first position. Also, the MOVE TO HOME macro can be used to move the robot to the home position.

Note Before you can run a program in production within SpotTool+, the robot must be at the home position. You must set up a reference position to be the home position and make sure the robot is at the home position before you run production.

In HandlingTool, "Home Position" is an input signal in UOP, not a predefined position. It can be one of the Reference Positions, but these are not utilities or built-ins used to send the robot to the "HOME" position.

Figure 5–3 shows an example of a home position.

Figure 5–3. Home Position



Repair Position

The *repair position* is a position where robot repair operations are performed. Program the robot to move to the repair position any time repair operations must be performed. Record the repair position away from other equipment and the transfer area.

In DispenseTool, the program MOV_REPR is used to move the robot to the repair position. The repair position is defined as the last position recorded in the MOV_REPR program. The AT SERVICE digital output signal turns ON when the robot is near the last position recorded in the MOV_REPR program. Refer to Section 5.3 for more information.

In SpotTool+, use the predefined MOV_REPR.TP macro program to safely move from HOME to REPAIR. The last position in MOV_REPR.TP is the repair position. Refer to Section 3.3 for information on setting up macros.

Table 5–2 shows an example of a repair position.

In HandlingTool, the "Repair Position" is not a predefined position, but you can use one of the *reference positions* for this purpose. A user program with the same trained position must be run to move the robot to "REPAIR" position

Table 5–2. Rack 1

Slot	1	2	3	4	5
Module Type	UOPINDIN	DIN	DIN(AccuChop (not with LR PaintTool) or MCP Option)	UOPOUT DOUT	DOUT

Purge Position

The *purge position* is located over a place where material can be dispensed freely, such as a purge bucket or waste sealant bin. The robot uses this position to dispense material for calibrations or to clear the dispenser hoses of material.

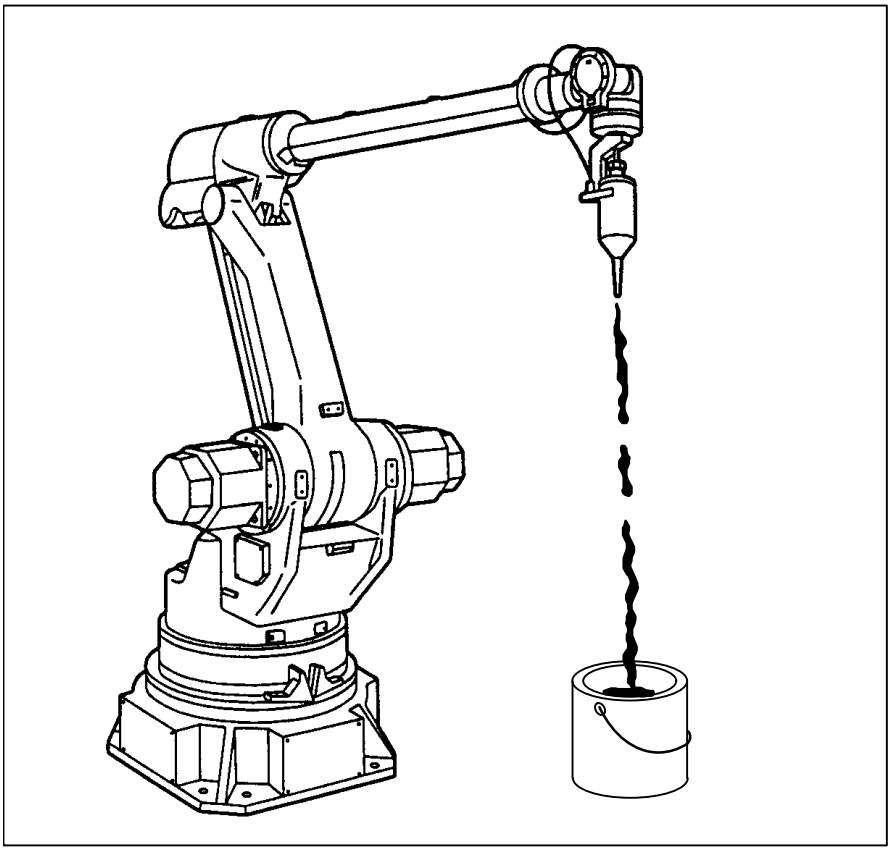
Note When using SpotTool+ with dispense plug-in and multi-group motion all programs that use motion while dispensing are required to use group mask [1,*,*,*,*].

Note For system level macro programs, the group mask cannot be changed. Refer to Section 3.3 for more information on macro commands.

The program, MOV_PURG, is used to move the robot to the purge position. The purge position is defined as the last position recorded in the MOV_PURG program. The AT PURGE digital output, if configured, turns ON when the robot reaches the last position in this program. You must define the MOV_PURG program before you perform calibration. Refer to “Defining Predefined Positions in DispenseTool Programs” in the *DispenseTool Setup and Operations Manual* for more information.

Figure 5–4 shows an example of a purge position.

Figure 5–4. Purge Position



Pounce Position

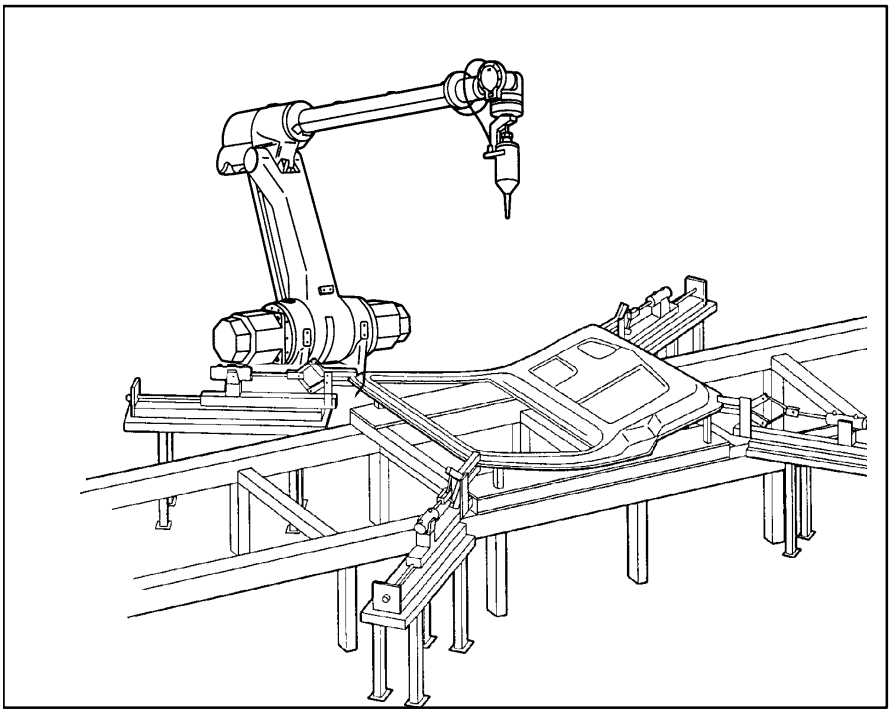
The *pounce position* is a position closer to the workpiece than the home position, but not at position where dispensing begins. Using a pounce position can improve cycle time slightly, if the workcell is set up so that it is safe for the robot to move closer to the workpiece when the workpiece is almost in place.

The pounce position should be set up for each job, in a position appropriate for that job. After reaching the pounce position, the robot should wait for a "go ahead" signal from the cell controller, informing the robot that the workpiece is now completely in position and the job can proceed.

Note No special provisions are made in DispenseTool for a pounce position. The usage and support of a pounce position must be handled as needed.

Figure 5–5 shows an example of a pounce position.

Figure 5–5. Pounce Position



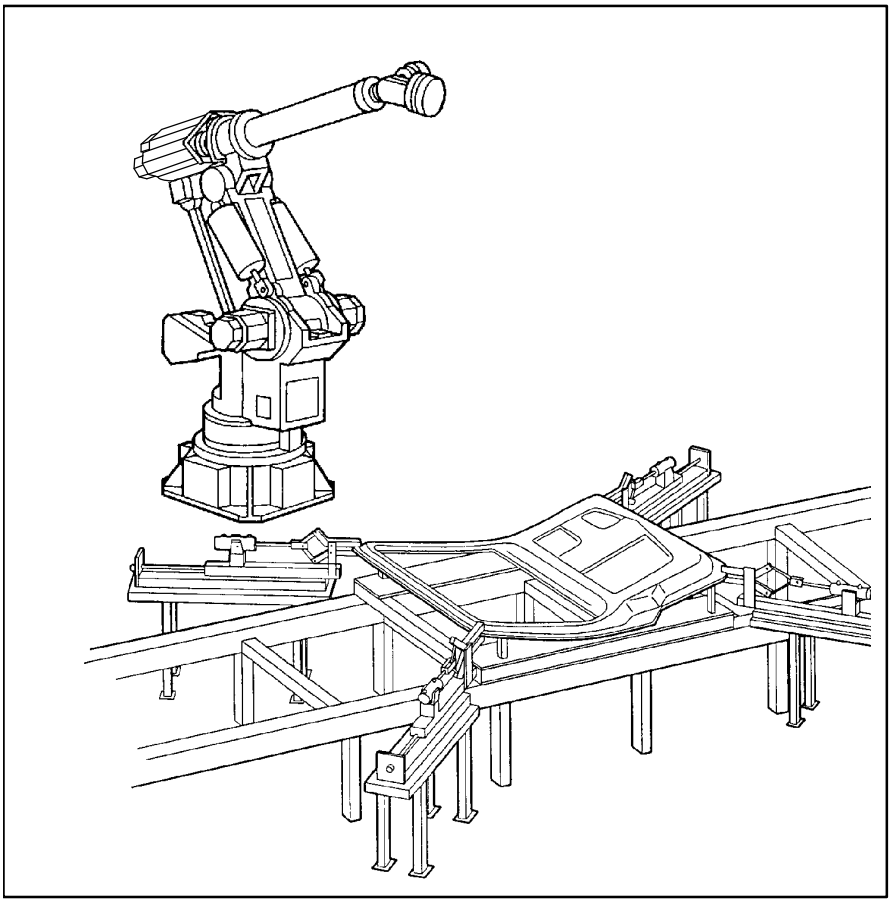
Safe Position

The *safe position* is away from fixtures and the workpiece transfer area. Program the robot to move to the safe position any time it is necessary to move the robot away from other workcell activities.

In HandlingTool, the "Safe Position" is not a predefined position, but you can use one of the *reference positions* for this purpose. A user program with the same trained position must be run to move the robot to "SAFE" position.

Figure 5–6 shows an example of a safe position.

Figure 5–6. Safe Position



Other Positions

You can define any other positions to be predefined positions. Define any position that the program uses more than once as a predefined position. This minimizes the time it takes to create and modify your program.

5.3 WRITING AND MODIFYING A PROGRAM

5.3.1 Overview

You can write new programs and modify existing programs to direct the robot to perform a task. Writing a program includes:

- Naming the program

- Defining default instructions
- Adding instructions to the program

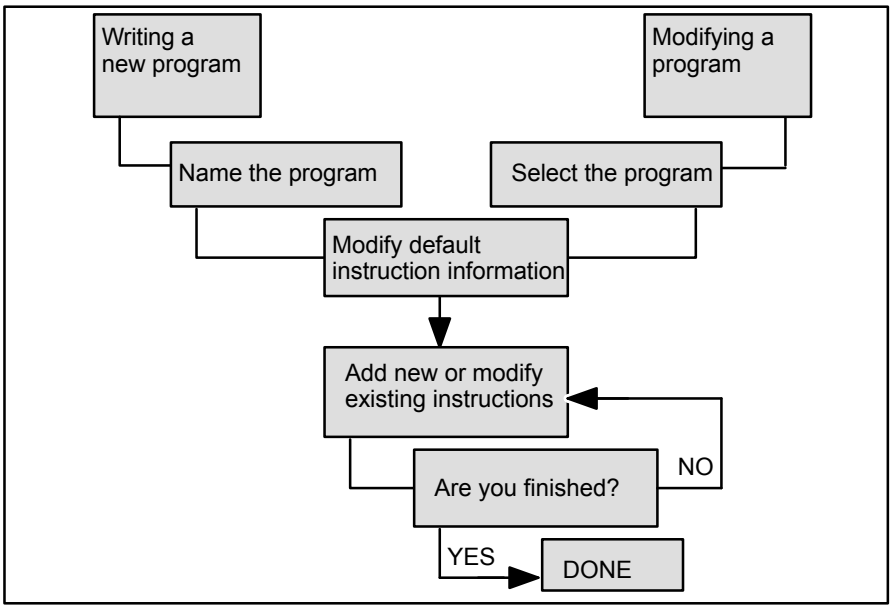
Modifying a program includes:

- Selecting the program
- Modifying default instructions
- Inserting instructions
- Deleting instructions
- Copying and pasting instructions
- Searching for instructions
- Renumbering instructions
- Undoing operations
- Displaying comments

In DispenseTool, defining predefined positions requires you to create and write a program directing the robot to those positions.

Figure 5–7 summarizes writing and modifying a program.

Figure 5–7. Writing and Modifying a Program



5.3.2 Writing a New Program

When you write a new program you must

- **Name** the program and set program header information. The program header information is a reserved area in the program that provides the controller with specific program characteristics.
- In PaintTool, **Define** the detailed information for the program (refer to Chapter 6 *PROGRAM ELEMENTS*).
  - — Whether it is a job or process
  - — All paint related data
- **Modify** default instruction information. This includes modifying motion instructions and application-specific instructions.
- **Add** motion instructions to the program.
- **Add** application-specific instructions and other instructions to the program.

Naming the Program

You can name a program using three different methods (refer to Chapter 6 *PROGRAM ELEMENTS* , for additional information):

- **Words** - This method provides a list of words you can use to build a program name.

Application	Words
ArcTool	• RSR • PNS • MAIN • SUB • TEST

Application	Words
DispenseTool	• TEST • JOB • STYLE • PROG • ALT
HandlingTool	• RSR • PNS • MAIN • SUB • TEST
PaintTool	• JOB • PROC • TEST • MM_
SpotTool+	• TEST • JOB • STYLE • PROG • ALT

- **Upper Case** - This method lets you use upper case letters and any numbers.
- **Lower Case** - This method lets you use lower case letters and any numbers. For the program name, lower case letters are automatically converted to upper case after you enter them.

Options allows you to change whether you are overwriting, inserting, or clearing the program name or comment information. The screen will display either Insert or Overwrite. Clear allows you to remove text from the current field. Space inserts or overwrites a space into the text.

The total length of the program name must be no more than eight characters. You can combine words, upper case letters, and lower case letters to form the program name. Give the program a unique name that indicates the purpose of the program.

Note Do not use the asterisk * symbol in program names.

Note Any program name that is longer than 12 characters is abbreviated on the SELECT screen and displayed with a > at the end. When the program name is selected, the entire program name is displayed at the bottom of the screen.

Note If you have common characters in long program names, you can set $UI_CONFIG.$PROG_COMMON[1–3] to contain the common characters. For instance, if your program name is TEST111_PLANT1_ROBOT15467_A and TEST111 is a common prefix, then set $UI_CONFIG.$PROG_COMMON = TEST111. When the system needs to truncate the program name it will remove the common portion and replace it with a tilde (~) character. See the following screen for an example.

```
Select
                 702566 bytes free   7/7
No.    Program name           Comment
 1   -BCKEDT-       [                  ]
 2   HOME_IO        [SET AT HOME I-O   ]
 3   TEST1          [TEST1             ]
 4   ~_PLANT1_ROBOT>[                  ]
 5   UOP_ROBOT_12>  [                  ]


TEST111_PLANT1_ROBOT15467_A
```

Defining Detail Information

The detail of program header information includes:

- Creation date
- Modification date
- Copy Source
- Number of positions and program size
- Program Name
- Sub Type
- Comment

- Group mask
- Write protection
- Ignore pause

In ArcTool, the detail of program header information also includes:

- Equipment Mask (Multi-Equipment option)
- Weld Weave Schedules (MP/RPM option)

In DispenseTool, the detail of program header information also includes:

- Cycle time
- Last cycle time
- Gun on time
- Last gun on time
- Material volume
- Last material volume
- Default user frame
- Default tool frame
- Part ID, for some dispensers
- Equipment number

In PaintTool, the detail of program header information can also include:

- Default user frame: (Process sub type)
- Default tool frame: (Process sub type)
- Line track schedule number: (Process sub type with tracking option)
- Line track boundary number: (Process sub type with tracking option)
- Continue track at program end: (Process sub type with tracking option)
- Opener program 1: (Job sub type with opener option)
- Last Cycle Time (Display only)
- Last Gun On Time (Display only)
- Group 2 (right) Job (DualArm PaintTool option)

In the SpotTool+ with DispenseTool Plug-in software, the detail of program header information also includes:

- Application Mask
- Cycle time
- Last cycle time
- Gun on time
- Last gun on time
- Material volume
- Last material volume
- Default user frame
- Default tool frame
- Part ID, for some dispensers
- Equipment number

Refer to Section 6.2 for details about program header information.

Defining Default Instruction Information

Motion instructions tell the robot to move to an area in the workcell in a specific way. When you create a program you can define, in advance, the way you want the robot to move when you add a motion instruction. You do this by defining default motion instruction information. Default motion instructions can include application-specific instructions to perform your work, as well as other motion options.

After you have defined the default instructions you can add them to the program. You select one of the available default instructions to be the current default instruction by moving the cursor to that instruction. You can define and change default instructions any time while you write or modify a program.

Modifying Default Motion Instructions by Adding Option Fields (adding instructions)

You can also add other instructions not included in the default motion instruction to your program. To add these instructions, select the kind of instruction you want to add to the program and use the information on the screen to enter specific instruction information.

You add all instructions using the same general procedure. Motion instructions, however, require some specific information. Refer to Procedure 5-2 for information on adding motion and other kinds of instructions.

Adding Instructions

You can also add other instructions not included in the default motion instruction to your program. To add these instructions, select the kind of instruction you want to add to the program and use the information on the screen to enter specific instruction information.

5.3.3 Creating and Writing a New HandlingTool Program

To create and write a new application program, refer to Procedure 5-1 .

Procedure 5-1 Creating and Writing a New HandlingTool Program

Note If you want to edit a program with the teach pendant off, this is called Background Program Editing. **Refer to Section 5.4 for more information.**

Conditions

- All personnel and unnecessary equipment are out of the workcell.

Steps

1. **Set the User frame number:**
   a. Press MENUS.
   b. Select SETUP.
   c. Press F1, [TYPE].
   d. Select Frames.
   e. If user frames are not displayed press F3, [OTHER], and select User Frame. If F3, [OTHER], is not displayed, press PREV.
   f. To select the user frame to use, press F5, SETIND, type the number of the user frame you want, and press ENTER. This sets the active user frame ($MNUFRAMNUM[1]) to the number of the frame you specify.
2. **Name the program:**
   a. Continuously press the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.
   b. Press SELECT. You will see a screen similar to the following.

```
Select
               702566 bytes free   7/7
No.  Program name           Comment
 1   -BCKEDT-         [                   ]
 2   HOME_IO          [SET AT HOME I-O    ]
 4   TEST1            [TEST1              ]
 5   UOP_ROBOT_12>    [                   ]




UOP_ROBOT_12436
```

c. If F2, CREATE, is not displayed, press NEXT, >.

d. Press F2, CREATE. You will see a screen similar to the following.

```
  --- Create Teach Pendant Program ---
Program Name
 ▌
                              -- End --




Enter program name
```

e. Move the cursor to a method of naming the program. The function key labels will change depending on the naming method you choose.

f. Type the program name:

Note As you move the cursor through the list of naming methods, the labels on the function keys will change.

- Press the function keys whose labels correspond to the name you want to give to the program. These labels vary depending on the naming method you chose in Step 2e .

  For example, if you chose Upper Case, press a function key corresponding to the first letter. Press that key until the letter you want is displayed in the program name field. Press the right arrow key to move the cursor to the next space. Continue until the entire program name is displayed.

  Note You can also use the numbers on the teach pendant to include numbers in your program name.

  To delete a character to the left of the cursor, press BACK SPACE.

- When you are finished, press ENTER. You will see a screen similar to the following.

```
  --- Create Teach Pendant  Program ---
Program Name
 TEST111_PLANT1_ROBOT15467_A
                              -- End --


Select function
```

3. To view and modify program header information:

a. Press F2, DETAIL. You will see a screen similar to the following.

Note Your program name will vary, depending on what application you are using and what method you used for creating the program name.

```
Program detail
Creation Date:              02-Jan-xxxx
Modification Date:          02-Jan-xxxx
Copy Source:
Positions:   10  Size:       312 Byte
 Program Name:
 1 TEST111_PLANT1_ROBOT15467_A
 2 Sub Type:           [NONE           ]
 3 Comment:            [               ]
 4 Group Mask:         [1,*,*,*,*.*.*,*]
 5 Write protect:          [ON       ]
 6 Ignore pause:           [OFF      ]
```

b. Review the settings on the screen.

- If you like the settings that are displayed on the screen and want to skip setting program header information and begin editing the program. press F1, END, and skip to **Defining Default Motion Instructions** in this procedure.
- If you want to change the settings, continue with Step 4 .

4. To change the program:

a. Move the cursor to the program name and press ENTER.

b. Move the cursor to select a method of naming the program.

c. Press the function keys whose labels correspond to the name you want to give to the program. These labels vary depending on the naming method you chose in Step 4b .**To delete a character**, move the cursor to the right of the character and press BACK SPACE.

d. When you are finished, press ENTER.

Note You cannot modify details if the program is set up as a system level macro. These macros are identified with the letter "s" at the far right side of the macro setup screen.

5. To change the sub type:

a. Move the cursor to the sub type and press F4, [CHOICE]. You will see a screen similar to the following. Refer to Section 6.2.7 for more information on sub types.

```
1 None
2 Macro
3 Cond
```

b. Select a sub type and press ENTER.

Note If $JOBPROC_ENB = TRUE, you can select the Job and Process sub types also.

6. To type a comment:

a. Move the cursor to Comment and press ENTER.

b. Select a method of naming the comment.

c. Press the appropriate function keys to add the comment.

d. When you are finished, press ENTER.

For example, if you chose Upper Case, press a function key corresponding to the first letter. Press that key until the letter you want is displayed in the comment field. Press the right arrow key to move the cursor to the next space. Continue until the entire comment is displayed.

To delete a character to the left of the cursor, press BACK SPACE.

7. To set the group mask (or motion group):

Note If your system is not set up for multiple groups, you will only be able to select a 1, for the first group, or a *, for no group.

a. Move the cursor to the group you want to enable or disable. You can use multiple groups in a single program, but only two groups can perform Cartesian motion within a single program. The first position in the group mask corresponds to the first group. Groups 1, 2, 3, 4, 5, 6, 7, and 8 are supported.

b. **If you want to enable a group**, scroll right to the group you want and press F4, 1, for each enabled group.

c. **If you want to disable a group**, scroll right to the group you want and press F5, *. If you disable all groups, you cannot add motion instructions to your program.

Note After the group mask has been set, and motion instructions have been added to the program, the group mask cannot be changed for that program.

8. To set write protection:

a. Move the cursor to Write protect. Refer to Section 6.2.10 for information on write protect.

b. If you want to turn write protection on, press F4, ON.

c. If you want to turn write protection off, press F5, OFF.

Note Write protection must be set to OFF to create a program.

9. To set ignore pause:

a. Move the cursor to Ignore pause. Refer to Section 6.2.11 for information on ignore pause.

b. If you want to turn on ignore pause, press F4, ON.

c. If you want to turn off ignore pause, press F5, OFF.

See the following screen as an example of*tracking* process program header information.

See the following screen as an example of a job header information.

For example, if the current value of $LNSCH[1].$sel_bound = 1, and a line tracking program is executed with Selected Boundary = 6, $LNSCH[1].$sel_bound will be set to 6. When the program is finished, the current boundary will be set back to 1.

10. To set stack size:

a. Move the cursor to Stack size. Refer to Section 6.2.12 for more information on Stack size.

b. Enter the new Stack size.

11. When you have finished entering program information, press F1, END. The teach pendant editor screen will be displayed.

Note Whenever you want to return to the first SELECT menu, press PREV until it is displayed.

12. Turn the teach pendant ON/OFF switch to OFF and release the DEADMAN switch.

13. To define default motion instructions:

a. Continuously press the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.

b. Press F1, POINT.

You will see a list of default motion instructions similar to the following.

```
Default Motion
1: J P[] 100% FINE
2: J P[] 100% CNT100
3: L P[] 100 mm/sec FINE
4: L P[] 100 mm/sec CNT100
```

Note If the instructions listed are the ones you want to use, do not modify them. Go to **Adding Instructions.**

c. To modify the default motion instructions, press F1, ED_DEF.

d. Move the cursor to a component in the default instruction you want to modify.

e. Use the appropriate arrow and function keys to modify the component and press ENTER.

If the [CHOICE] function key is displayed, press F4 to display a list of values for the selected component.

For example, to change the speed value, move the cursor to 100%. Type a new value and press ENTER. The new value will be displayed. Each time you add this instruction to the program the new value will be used.

f. Repeat Step 13d through Step 13e for each default instruction that you want to define.

g. **When you are finished defining default motion instructions**, move the cursor to the instruction you want to be the current default instruction and press F5, DONE.

h. **To save the modified default motion instructions**, refer to Section 9.3.4 , "Backing Up Program, System, and Application Files."

i. **To load default motion instruction files**, refer to Section 9.3.5 , "Loading Files to Controller Memory."

14. To record a position using the current default motion instruction:

a. Jog the robot to the location in the workcell where you want to record the motion instruction.

b. Press and hold in the SHIFT key and press F1, POINT. The instruction will be added to the program automatically.

Note The @ indicator will be displayed on the screen at the current line in the program indicating that the robot is at the current position.

15. To record the position using one of the other three default motion positions:

a. Jog the robot to the location in the workcell where you want to record the motion instruction.

b. Press F1, POINT.

c. Move the cursor to a new default position.

d. Press ENTER and the position is recorded. This then becomes the current default position.

Note The @ indicator will be displayed on the screen at the current line in the program indicating that the robot is at the current position.

16. To add other instructions:

a. Press NEXT until F1, [INST] is displayed, then press F1, [INST].

b. Select the kind of instruction you want and use the appropriate selections on the screen to build the instruction.

Refer to Chapter 6 *PROGRAM ELEMENTS* for details about each instruction.

Note The @ indicator will be displayed on the screen at the current line in the program indicating that the robot is at the current position.

17. When you are finished, turn the teach pendant ON/OFF switch to OFF and release the DEADMAN switch.

Note To test the program, refer to Section 7.3 .

Note If the system variable $BACKGROUND is FALSE, the teach pendant must remain on during programming. If you want to turn the teach pendant off, you must edit the program in the *background*. Refer to Section 5.4 , for more information.

5.3.4 Modifying a Program

You can modify an existing program any time you want to change the content of the program. Modifying a program includes

- **Selecting** a program
- **Modifying** motion instructions
- **Modifying** application-specific instructions
- **Modifying** other instructions
- **Inserting** instructions
- **Deleting** instructions
- **Copying and pasting** an existing instruction or program element
- **Finding and replacing** an existing instruction or program element
- **Renumbering** positions after instructions have been added, removed, or moved

- **Undoing** operations, such as modifying instructions, inserting instructions, and deleting instructions
- **Remarking** instructions, so they are disabled (commented out)
- **Displaying** comments on the teach pendant program screen

Note You cannot modify details if the program is set up as a system level macro. These macros are identified with the letter "s" at the far right side of the Macro Setup screen.

Selecting a Program

You select a program by choosing the program name from a list of existing programs in controller memory. Refer to Chapter 9 *PROGRAM AND FILE MANIPULATION* , "Program and File Manipulation," for more information on loading programs.

Modifying Instructions

Modifying arc welding instructions changes any element of the Arc Start, Arc End, Weave, or Weave End instructions.

Modifying spot welding instructions changes any element of the Spot and Backup instructions.

Modifying dispensing instructions changes any element of the seal start and seal end instructions.

Modifying Other Instructions

When you modify other instructions, you change any element of the instruction.

Inserting Instructions

When you insert instructions, you place a specified number of new instructions between existing instructions. When you insert an instruction, the instructions that follow the new instruction are automatically renumbered.

Deleting Instructions

When you delete instructions, you remove them from the program permanently. When you remove an instruction the remaining instructions are automatically renumbered.

Modify Motion Instructions

Each of the five components of a motion instructions can be changed by moving the cursor to the component and pressing F4, [CHOICE], or entering new numeric data.

Touching up a Motion Instruction

When you touch up a motion instruction, only the positional data at that memory location is changed. Other components of the motion instruction are unchanged.

Copying and Pasting Instructions

When you copy and paste, you select a group of instructions, make a copy of the group, and insert the group at one or more locations in the program. You can paste copied program instructions using the methods described in Table 5–3 .

Table 5–3. Paste Methods

Method	Description	Pasted Program Instructions

Copied program instructions:
 	
`1: J P[1] 100% CNT100`
`2: L P[2] 500mm/sec CNT80`
`3: DO[1] = ON`
`4: J P[3] 50% CNT50`
`5: L P[4] 10mm/sec FINE`
`6: CALL HOME`		

LOGIC (F2)	• Pastes the lines exactly as they were copied
	• Does not record positions
	• Leaves the position numbers blank
		`7: J P[...] 100% CNT100`
		`8: L P[...] 500mm/sec CNT80`
		`9: DO[1] = ON`
		`10: J P[...] 50% CNT50`
		`11: L P[...] 10mm/sec FINE`
		`12: CALL HOME`

POS_ID (F3)	• Pastes the lines exactly as they were copied
	• Retains the original position numbers
		`7: J P[1] 100% CNT100`
		`8: L P[2] 500mm/sec CNT80`
		`9: DO[1] = ON`
		`10: J P[3] 50% CNT50`
		`11: L P[4] 10mm/sec FINE`
		`12: CALL HOME`

POSITION (F4)	• Pastes the lines exactly as they were copied
	• Renumbers the copied positions with next available position numbers
	• Retains copied positional data
		`7: J P[5] 100% CNT100`
		`8: L P[6] 500mm/sec CNT80`
		`9: DO[1] = ON`
		`10: J P[7] 50% CNT50`
		`11: L P[8] 10mm/sec FINE`
		`12: CALL HOME`

Table 5–3. Paste Methods (Cont'd)

Method	Description	Pasted Program Instructions
CANCEL (F5)	Cancels the paste and retains the copied lines so you can paste them elsewhere	
R-LOGIC (NEXT+F1)	• Pastes the lines in reverse order • Does not record positions • Leaves the position numbers blank	7: CALL HOME 8: L P[...] 10mm/sec FINE 9: J P[...] 50% CNT50 10: DO[1] = ON 11: L P[...] 500mm/sec CNT80 12: J P[...] 100% CNT100
R-POS-ID (NEXT+F2)	• Pastes the lines in reverse order • Retains the original position numbers	7: CALL HOME 8: L P[4] 10mm/sec FINE 9: J P[3] 50% CNT50 10: DO[1] = ON 11: L P[2] 500mm/sec CNT80 12: J P[1] 100% CNT100
R-POS (NEXT+F4)	• Pastes the instructions in reverse order • Renumbers the copied positions with the next available position numbers	7: CALL HOME 8: L P[8] 10mm/sec FINE 9: J P[7] 50% CNT50 10: DO[1] = ON 11: L P[6] 500mm/sec CNT80 12: J P[5] 100% CNT100

Table 5–3. Paste Methods (Cont'd)

Method	Description	Pasted Program Instructions

RM-POS-ID (NEXT+F3)	• Pastes all instructions except motion instructions in reverse order
	• Motion instructions are created using the current (first) and next (second) motion instruction:
	— Motion type, speed, and motion-related options are used from the second motion instruction
	— Positional data, termination type, and non-motion-related options are used from the first motion instruction
	— The bottom motion instruction in the copied area is pasted as is
	• Retains the original position numbers
	• Modal instructions, such as I/O, are pasted in reverse order, but the states are not changed (ON to OFF, or OFF to ON); you must modify these after the paste
	• UTOOL and UFRAME change instructions and position register instructions are pasted in reverse order, but the frame numbers and register numbers are not changed; you must modify these after the paste
	• Motion instructions that contain the following motion options are only copied in reverse order and cause a warning to be displayed:
	— Application commands
	— SKIP
	— INC (incremental motion)
	— Continuous turn
	— TIME BEFORE/AFTER
		1st:
		`J P[i] 100% CNT100, Offset,PR[x]`
		2nd:
		`L P[j] 50 mm/sec FINE ACC150 PTH`
		Result:
		`L P[j] 50mm/sec FINE ACC150 PTH`
		`L P[i] 50mm/sec CNT100`
		`Offset,PR[x] ACC150 PTH`
		`7: CALL HOME`
		`8: L P[4] 10mm/sec FINE`
		`9: L P[3] 10mm/sec CNT50`
		`10: DO[1] = ON`
		`11: J P[2] 50% CNT80`
		`12: L P[1] 500mm/sec CNT100`

Table 5–3. Paste Methods (Cont'd)

Method	Description	Pasted Program Instructions

RM-POS (NEXT+F5)	• Pastes all instructions except motion instructions in reverse order
	• Motion instructions are created using the current (first) and next (second) motion instruction:
	— Motion type, speed, and motion-related options are used from the second motion instruction
	— Positional data, termination type, and non-motion-related options are used from the first motion instruction
	— The bottom motion instruction in the copied area is pasted as is
	• Renumbers the copied positions with the next available position numbers
	• Modal instructions, such as I/O, are pasted in reverse order, but the states are not changed (ON to OFF, or OFF to ON); you must modify these after the paste
	• UTOOL and UFRAME change instructions and position register instructions are pasted in reverse order, but the frame numbers and register numbers are not changed; you must modify these after the paste
	• Motion instructions that contain the following motion options are only copied in reverse order and cause a warning to be displayed:
	— Application commands
	— SKIP
	— INC (incremental motion)
	— Continuous turn
	— TIME BEFORE/AFTER
		1st:
		`J P[k] 100% CNT100, Offset,PR[x]`
		2nd:
		`L P[l] 50 mm/sec FINE ACC150 PTH`
		Result:
		`L P[l] 50mm/sec FINE ACC150 PTH`
		`L P[k] 50mm/sec CNT100`
		`Offset,PR[x] ACC150 PTH`
		`7: CALL HOME`
		`8: L P[8] 10mm/sec FINE`
		`9: L P[7] 10mm/sec CNT50`
		`10: DO[1] = ON`
		`11: J P[6] 50% CNT80`
		`12: L P[5] 500mm/sec CNT100`

Note When you use RM-POS-ID and RM-POS, motion instructions that contain the following motion options are only copied in reverse order and cause a warning to be displayed:

- Application commands
- SKIP
- INC (incremental motion)
- Continuous turn
- TIME BEFORE/AFTER

Finding and Replacing Instructions

When you find and replace, you find specific instructions and, if desired, replace those instructions with new instructions. This function is useful, for example, when setup information that affects the program is changed. It is also useful when you need to find a specific area of a long program quickly.

Determining Paint Instruction Status (PaintTool only)

Determining paint instruction status allows you to find the most recent paint instruction and how it was used. For example, depending on the position of the cursor in the program, you can determine the status of the most recent GUN instruction to find whether the applicator is ON or OFF. You can determine the status of the following instructions:

- Preset[x] (Application Parameter Option)
- Electrostat[x]
- GunSel[x]
- Utool[x]

Renumbering Positions

You can renumber positions in the program. When you add positions in a program, the first available position number is assigned to the position, regardless of its place in the program. When you delete motion instructions, all remaining positions keep their current numbers. Renumbering reassigns all position numbers in the program so that they are in sequential order.

Undoing Operations

You can undo an operation to recover from the last incorrect or faulty operation. You can undo the previous operation to edit the program, and you can redo the last Undo operation. Several modifications for one line can be recovered with a single Undo operation if you have not modified any other line in between modifications.

You cannot Undo an operation when any of the following conditions exist:

- The controller has been turned off and turned on

- Another program has been selected
- You are using a CRT and keyboard
- The teach pendant ON/OFF switch is OFF
- The program is write-protected
- The teach pendant does not have enough available memory

Also, you cannot Undo an operation in HandlingTool when any of the following conditions exist:

- Modifying palletizing command
- Deleting program lines that include palletizing instructions.
- Copying program lines that include palletizing instructions.
- Pasting lines that include palletizing instruction.
- Replacing or re-numbering at the place in the program that has palletizing instructions.

In addition, Undo will not work if you have executed any of the following instructions in a program:

- Palletizing instructions (HandlingTool)
- Dispense instructions (DispenseTool)
- Line tracking instructions
- On the fly (ArcTool)
- On-line touch up

Note If power fails in the process of performing an Undo operation, unexpected results can occur, and the desired modification is not guaranteed to have taken effect when power is restored.

Remarking Instructions

When you remark an instruction, you disable the instruction by commenting it out. // is shown at the beginning of the instruction. You cannot remark just a portion of the instruction. You will not be able to edit the instruction until you unremark it. You can perform other modifications on the instruction such as delete, copy and paste, find and replace, and renumber.

Displaying Comments on the Teach Pendant Program Screen

This function displays comments of I/O and registers on the teach pendant program screen. This gives you the ability to confirm comments while you edit a teach pendant program, without having to display another screen.

You can only**display** comments for the following instructions while you are editing a teach pendant program. You cannot change the comments on the teach pendant program screen.

- I/O instructions (DI[i:COMMENT], DO[i:COMMENT], RI[i:COMMENT],RO[i:COMMENT], GI[i:COMMENT], GO[i:COMMENT], AI[i:COMMENT], AO[i:COMMENT], UI[i:COMMENT], UO[i:COMMENT], SI[i:COMMENT], SO[i:COMMENT])
- Register instructions (R[i:COMMENT])
- Position register instructions (PR[i:COMMENT])
- Position register element instructions (PR[i.j:COMMENT])

The display of the comments is turned on and off when you select the [Comment] by pressing F5, [EDCMD].

Note Instructions that include indirect addressing do not display the comments, as in the following example:**R[R[i]], DO[R[i]], ...**

Use Procedure 5-2 to modify a program.

Procedure 5-2 Modifying a Program

Conditions

Note You cannot modify a SpotTool+ program that has been created prior to software version 4.10 unless you have converted it to the appropriate version format.

- All personnel and unnecessary equipment are out of the workcell.
- The program has been created and all detail information has been set correctly (refer to Section 5.3.2).

Steps

1. **Select a program to modify:**
   a. Press SELECT. You will see a screen similar to the following.

```
Select
               702566 bytes free   7/7
No.  Program name           Comment
 1   -BCKEDT-         [                   ]
 2   HOME_IO          [SET AT HOME I-O    ]
 4   TEST1            [TEST1              ]
 5   UOP_ROBOT_12>    [                   ]
  UOP_ROBOT_12436
```

Note Any program name that is longer than 12 characters is abbreviated on the SELECT screen and displayed with a > at the end. When the program name is selected, the entire program name is displayed at the bottom of the screen.

b. Press F1, [TYPE].

If F1, [TYPE], is not displayed on the screen, press >, NEXT, until it is displayed.

c. Select the list you want:

- **All** displays all programs.
- **TP Programs** displays all teach pendant programs.
- **KAREL Progs** displays all KAREL programs.
- **Macro** displays all macro programs.
- **Cond** displays all conditional programs.
- **Jobs** displays all job programs for DispenseTool and PaintTool.
- **Processes** displays all process programs for DispenseTool and PaintTool.

Note You cannot modify details if the program is set up as a system level macro. These macros are identified with the letter "s" at the far right side of the Macro SETUP screen.

d. Move the cursor to the name of the program you want to modify and press ENTER.

e. Continuously press the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.



Caution

Recorded positions are not affected by UFRAME, and UFRAME has no effect during playback. However,**position registers are recorded with respect to the current UFRAME.** If you change UFRAME, any recorded position registers will also change.

2. To touch up and modify motion instructions:

a. Move the cursor to the **line number** of the motion instruction you want to modify.

b. **To change only the position component of the motion instruction,** jog the robot to the new position, press and hold in the SHIFT key and press F5, TOUCHUP. If F5, TOUCHUP, is not displayed, press >, NEXT.



Warning

If you change the motion type of a positional instruction from linear to joint you can cause the speed value to change from mm/sec to a default value as high as 100%. Be sure to check the speed value before you execute the instruction; otherwise, you could injure personnel or damage equipment.

c. **To change other motion instruction components**, move the cursor to the component using the arrow keys, and press the appropriate function keys to modify the component:

- **If function key labels are available**, press the appropriate one.
- **If no function key labels are available**, press F4, [CHOICE], and select a value.
- **To change the position value**, move the cursor to the position number and press F5, POSITION. The position screen will be displayed showing the Cartesian coordinates or joint angles of the selected position. Move the cursor to the component you want to change and enter the new value using the number keys. To make other changes, use the function keys, described here.

```
Position Detail
 P[1] UF:0  UT:1      CONF: N   0   0
 X  1829.992   mm   W -179.998  deg
 Y      .050   mm   P  -90.000  deg
 Z  1170.024   mm   R    0.000  deg
```

— **To change the motion group number**, press F1, GROUP. This applies only to systems that have been set up for multiple groups.

— **To display components for extended axes**, press F2, PAGE. This only applies to systems that include extended axes.

— **To change the configuration** between flip (F) and no-flip or normal (N), press F3, CONFIG, and then use the up and down arrow keys to change F to N and N to F.

Note Joint angles are useful for zero-positioning the robot or for non-kinematic motion control such as controlling the motion of a positioning table.

— **To change the format of the position from Cartesian coordinates to joint angles or from joint angles to Cartesian coordinates,** press F5, [REPRE] and select the coordinate system. The position is converted automatically.

— When you are finished, press F4, DONE.



Caution

When you change the representation from Cartesian to joint, the user frame and tool frame values currently in effect will be used to calculate the joint angles. After you have changed from Cartesian to joint, if you modify user frames and tool frames, these changes will have no effect on the position with joint representation, regardless of the motion type of the motion instruction that contains the position.

— **To change the comment associated with the position**, move the cursor to the position value and press ENTER. Follow the information on the screen to enter the position comment. Press ENTER when you are finished changing the comment.

d. Repeat Step 2a through Step 2c for each motion instruction you want to modify.

3. To modify other instructions:

a. Move the cursor to the line number of the instruction you want to modify.

b. Move the cursor to the component you want to modify and press the appropriate key:

- **If function key labels are available**, press the appropriate one.
- **If no function key labels are available**, press F4, [CHOICE], and select a value.

c. Repeat Step 3a and Step 3b for each instruction you want to modify.

4. To insert instructions:

a. Decide where you want to insert the instruction. Move the cursor to the line following that point. The cursor must be on the**line number** . For example, if you want to insert between lines 5 and 6 place the cursor on line 6.

b. Press NEXT, >, until F5, [EDCMD] is displayed.

c. Press F5, [EDCMD].

d. Select Insert.

e. Type the number of lines to insert and press ENTER. A blank line will be inserted into the program for each line you want inserted. All lines in the program will be renumbered automatically. Your program will look similar to the following.

```
5: J P[4] 50% CNT50
6:
7:
8:
9: J P[5] 50% CNT10
```

f. Move the cursor to the line number of any inserted line and add any instruction.

5. To delete instructions:

a. Move the cursor to the line number of the instruction you want to delete. If you want to delete several instructions in consecutive order, move the cursor to the first line to be deleted.

> **Caution**
>
> If you delete an instruction, it will be permanently removed from the program. Be sure you want to remove an instruction before you continue; otherwise, you could lose valuable information.

b. Press NEXT, >, until F5, [EDCMD] is displayed.

c. Press F5, [EDCMD].

d. Select Delete.

e. **To delete a range of lines,** move the cursor to select the lines to be deleted. The line number of each line to be deleted will be highlighted as you move the cursor.

f. Delete the line or lines:

- **If you do not want to delete the selected line(s)**, press F5, NO.
- **To delete the selected line(s)** press F4, YES.

Note You can copy instructions from one program and paste them within that program or into another program.

6. To copy and paste instructions:

a. Press NEXT, > until F5, [EDCMD] is displayed.

b. Press F5, [EDCMD].

c. Select Copy.

d. Move the cursor to the first line to be copied.

e. Press F2, COPY.

f. Move the cursor to select the range of lines to be copied.

The line number of each line to be copied will be highlighted as you move the cursor.

g. Press F2, COPY, again.

h. Decide where you want to paste the lines. Move the cursor to the line following that point. The cursor must be on the**line number**.

i. Press F5, PASTE.

j. Press the function key that corresponds to the way you want to paste the copied lines (refer to Table 5–3 for details and examples of each paste method).

k. Repeat Step 6i through Step 6j to paste the same set of instructions as many times as you want.

Note When you use RM-POS-ID and RM-POS, motion instructions that contain the following motion options are only copied in reverse order and cause a warning to be displayed:

- Application commands
- SKIP
- INC (incremental motion)
- Continuous turn
- TIME BEFORE/AFTER

l. **When you are finished copying and pasting instructions**, press PREV.

7. To find instructions:

a. Move the cursor to the line number of any instruction.

b. Press NEXT, >, until F5, [EDCMD], is displayed.

c. Press F5, [EDCMD].

d. Select Find.

e. Select the kind of instruction to find.

f. When prompted, enter the necessary information.

The system searches forward from the current cursor position for the item you want. If it finds an instance of the item, it highlights it on the screen.

g. **To find the next instance of the item**, press F4, NEXT.

h. **When you are finished finding items**, press F5, EXIT.

i. If you pressed F5, EXIT, press PREV. You do not need to press PREV if you pressed F4, NEXT, until all instances had been found.

Note In ArcTool and HandlingTool, you can not use the Find command to locate track/offset instructions or touch sense instructions.

8. To replace instructions:

a. Move the cursor to the line number of any instruction.

b. Press NEXT, > until F5, [EDCMD], is displayed.

c. Press F5, [EDCMD].

d. Select Replace.

e. Select the instruction you want to replace from the list of instructions. Follow the information on the screen to specify the instruction.

The system finds the first instance of the existing instruction and highlights it.

f. Select the replacement item and enter the necessary information.

g. Decide how to replace the instruction:

- **To replace the existing instruction with a new instruction in all instances,** press F2, ALL. The system will prompt you to confirm that you want to replace all instances of the instruction.
- **To replace the existing instruction with the new instruction,** press F3, YES. The system will prompt you to search for the next one.
- **To ignore this instance and find the next**, press F4, NEXT, and the system will find the next instance, if there is one.
- **To stop the cancel and replace operation**, press F5, EXIT.



Caution

In ArcTool and HandlingTool, you cannot use the*Replace* command to replace a motion instruction with a touch sense or track/offset instruction. Doing so causes a memory write failure error. If you want to replace the motion instruction, first delete the motion instruction and then insert the touch sense or track instruction.

9. If you are running PaintTool, perform the following steps. Otherwise, skip to Step 10 .

a. To determine the Paint Instruction Status, move the cursor to the line number of any instruction.

b. Press NEXT, >, until F2, STAT, is displayed.

c. Press F2, STAT. See the following screen for a paint status example when the cursor is at an instruction in the program that is after a gun=on instruction, and an electrostatic instruction.

```
Paint Status
  1 Gun=ON              5
  2                     6
  3 ElectroStat[4]      7
  4                     8
PROC742
  5:L   P[ 6] CNT100  GUN=ON
  6:L   P[ 7] CNT100
  7:L   P[ 8] CNT100
  8:L   P[ 9] CNT100
  9:L   ElectroStat[4]
 10:L   P[10] CNT100
```

10. To renumber positions:

a. Move the cursor to the line number of any instruction.

b. Press NEXT, >, until F5, [EDCMD], is displayed.

c. Press F5, [EDCMD].

d. Select Renumber.

e. Renumber the positions:

- **If you do not want to renumber positions** press F5, NO.
- **To renumber positions** press F4, YES.

11. To undo an operation:

a. Press NEXT, >, until F5, [EDCMD], is displayed.

b. Press F5, [EDCMD].

c. Select Undo.

d. Undo the operation. The recovered operation is displayed.

- **If you do not want to undo the operation**, press F5, NO.
- **To undo the operation**, press F4, YES.

e. **To cancel the undo operation**, press F5, [EDCMD] , then press Undo. Press F4, YES, to redo the operation.



Warning

If you have used Undo, you should thoroughly test the modified program at a low motion speed before allowing it to run continuously. Otherwise, you could injure personnel or damage equipment. Refer to Chapter 7 *TESTING A PROGRAM AND RUNNING PRODUCTION* for more information on testing a program.

12. To remark/unremark instructions:

a. Move the cursor to the line number of the instruction you want to remark or unremark. If you want to change several instructions in consecutive order, move the cursor to the first line to be changed.

b. Press NEXT, >, until F5, [EDCMD] is displayed.

c. Press F5, [EDCMD].

d. Select Remark.

e. **To remark/unremark a range of lines,** move the cursor to select the lines to be changed. The line number of each line to be changed will be highlighted as you move the cursor.

f. Remark or unremark the line or lines:

- **To remark the selected line(s)** press F4, REMARK
- **To unremark the selected line(s)** press F5, UNREMARK

Note When remarking lines, any lines already remarked are skipped. When unremarking lines, any lines not remarked are skipped. You will not be able to edit a line if it is remarked.

13. To display comments on the teach pendant program screen, see the following screen for an example.

```
PNS0001
   1:  R[2:COUNTER1]=DI[3:HAND1]
   2:  DO[1:HAND1ACK]=ON
[END]
```

a. Press F5, [EDCMD].

```
PNS0001
                                        1/3
     1:  R[2:COUNTER1]=DI[3:HAND1]
     2:  DO[1:HAND1ACK]=ON +-----------------+
                           | 1 Insert        |
[END]                      | 2 Delete        |
                           | 3 Copy          |
                           | 4 Find          |
                           | 5 Replace       |
                           | 6 Renumber      |
                           | 7 Comment       |
                           | 8 Undo          |
                           +----------+      |
```

b. If you select Comment, the comments turn off.

```
PNS0001
                                        1/3
   1:  R[2]=DI[3]
   2:  DO[1]=ON
[END]
```

c. Press F5, [EDCMD], again. If you select Comment, the comments turn on.

```
PNS0001
                                          1/3
   1:  R[2:COUNTER1]=DI[3:HAND1]
   2:  DO[1:HAND1ACK]=ON
[END]
```

14. **When you are finished**, turn the teach pendant ON/OFF switch to OFF and release the DEADMAN switch.

Note To test the program, refer to Chapter 7 *TESTING A PROGRAM AND RUNNING PRODUCTION* .

5.3.5 Setting Up Predefined Positions and Using Them in a Program

Setting up predefined positions, such as home, gun clean in/out involves writing a program that contains motion instructions that move to the predefined position. The program names for home, repair, and gun clean in/out are predefined.

Using predefined positions in a program involves adding CALL PROGRAM instructions that call the program in which you set up the predefined position.

Procedure 5-3 describes how to set up predefined positions. Procedure 5-4 describes how to use predefined positions in a program.

Procedure 5-3 Setting Up Predefined Positions

Conditions

- All personnel and unnecessary equipment are out of the workcell.

Steps

1. Press SELECT.
2. Select the predefined program name you want to use. For example, if you want to build the program for the home position, select the program name HOME.PR.
3. **To display program header information**, press F2, DETAIL. You will see a screen similar to the following.

```
Program Detail
                                          1/6
   Creation date:               01-Jan-xxxx
   Modification Date:           01-Jan-xxxx
   Copy source:          [                 ]
   Positions:   10  Size             17 Byte
    Program Name
   1 HOME
   2 Sub Type:           [PROCESS          ]
   3 Comment:            [                 ]
   4 Group Mask:         [1,*,*,*,*,*,*,*  ]
   5 Write protect:      [ON               ]
   6 Ignore Pause:       [OFF              ]
```

4. Add motion instructions to the program to move the robot to the desired position. Keep motion speed slow to ensure the safety of personnel and equipment any time the position is reached.

Procedure 5-4 Using Predefined Positions in a Program

Conditions

- The predefined position has been set up. (Refer to Procedure 5-3 .)

Steps

1. Press SELECT.
2. Select the program in which you want to use the predefined position and press ENTER.
3. Position the cursor on the line before which you want to move to the predefined position.
4. Insert one line for the instruction. (Refer to Section 5.3.4 .)
5. Press NEXT until F1, [INST] is displayed.
6. Press F1, [INST].
7. Select CALL.
8. Select CALL program.
9. Select the program for the predefined position you want to use:
   - **For the home position,** select HOME.
   - **For the bypass position,** select BYPASS.
   - **For the clean_in position,** select CLNIN.
   - **For the clean_out position,** select CLNOUT.
   - **For the purge position,** select PURGE.

- **For the first special position,** select SPECIAL1.
- **For the second special position,** select SPECIAL2.

5.3.6 Creating a Macro Program

To create a macro program, you must create a program of sub type macro and include the appropriate instructions. Use Procedure 5-5 to create a macro program.

Procedure 5-5 Creating a Macro Program

Conditions

- All personnel and unnecessary equipment are out of the workcell.

Steps

1. Create a program, using the**Macro** sub type. See the following screen for an example.

```
Program Detail
                                          1/6
   Creation date:              01-Jan-xxxx
   Modification Date:          01-Jan-xxxx
   Copy source:          [                ]
   Positions:  FALSE   Size         17 Byte

   Program Name
   1 PROG742
   2 Sub Type:           [Macro           ]
   3 Comment:            [                ]
   4 Group Mask:         [1,*,*,*,*,*,*,* ]
   5 Write protect:      [ON              ]
   6 Ignore Pause:       [OFF             ]
```

2. Add the instructions you want to include in the macro program.

Note To execute the macro program, refer to Chapter 3 *GENERAL SETUP* .

5.4 MODIFYING A PROGRAM IN THE BACKGROUND (BACKGROUND EDITING)

5.4.1 Overview

Background editing is used to modify a program when the teach pendant is off. This can also be used to edit a program while another program is running. You do not have to stop the robot to modify or check another program. This option can improve the efficiency of production and maintenance.



Warning

If the teach pendant is off, make sure you are at a safe distance (outside of the robot workcell) when you edit a program while another program is running. The teach pendant is not in control of the robot during this time. Otherwise, the robot could injure personnel or damage equipment.

During background editing, you **can**

- Create and delete programs.
- Add new program instructions.
- Add new motion instructions.

  The position recorded will be the current position of the robot.

  — If the robot is currently executing a motion instruction in another program, the robot position at the time you add the motion instruction will be the recorded position.

  — If the robot is not executing a motion instruction in another program, the current robot position will be the recorded position.
- Modify existing program instructions.

During background editing, you **cannot** move the robot. You cannot move the robot unless the teach pendant is enabled.

If you add motion instructions during background program editing, you must remember to touch up the positions using TOUCHUP in the foreground, before you run the program.

For more information about the system variables related to background editing, refer to the *Software Reference Manual*.

Use Procedure 5-6 to modify a program in the background.



Caution

You cannot edit system level macros in the foreground or the background. If you attempt to do so, you will not be able to save your changes.

5.4.2 Background Edit Process Flow

Figure 5–8 and Figure 5–9 show how the Background Edit process flows.

Figure 5–8. Background Edit Process

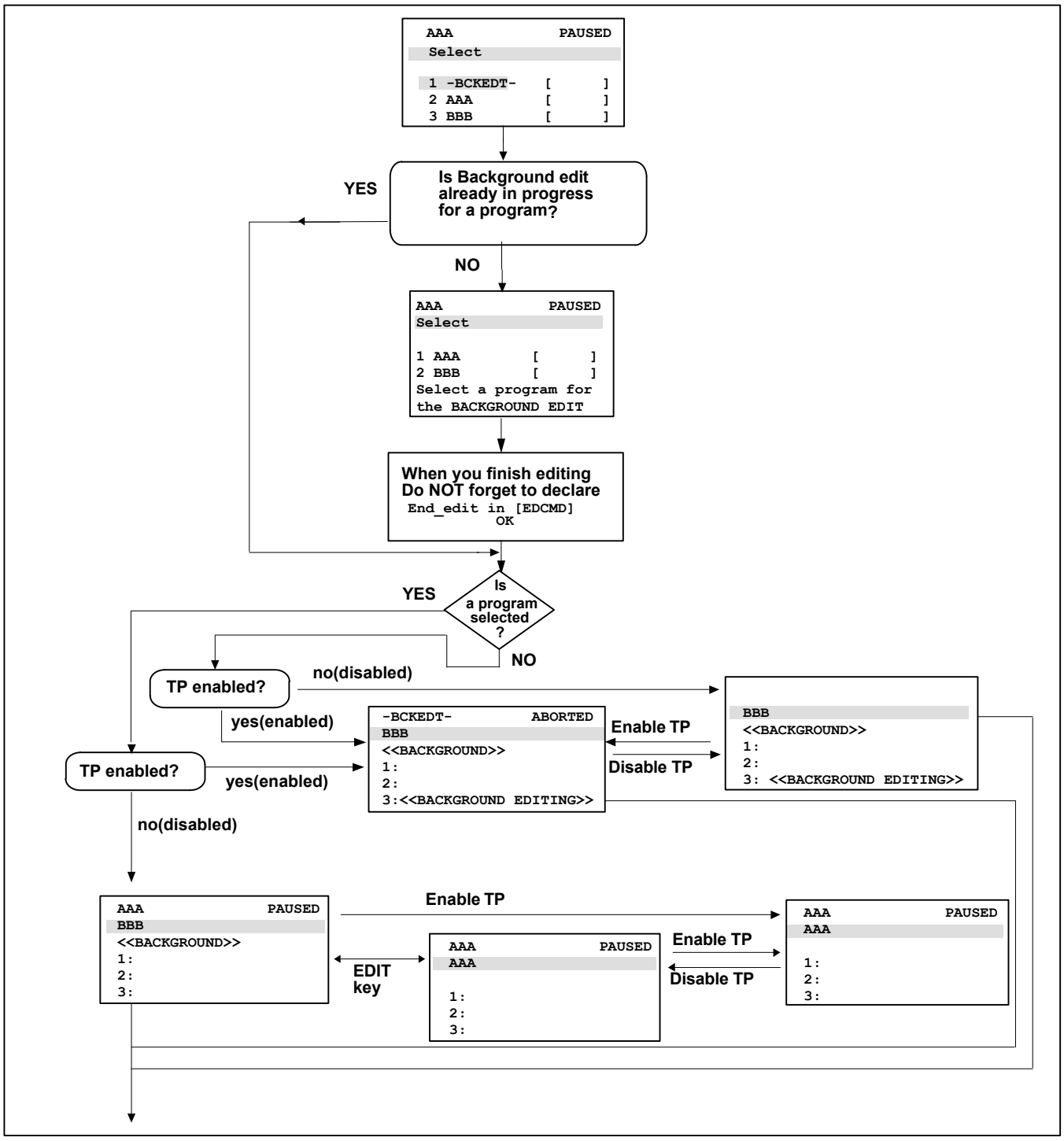
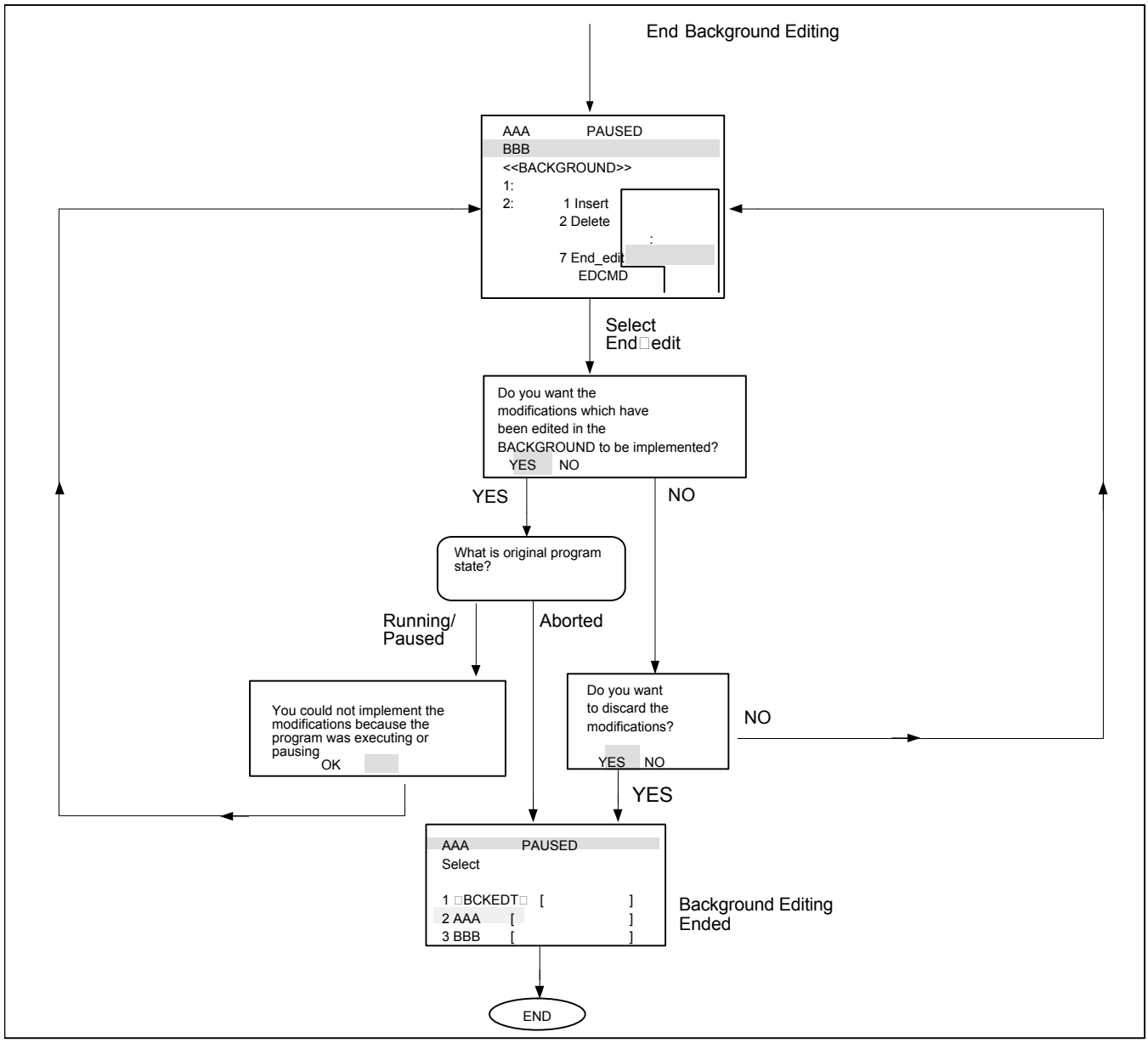


Figure 5–9. Background Edit Process (Continued)



5.4.3 Background Editing Using the iPendant

When you use background edit on the *i*Pendant, you must remember the following guidelines:

- Background edit is supported in each window. There will be a background edit program unique for each window so multiple edit sessions will not affect each other.

- The background edit program for dual window is –BCKED2- and for third window is –BCKED3-.
- When a program is selected for background edits, then the comment for the background edit program will show the selected program.
- The same program cannot be background edited in multiple edit sessions. If you select the same program already being background edited in another window, then the warning " TPIF-167 Program already in background edit" is posted. You need to End_edit the background session in the other window first.
- Background edit programs are retained during cycle power.

5.4.4 Modifying a Program in the Background

Procedure 5-6 Modifying a Program in the Background

Conditions

- All personnel and unnecessary equipment are out of the workcell.
- The program has been created and all detail information has been set correctly. (Section 5.3.2)

Steps

1. Disable the teach pendant.

   Note If the teach pendant is enabled when you perform this procedure, the program you select for background edit will be executed instead.

2. Press SELECT. You will see a screen similar to the following.

```
Select
               702566 bytes free   7/7
No.  Program name           Comment
 1   -BCKEDT-         [                    ]
 2   COND             [SET AT HOME I-O     ]
 4   MAIN             [TEST1               ]
 5   MSG              [                    ]
 6   TEST111_PLAN>    [                    ]
 7   UOP_ROBOT_12>    [                    ]




TEST111_PLANT1_ROBOT15467_A
```

3. Select the special program used for background editing. The name of this program is -BCKEDT-.

   - **If a program is currently running in the background**, you will automatically be taken back to the background editing session. Go to Step 7 .

   - **If a program is not already running in the background**, you must select a program to edit in the background. You will see a screen similar to the following.

```
Select
               702566 bytes free   7/7
No.  Program name           Comment
 1   -BCKEDT-         [                    ]
 2   COND             [SET AT HOME I-O     ]
 4   MAIN             [TEST1               ]
 5   MSG              [                    ]
 6   TEST111_PLAN>    [                    ]
 7   UOP_ROBOT_12>    [                    ]


Select a program for the BACKGROUND EDIT.
```

4. Move the cursor to the name of the program you want to edit.
5. Press ENTER.
6. You will see a confirmation message. Press ENTER. "<<BACKGROUND>" will be displayed at the beginning of the program.
7. Modify the program. Refer to Procedure 5-2 .

   Note Any modifications you make to the background program do not take effect until you end background editing.

8. **To toggle between two programs (EDIT key)**, one in the foreground and one in the background, press EDIT. If no program is selected in the foreground, the special program (BCKEDT) is brought to the foreground.

   Note You cannot edit two or more programs in the background at the same time in the same pane. To edit another program in the background, you must first end the background editing of the first program by selecting End_edit. Then restart background editing. Disable the teach pendant. Press the EDIT key or re-select -BCKEDT- on the program Select screen.

If an external start signal is received during background editing, the program selected in the foreground is started.

If the program you selected for background edit is run during automatic operation, or called as a subprogram, the original program is executed (instead of the program which you changed in the background).

If you select a program with an external program selection function, (such as PNS) during background editing, background editing will continue normally.

9. **When you are finished editing the program in the background**, end the background editing session:
   a. Press NEXT, > until F5, [EDCMD] is displayed.
   b. Press F5, [EDCMD].
   c. Select End_edit.
10. Save the changes.

- **If you want to save the changes you made**, move the cursor to YES and press ENTER. You will be returned to the program SELECT screen and < < BACKGROUND> will no longer be displayed at the beginning of the program.

- **If you do not want to save the changes you made**, move the cursor to NO and press ENTER. You will be given the option to disregard the changes or be returned to the current background edit session.

  Note You cannot implement the changes you made if the currently selected program is running or paused. You must first select OK and press ENTER before you can save the program. The background editing session will be resumed.

5.4.5 Troubleshooting Background Edit

When you use Background Edit you might experience one or more of the problems mentioned in Table 5–4 . Use the Cause and Remedy information in Table 5–4 to eliminate any problems that might occur during normal operation.

Table 5–4. Troubleshoot Background Edit - Problem Cause and Remedy

Problem	Cause	Remedy
TPIF-005 Program is not selected was displayed when you pressed the EDIT key.	A program is not selected in the foreground.	There must be two programs selected (one in the foreground and one in the background) in order to use the EDIT key to toggle between them.
TPIF-053 Not editing background program was displayed when you pressed the EDIT key.	A program is not selected in the background.	
You cannot start background edit for the selected program.	There is not enough memory available to copy the selected program for editing.	The amount of available memory must be larger than the size of the selected program to start background editing.
TPIF-054 Could not end editing or **MEMO-126 No more available memory** Not enough memory available to save the changes you made during background editing.	When background editing ends, a backup copy of the original program is created. The original program is then modified. These two programs require more memory than is currently available.	The amount of memory must be larger than the original program and the program copied for background editing, in order to save any changes you made during the background editing session.
TPIF-055 Could not recover original program The original program is corrupt and cannot be recovered.	Power to the robot was turned off, then back on during background editing.	You must recover the backup version. Check the original program. Then test the program continuously to eliminate the possibility of any errors occurring. Refer to Chapter 7 *TESTING A PROGRAM AND RUNNING PRODUCTION* .

Table 5–4. Troubleshoot Background Edit - Problem Cause and Remedy (Cont'd)

Problem	Cause	Remedy
The robot stops and the following message is displayed.**Program was executing or ...**	You tried to run the original program before ending the background editing session.	Do not run the original program until you end (End_edit) background editing.
The robot stops, and the following error messages are displayed:**SYST-011 Failed to run task or MEMO-004 Specified program is in use**		
The robot stops, and the following error messages are displayed:**SYST-011 Failed to run task or MEMO-008 Specified line no. not exist**		
TPIF-054 Could not end editing or TPIF-008 Memory protect violation	The original program is write protected.	You cannot end background editing. First change the write protection on the original program. Then edit the program in the background.
After you abort a subprogram, the status line continues to indicate the name of the subprogram.	If a main program is selected in the foreground.	The status line indicates the execution state of the selected program.
TPIF-104 Teach Pendant is disabled	The teach pendant is disabled and you are trying to create or delete a program.	Select the program from the Program SELECT screen. The background editing screen will then be displayed.

Table 5–4. Troubleshoot Background Edit - Problem Cause and Remedy (Cont'd)

Problem	Cause	Remedy
The status of a running subprogram, that was executed from the BCKEDT program, changed to ABORTED when you disabled the teach pendant. The program list screen was displayed.	You selected the special program (BCKEDT) for background editing, while the teach pendant was enabled. You then disabled the teach pendant before the program completed.	If you select the special program for background editing while the teach pendant is enabled, do not disable the teach pendant until the program completes.
A program status changes from RUNNING to ABORTED.		
This program is being edited	The special program (BCKEDT) cannot be loaded from the default device if there is already a program for which completed editing is being held in memory.	End background editing (End_edit) before you attempt to load the special program (BCKEDT) from the default device.
TPIF-054 Could not end editing or **TPIF-008 Memory protect violation**	You cannot edit a macro in the background.	Macros must be edited when no programs are running.

5.5 SETTING PROGRAM STORAGE CONFIGURATION

This option "R709 TP DRAM/FILE Storage" provides for storage of TP programs outside of CMOS. This allows for an order of magnitude larger program space when using SHADOW DRAM and virtually unlimited space when using files.

The LARGEST TPP program supported is dependent on CMOS size. So without this feature you get ONE 7000 point program. With this feature you can get 8000 7000 point programs. The total number of points loaded into memory at any one time depends on memory configuration. However, with a 64M DRAM almost 150,000 points in memory at the same time is feasible. More points than that requires the use of the FILE type or ONDEMAND type. For these cases not all points are in DRAM at the same time.

In general this enhanced capacity is provided through SHADOW programs. The user model for SHADOW is the same as CMOS. However, the storage for the programs consumes DRAM rather than limited CMOS. So any system configuration with unused DRAM can take advantage of this feature

This feature also provides the option that not ALL programs are in DRAM at the same time. Many applications which have thousands of TP programs but only 100 are used on any given day. This is the FILE option. The users 1000's of programs might be on a memory card, FlashROM or other storage device. The storage device can be Network storage which provides almost unlimited storage.

As FILE programs are called on to be executed they get moved into memory. This loading into memory can be time consuming but in a lot of cases on systems this is not a constraint. Of course once it's moved into memory it stays in memory until some action is taken to remove it.

As an intermediate option SHADOW ONDEMAND is provided. Like the FILE option programs are only loaded as needed. The limitation of this option is that the user programs are on the internal FlashROM only. So this option doesn't support memory card of other storage devices. In this case the user doesn't deal with or even need to know that there are files involved. This is just a way for him to structure a system where all the programs don't need to be loaded all the time. However, the load times for SHADOW ONDEMAND are ten times faster than FILE type programs.

The key is the STORAGE states that a program can take on. The "permanent" states are:

Table 5–5. Permanent Program Storage States

Storage	Meaning	Constant
CMOS	Same as standard controller	TPSTOR_CMOS
SHADOW	Program is in SHADOW memory	TPSTOR_SHAD
SHADOW ONDEMAND	Program will be loaded to SHADOW when needed	TPSTOR_SHOD
FILE	Program is in a TP file in $FILE_PATH	TPSTOR_FILE

Programs in SHADOW behave the same way as CMOS. There is no difference in the user model. There is a time lag when a user first EDIT's a program because it is moved to CMOS for editing. This is noticeable if the program is large. For example a 1000 point program takes about 1 second to be moved into CMOS for editing. Other operations such as program header updates do not require that the program be moved to CMOS so there is no time lag.

Programs in SHADOW ONDEMAND differ from SHADOW programs in that they are loaded when required. This can save startup time as it may take several minutes to load several thousand SHADOW programs. The system will never unload SHADOW ONDEMAND programs. So the system must have enough DRAM available to load ALL SHADOW ONDEMAND programs. The "total" amount of DRAM allocated to ONDEMAND programs is tracked.

Programs in FILE and SHADOW ONDEMAND will show a time lag the first time that they are accessed for execution (or any other access). The entire program tree is loaded on the first access. This means that call statements do not take any longer. Also, FILE programs will take longer to load than SHADOW ONDEMAND programs.

When a CMOS program calls a FILE or SHADOW ONDEMAND program there will be a time lag the first time it is called. This is because the call is the first access to that ON DEMAND resource. This is not expected to be a typical use of the ONDEMAND feature.

Table 5–6. FILE Programs Limitations and Attributes

The file is just an ordinary TP file that can be shared among robots or even directly shared with ROBOGuide.
Programs of this type can be edited using the TP editor just like other programs.
Dynamic changes to header information temporarily move a program from FILE to CMOS to do these adjustments.
Programs of type FILE may not be compatible with certain applications

Certain utilities like “Program Adjust” and “On the fly” do not work efficiently with FILE programs. These utilities are designed to be able to modify the program while it is running. The system must temporarily move a program from FILE to CMOS to do these adjustments and then move the program back to FROM. It is better to use the SHADOW and SHADOW ONDEMAND storage for applications requiring constant adjustments.

In addition there are a number of “temporary” states that a program can be in. It is temporary because the system will in some cases move the program back to its permanent state. These states are:

Table 5–7. Temporary Program Storage States

Temporary State	Meaning
SHAD (DRAM) ONDE	SHADOW ONDEMAND program is loaded to DRAM
FILE (DRAM)	FILE program is loaded into DRAM
SHAD(CMOS)	SHADOW program moved to CMOS for editing
SHAD(CMOS) ONDE	SHADOW ONDEMAND program moved to CMOS for editing
FILE (CMOS)	FILE program moved to CMOS for editing

Some of the operational modes associated with the new types have some time lags due to loading programs from a file or moving programs to CMOS. The rule of thumb is 1ms/position. However, the time will vary depending on a number of circumstances. There is a section later on devoted timing issues.

Table 5–8. Load/Convert Times for 1,000 Point Program

Source	Destination	Time (ms)
SHADOW ONDEMAND	DRAM	112
DRAM	CMOS	1,282
CMOS	DRAM	1,116

Table 5–8. Load/Convert Times for 1,000 Point Program (Cont'd)

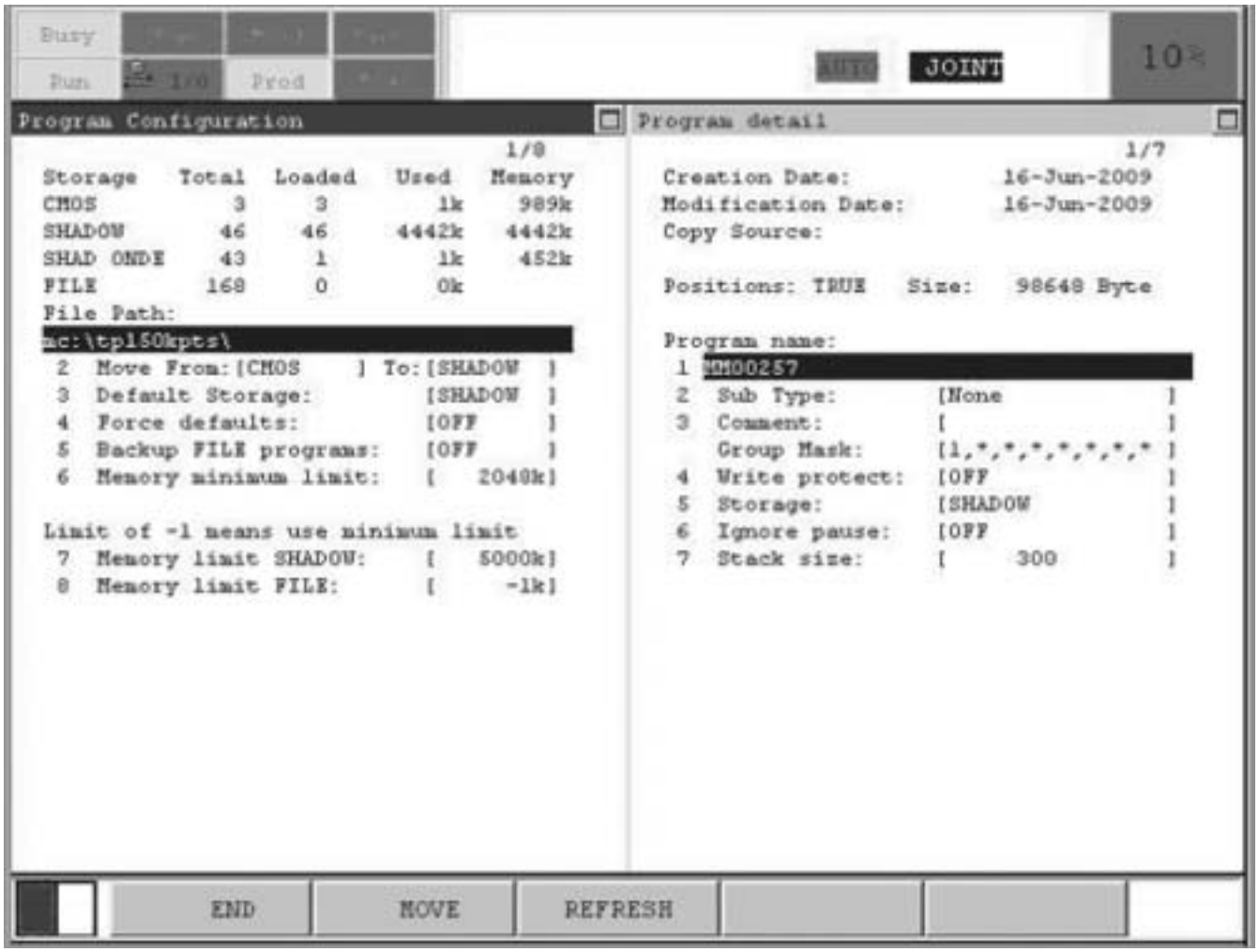
Source	Destination	Time (ms)
FILE (PC Share)	FILE(DRAM	2,251
FILE (MC)	FILE(DRAM	2,791
FILE (FR)	FILE(DRAM	2,838
FILE (RD)	FILE(DRAM	1,300
FILE (USB)	FILE(DRAM	2,904

5.5.1 User Interface

In the select screen a new [ATTR] setting is provided which displays the current program storage attribute. This can be a permanent or temporary attribute. It is the actual current storage state of a program.

New settings are provided to show the status of programs in general and specific storage state per program.

Figure 5–10. Configuration/Detail Screens



Select Detail

In the detail menu there is a new entry for STORAGE. This allows the user to set the storage state to one of the “permanent” storage states of CMOS, SHADOW, SHADOW ONDEMAND and FILE. This entry is only available when the TP DRAM option is loaded. The detail screen allows the user to set and display the storage attribute in addition to other program parameter. The user may only set permanent storage states but the display may reflect a temporary storage state. For example if the user sets the storage state to FILE the state will show up as FILE(DRAM) because the program must be in DRAM in order to set the storage state.

Storage Configuration Screen

There is a new screen that is accessible from the [TYPE] pull-up in the select screen. Traditionally the [TYPE] key has not been used to pull up associated screens. In this case it is a very good fit to SELECT screen functionality. This MENU is only available when the TP DRAM option is loaded

This form may be displayed at power up if there are many SHADOW programs that take time to load. In this case the user can continue to normal menus. This function is ONLY available at COLD start.

The [TYPE] entry is:

Configure…

This will pull up a new form that allows the user the ability to display the following information:

Table 5–9. Storage Configuration Status and Settings

Type	Information
Status	Total number of programs in CMOS,SHADOW,SHADOW ONDEMAND, FILE
Status	Loaded programs in SHADOW,SHADOW ONDEMAND, FILE This is dynamic on startup and will show load progress
Status	Total memory use of programs in CMOS,SHADOW,SHADOW ONDEMAND, FILE
Setting	String which is validated. If changed FILE programs list is updated according to new setting ($TPDRAM_CFG.$FILE_PATH)
Setting	To and from locations for moving programs between storage types
Setting	CMOS,SHADOW or SHADOW ONDEMAND. Designates the default storage location for new programs ($TPDRAM_CFG.$TPSTORE_DEF)
Setting	Default setting overrides setting in TP file. ($TPDRAM_CFG.$FORCE_DEF = 0 or 1)
Setting	Exclude or include FILE program types in backup. ($TPDRAM_CFG.$BACK_FILE = 0 or 1)
Setting	SHADOW non-volatile storage size ($TPDRAM_CFG.$SHAD_SIZE)
Setting	FILE non-volatile storage size ($TPDRAM_CFG.$FILE_SIZE)
Setting	Minimum allowable DRAM (does not allow remaining DRAM to go below this limit) ($TPDRAM_CFG.$MIN_DRAM)
Function	MOVE — Move programs according to specified storage types
Function	REFRESH — Update the information according to the FILE storage TP files
Function	END — Continue to SELECT screen or normal TPMM operation at startup.

This screen is also shown on startup if program loading is still in progress when the teach pendant is ready. In this case the user can “watch” the shadow programs load. If he hits END while the programs are still loading the pendant will move on to normal operations.

5.5.2 Maximum Program Size

The largest TP program that can be created is limited by the amount of CMOS. This is because program creation (no matter HOW the program is created) relies on the entire program residing in CMOS.

Table 5–10. Maximum Program Size for Typical CMOS Allocations

CMOS Size	Number of Points	CMOS Allocation
1 Meg	2,000	240k
2 Meg	8,800	800k
3 Meg	12,600	1000k

For other tools there is a similar or same limitation. However, the tool sometimes changes these limits. No enhancements are planned at this time to eliminate this limitation.

The maximum number of programs that can be stored in on the controller is 8,000. With FILE storage types it is possible to reach that 8,000 file limit. If the files are relatively small it can be reached but it is unlikely.

5.5.3 KAREL Control of STORAGE

KAREL programs can access and set the permenant storage for TP programs via:

SET_ATTR_PRG(progname, AT_STORAGE, TPSTOR_SHAD, ’’, status)

KAREL programs can retrieve the current storage state via:

GET_ATTR_PRG(progname, AT_STORAGE, this_storage, ’’, status)

See the KAREL reference manual for more information on the usage of these built-ins.

5.5.4 File Formats

The format of the TP file is unchanged from previous versions. Programs created on V7.50 can be loaded on V7.40 and vise versa.

The format of the LS file is more restrictive. Legacy LS files do not specify STORAGE and can be loaded on V7.50. They will take on the default storage just like a legacy TP file. Storage is explicitly specified in the LS file. So LS files created on V7.50 cannot be loaded into V7.40 unless the user edits the LS file to remove the STORAGE specification.

The LS file will only contain the “permanent” storage state. So for example of a file is SHAD(CMOS) at the time that the LS file is output the LS file will contain SHADOW. ASCII to binary conversion will only recognize the “permanent” storage states of CMOS, SHADOW, SHADOW ONDEMAND and FILE.

5.5.5 Storage When Loading from a File

On load the controller must make a decision where to put a TP program. In general the storage attribute is stored in the TP file. For legacy TP and LS files this is not the case.

If the storage is not specified because this is a legacy TP file the program will be stored according to the setting of $TPDRAM_CFG.$TPSTORE_DEF (see configuration screen). The default setting for this is SHADOW. So in general legacy programs are loading into SHADOW. The user can set this to be CMOS, SHADOW or SHADOW ONDEMAND.

FILE storage does not behave this way. The file storage attribute is based on the presence of the TP file in the directory specified by $TPDRAM_CFG.$FILE_PATH(see configuration screen). A program will take on the FILE attribute based on the fact that it is loaded via the file mechanism.

The user also has the option to load all programs into the default storage regardless of what the TP file setting is. Setting $TPDRAM_CFG.$FORCE_DEF(see configf screen) will force all TP pgormas to be laoded into the default storage.

5.5.6 Backup and restore

Any SHADOW or SHADOW ON DEMAND programs are treated exactly the way that CMOS programs are with respect to backup and restore. The FROM file components of these programs are hidden and the only TP files that result from these programs types are in the file backup.

All FILE programs are NOT part of backup and restore unless specified $TPDRAM_CFG.$BACK_FILE (see configuration screen). It is up to the user to back these up as files.

TP files that are stored in FILE do not appear on MDB: at all. They do appear on MD: and the corresponding LS files appear on MD:. So the user is expected to copy TP files from MDB:. Any user who deals directly with MD: will get all the TP programs via MD:. This means that the TP files are loaded from the FILE store and then saved to the backup location.

Restore is similar. Any FILE program TP files are put in the $TPDRAM_CFG.$FILE_PATH directory and that’s it. If they are loaded using the standard restore mechanism they wil be retored if TPDRAM_CFG.$FILE_PATH is properly set up.

In general loading a TP program that is a FILE type results in a TP file being created in $TPDRAM_CFG.$FILE_PATH. If the path is invalid then an error will result

5.5.7 Issues relating to SHADOW programs

The main issue here is that SHADOW programs take time to load. Therefore they are loaded at the very end of startup. The controller is allowed to complete its general power up sequence and the programs continue to load at a rate of about 1000pts/minute.

If the system is up and ready to go into the standard TPMM menu it will display a status screen Indicating that it is still loading programs. This is the same screen which is available as part of the SELECT menus (under the type screen). The user can force this screen to exit while programs are loading or wait until programs are loaded.

However, at this time program stubs are created and these programs are treated as “ONDEMAND”. So any access to these programs during the load will work. It just may incur unexpected latencies.

The system variable $TPDRAM_CFG.$LOADED can be used to monitor the progress of the load from an application. This will initially have the value 0 but will be 100 when all programs are loaded.

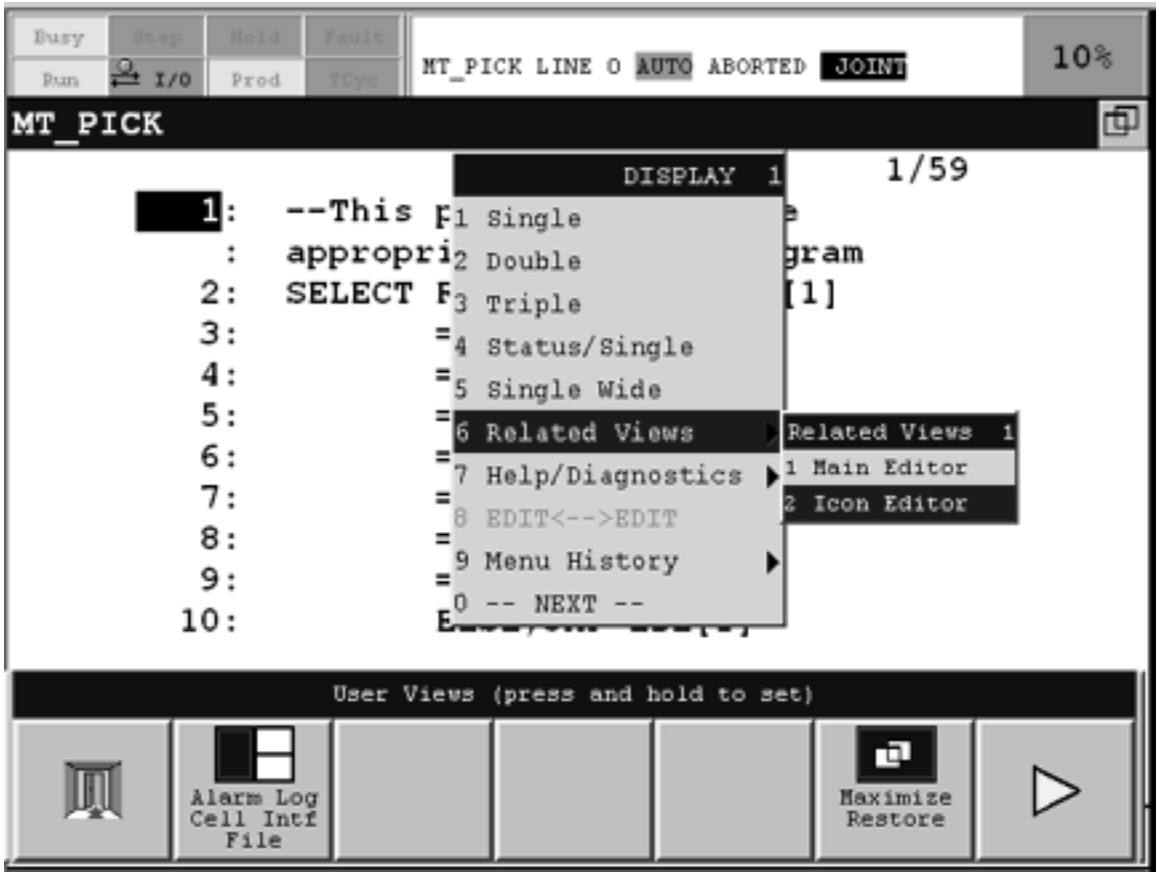
5.6 ICON EDITOR

If you have the optional touch screen available on your *i*Pendant, you can edit a program using the icon editor. Use Procedure 5-7 to enter the Icon Editor.

Procedure 5-7 Enter Icon Editor

1. Edit the program in the primary pane.
2. Press SHIFT and then press DISP key.
3. Select Related Views. Select Icon Editor.

Figure 5–11. Editor Related Views



The Editor will switch to an alternative mode which supports touch editing of icons. When you select a different menu and return to the editor, the Icon Editor will be remembered. To exit the Icon Editor and return to the Main Editor, press the EXIT icon. If you do not have the optional touch screen available, use Procedure 5-8 to exit the Icon Editor.

Procedure 5-8 Exit Icon Editor

1. Press SHIFT and then press DISP key.
2. Select Related Views. Select Main Editor.

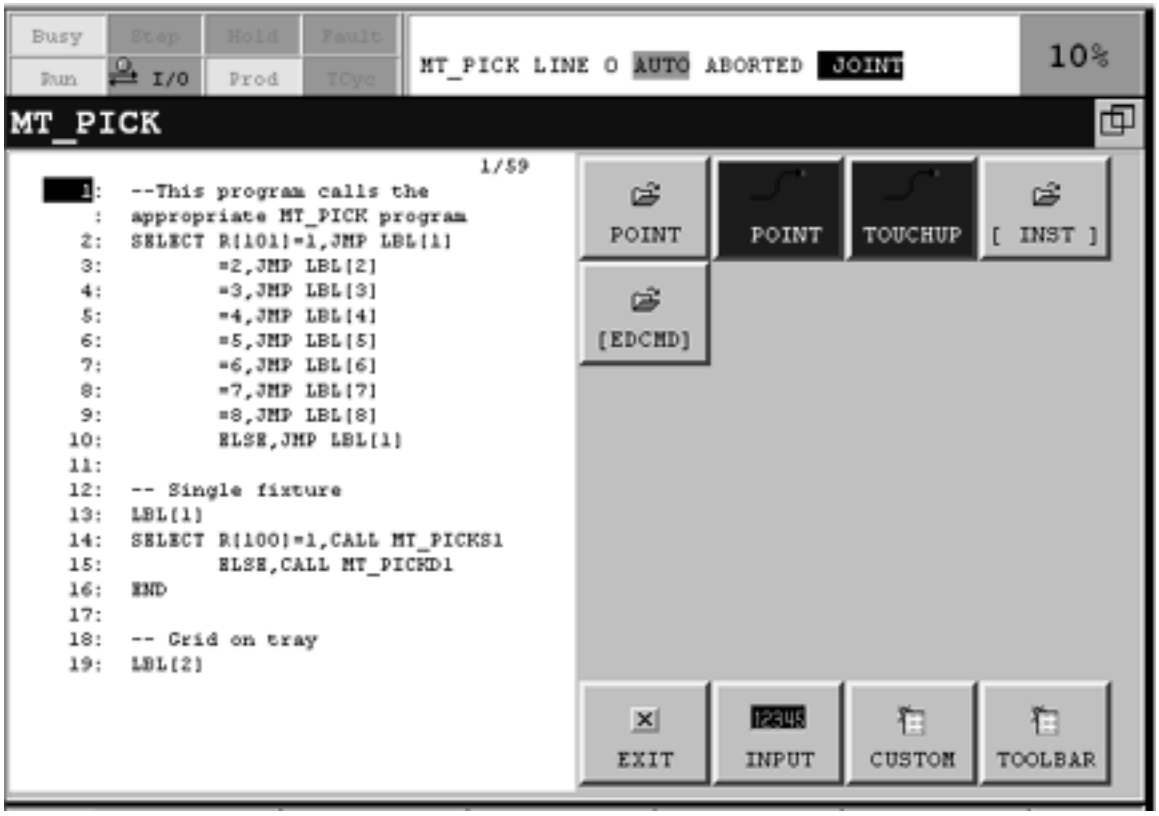
The Main Editor will now be used.

5.6.1 Using the Icon Editor

The left side displays the lines in the program. You may use touch events to select a line. Touching above the first line will scroll up and touching below the last line will scroll down.

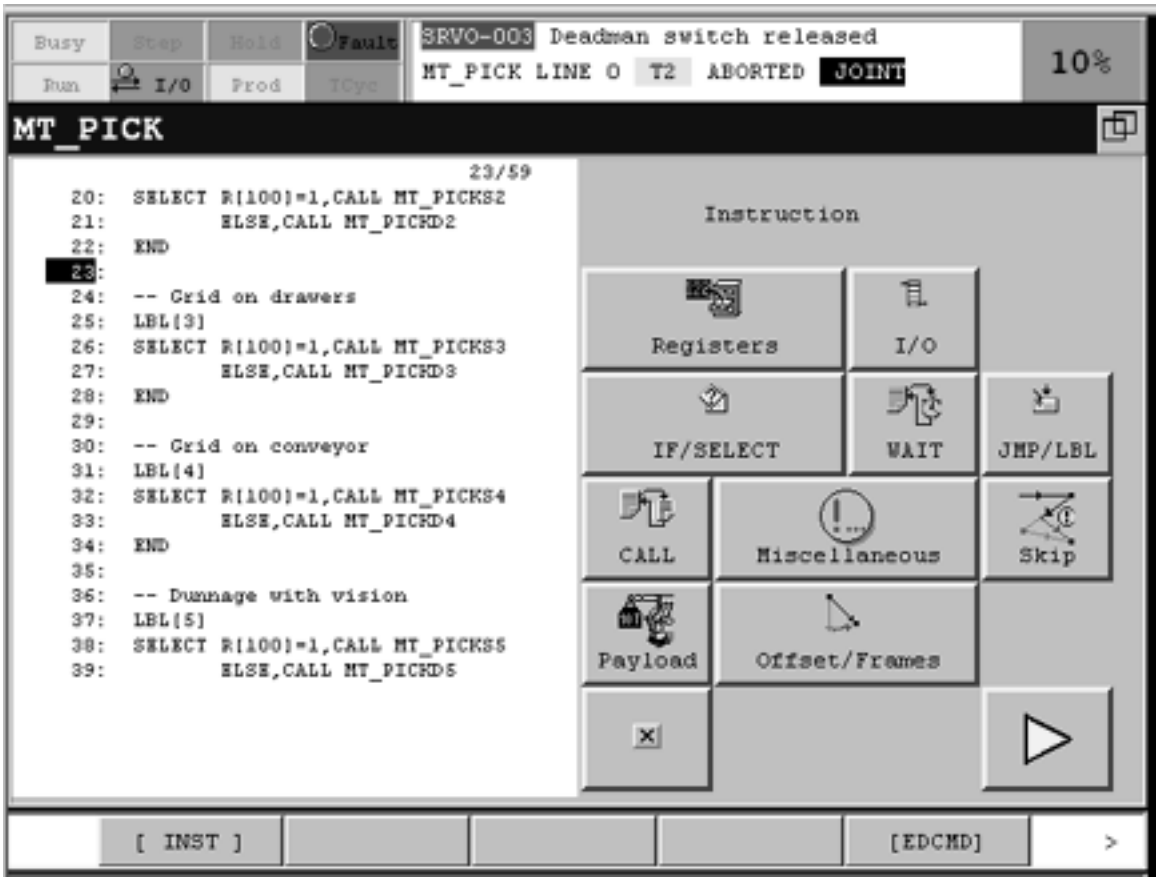
When the cursor is on the line number, the right side will display the valid selections. The blue icons are normally shift operations although shift is not required. See Figure 5–12

Figure 5–12. Icon Editor



The open icons will display another set of icons when pressed. For instance, when [INST] is pressed, Figure 5–13 is shown.

Figure 5–13. Instruction Icons

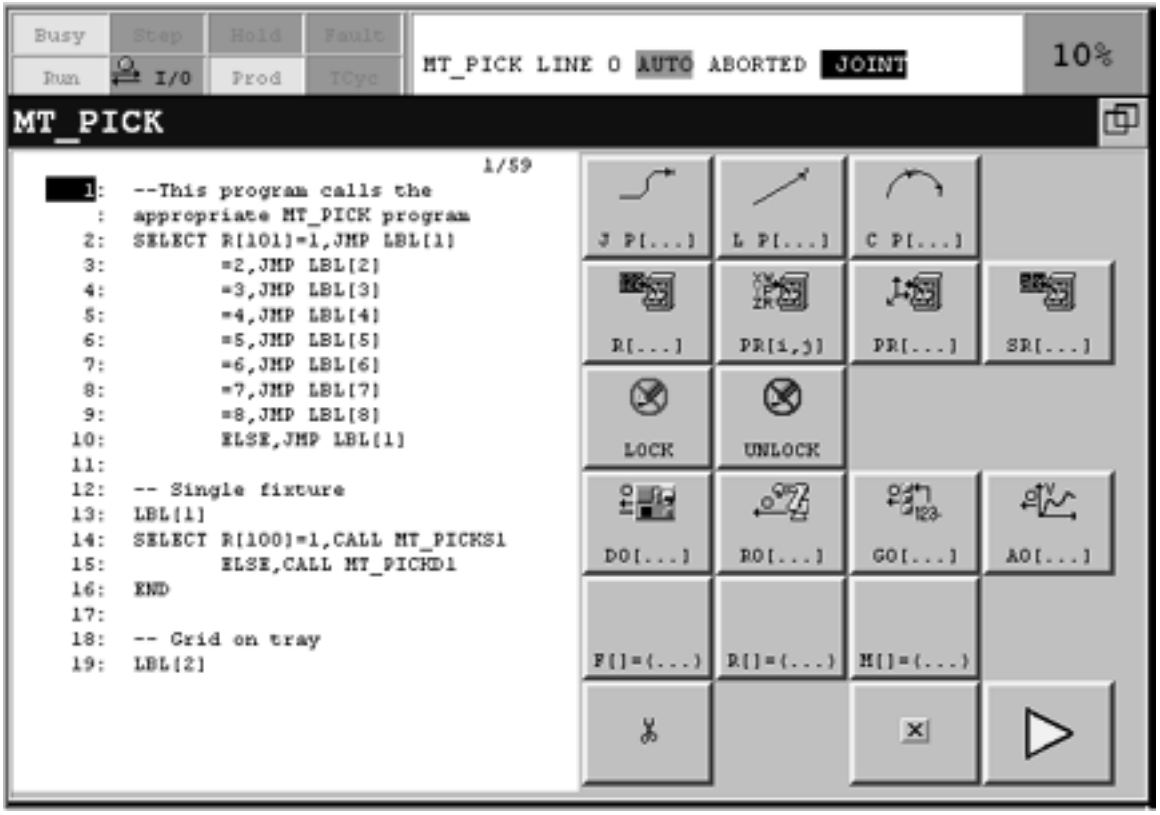


The close icon on the bottom will close the Instruction icons. The right arrow displays another page of instructions.

When the cursor is on the line number and the statement contains a call program, then pressing enter will edit the called program instead. While you are in the called program, you may press prev on any line number to return. Also pressing on the End statement will return. Up to 5 levels of called programs are allowed.

When the cursor is on the line number, the right side will display 3 additional icons along the bottom if the "Ascii Upload" option is loaded. Pressing the TOOLBAR icon displays a selection of icons that allow you to quickly add new instructions to the program. Figure 5–14 shows the TOOLBAR icons.

Figure 5–14. TOOLBAR Icons

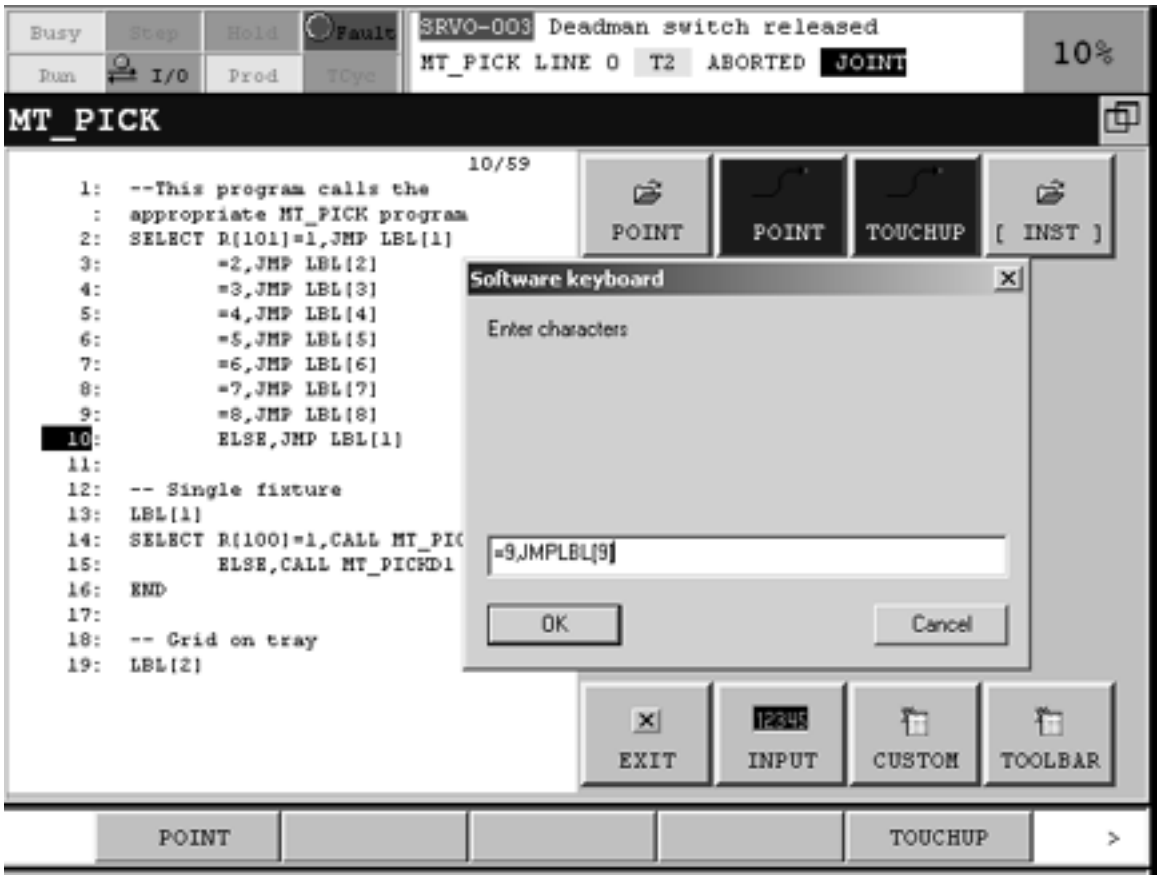


On the bottom row, the scissor icon allows you to quickly delete an instruction without confirmation. The close icon will close the TOOLBAR icons. The right arrow displays another page of icons.

Pressing the CUSTOM icon displays a selection of icons specific to your application. They can be customized as described in Section 5.6.2 .

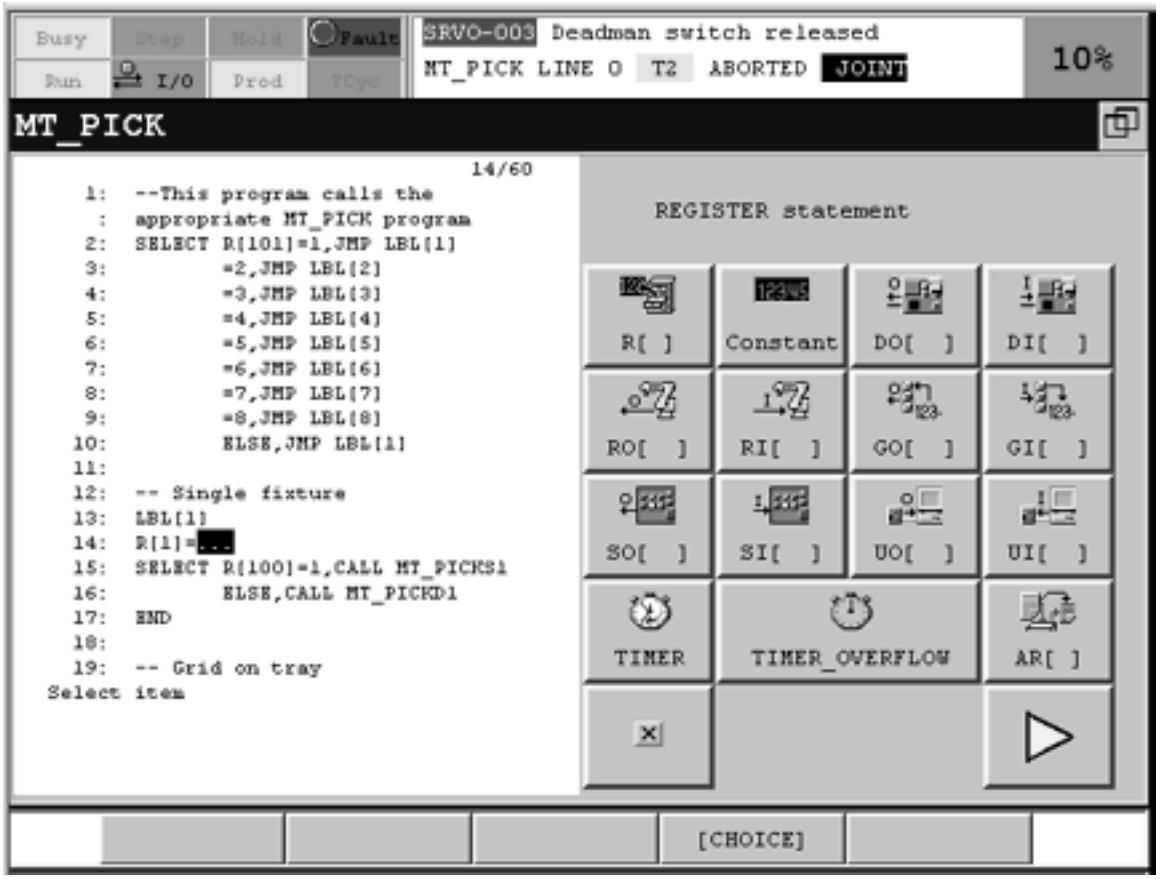
Pressing the INPUT icon allows you to insert 1 line into the program using the keyboard. The syntax must be correct. If not, an alarm will be posted and the line will not be inserted. See Figure 5–15 .

Figure 5–15. INPUT Icon



When you touch an item in a line, the right side will display icons that are valid selections for editing that item. See Figure 5–16 .

Figure 5–16. Context Sensitive Icon Editing



5.6.2 Configuring the Custom Toolbar

The CUSTOM Toolbar is only available when the “Ascii Upload” option is loaded. Pressing CUSTOM brings up a custom toolbar which can be configured using the XML file, FR:\TPEINS.XML. This file will be backed up and restored as Application Data.

An example syntax for a toolbar button is shown below:

```
<!-- Require Multi-Pass -->
<BUTTON pos='24'
REQUIRE='MPAS'
TEXT='MP OFFSET'
TEXTKN='Insert KANJI here'
IMAGE='mp_ofst.gif'
LINE='MP Offset PR[...] RPM[..]'
/>
```

- The BUTTON element begins a new button. The pos attribute is optional. It is the relative position of the button when it is displayed. The toolbar will reorder the buttons based on the position number. When the position number changes, it forces a line feed. If the pos attribute is omitted, then it defaults to 0 which is the top of the list.
- The REQUIRE element is optional. It is the entry in $FEATURE.$MOD[] which must exist for the button to display. There can be multiple REQUIRE statements which are OR'd together. Typically a single REQUIRE statement is sufficient if the order name which loads the instruction is used, such as 'MPAS' for Multi-Pass. If the REQUIRE element is omitted, then the button is always displayed.
- The TEXT element is optional. It is the text shown on the button for any languages which are not explicitly stated. If the TEXT element is omitted, then the button has no default text.
- The TEXTKN element is optional. It is the text for Japanese and KANJI languages. It supports either Katakana or KANJI characters. Any language can be specified using TEXTXX where XX is the language suffix. When the language changes, the editor is notified and the toolbar changes the language automatically. Other language suffixes supported are "FR", "GR", "SP", "CH", "TW", and "OT" for other.
- The IMAGE element is optional. It is the button icon. Icons are typically installed in FRH:\GUI\ directory. If the icon is elsewhere the path can be specified, such as '/frh/cgtp/tcyc_sb.gif'. If the IMAGE element is omitted, then the button has no icon.
- The LINE element is the ascii line which will be inserted into the program. It should be in English. This works even if English is not loaded on the controller. If the LINE element is omitted, then the button does nothing when pressed.

Chapter 6

PROGRAM ELEMENTS

Contents

Chapter 6	PROGRAM ELEMENTS	6–1
6.1	OVERVIEW	6–5
6.2	PROGRAM HEADER INFORMATION	6–7
6.2.1	Overview	6–7
6.2.2	Creation Date	6–12
6.2.3	Modification Date	6–13
6.2.4	Copy Source	6–13
6.2.5	Positions and Program Size	6–13
6.2.6	Program Name	6–13
6.2.7	Sub Type	6–14
6.2.8	Program Comment	6–18
6.2.9	Group Mask	6–18
6.2.10	Write Protection	6–19
6.2.11	Ignore Pause	6–20
6.2.12	Stack Size	6–20
6.2.13	Application Mask	6–20
6.2.14	Program Type	6–21
6.2.15	Multi-Arm Main	6–22
6.2.16	Head of Family	6–22
6.2.17	Family Members	6–22
6.3	LINE NUMBER AND PROGRAM END MARKER	6–22
6.4	MOTION INSTRUCTION	6–22
6.4.1	Overview	6–22
6.4.2	Motion Type	6–23
6.4.3	Positional Information	6–28
6.4.4	Position Confirmation	6–29
6.4.5	Motion Status Display	6–30
6.4.6	Frame Number of Positional Data	6–40
6.4.7	Switch Frame Check Type	6–41
6.4.8	Speed	6–42

6.4.9 Termination Type 6–50
6.4.10 Motion Options 6–52
6.4.11 Correspondence Between Teach Pendant Program Motion and KAREL Program Motion 6–117
6.5 REGISTER INSTRUCTIONS 6–118
6.6 POSITION REGISTER INSTRUCTIONS 6–123
6.6.1 Overview 6–123
6.6.2 PR[x] Position Register Instructions 6–123
6.6.3 PR[i,j] Position Register Element Instructions 6–125
6.7 STRING REGISTER INSTRUCTIONS 6–128
6.7.1 Overview 6–128
6.7.2 String Register Assignment and Concatenation 6–128
6.7.3 String Conversion and Precedence 6–130
6.7.4 STRLEN Instruction 6–131
6.7.5 FINDSTR Instruction 6–131
6.7.6 SUBSTR Instruction 6–132
6.8 INPUT/OUTPUT INSTRUCTIONS 6–134
6.8.1 Overview 6–134
6.8.2 Digital Input and Output Instructions 6–134
6.8.3 Robot Digital Input and Output Instructions 6–136
6.8.4 Analog Input and Output Instructions 6–137
6.8.5 Group Input and Output Instructions 6–138
6.9 BRANCHING INSTRUCTIONS 6–139
6.9.1 Overview 6–139
6.9.2 Label Definition Instruction LBL[x] 6–140
6.9.3 Unconditional Branching Instructions 6–140
6.9.4 Conditional Branching Instructions 6–142
6.10 WAIT INSTRUCTIONS 6–146
6.11 MISCELLANEOUS INSTRUCTIONS 6–151
6.11.1 Overview 6–151
6.11.2 RSR Enable/Disable Instruction 6–151
6.11.3 User Alarm Instruction 6–151
6.11.4 Timer Instruction 6–152
6.11.5 OVERRIDE Instruction 6–152
6.11.6 Remark Instruction 6–153
6.11.7 Multi-lng Remark Instruction 6–153
6.11.8 Line Remark (Comment out) Instruction 6–154
6.11.9 Message Instruction 6–155
6.11.10 Parameter Name Instruction 6–155
6.11.11 Maximum Speed Instruction 6–157
6.12 SKIP INSTRUCTION 6–159
6.13 QUICK SKIP (HIGH-SPEED SKIP) 6–162
6.13.1 Overview 6–162
6.13.2 Execution 6–162
6.13.3 Limitations 6–163
6.14 OFFSET/FRAME INSTRUCTIONS 6–163
6.15 TOOL OFFSET CONDITION INSTRUCTION 6–166
6.16 MULTIPLE CONTROL INSTRUCTIONS 6–167
6.17 MACRO COMMAND INSTRUCTION 6–168

6.18 PARAMETERS FOR PROGRAM CALL AND MACRO INSTRUCTIONS 6–169
6.18.1 Overview 6–169
6.18.2 Parameter Instruction Syntax 6–170
6.18.3 String Value Selections 6–172
6.18.4 Argument Registers 6–174
6.18.5 Guidelines for Using Parameters 6–175
6.18.6 Including Parameters in Program Call and Macro Instructions 6–177
6.18.7 Including Argument Registers in Sub-Programs 6–181
6.19 PROGRAM CONTROL INSTRUCTIONS 6–182
6.19.1 Overview 6–182
6.19.2 PAUSE Instruction 6–183
6.19.3 ABORT Instruction 6–183
6.19.4 Error Program Instruction 6–183
6.19.5 Resume Program Instruction 6–184
6.19.6 Maintenance Program Instruction 6–184
6.19.7 Clear Resume Program Instruction 6–185
6.19.8 Return Path Disable Instruction 6–185
6.20 MOTION GROUP INSTRUCTIONS (OPTION) 6–186
6.20.1 Overview 6–186
6.20.2 Independent Motion Group Instructions 6–186
6.20.3 Simultaneous Motion Group Instructions 6–186
6.20.4 Defining Motion Group Instructions 6–187
6.21 POSITION REGISTER LOOK-AHEAD INSTRUCTIONS 6–189
6.22 CONDITION MONITOR INSTRUCTIONS 6–190
6.23 PAYLOAD INSTRUCTION 6–194
6.24 COLLISION GUARD INSTRUCTIONS (OPTION) 6–196
6.25 BASIC PROCESS AXES INSTRUCTIONS (OPTION) 6–197
6.25.1 Overview 6–197
6.25.2 SET ISDT SPEED Instruction 6–198
6.25.3 STOP ALL ISDT Instruction 6–199
6.26 PALLETIZING INSTRUCTIONS 6–200
6.26.1 Overview 6–200
6.26.2 PALLETIZING-B Instruction 6–203
6.26.3 PALLETIZING-BX Instruction 6–209
6.26.4 PALLETIZING-E Instruction 6–216
6.26.5 PALLETIZING-EX Instruction 6–224
6.26.6 PALLETIZING-END Instruction 6–232
6.27 ALL-POINT TEACHING FOR PALLETIZING 6–232
6.28 PALLET REGISTER INSTRUCTIONS 6–235
6.29 TOUCH SENSE INSTRUCTIONS 6–237
6.29.1 Overview 6–237
6.29.2 Search Start Instruction 6–238
6.29.3 Search End Instruction 6–239
6.29.4 Touch Offset Instruction 6–239
6.29.5 Touch Offset End Instruction 6–240
6.30 MIXED LOGIC INSTRUCTIONS 6–241
6.30.1 Overview 6–241
6.30.2 Data Types 6–241
6.30.3 Operators 6–242
6.30.4 Expressions 6–243

6.30.5 Adding Mixed Logic Instructions 6–245
6.30.6 Background Logic 6–247
6.30.7 Backup/Restore 6–260
6.31 PROCESS SYNCHRONIZATION 6–260
6.32 VISION INSTRUCTIONS 6–263
6.32.1 Overview 6–263
6.32.2 RUN_FIND 6–264
6.32.3 GET_OFFSET 6–264
6.32.4 SET_REFERENCE 6–265
6.32.5 CAMERA_CALIB 6–265
6.32.6 VR[].OFFSET 6–265
6.32.7 VR[]. MODELID 6–266
6.32.8 SAMPLE PROGRAM 6–266

6.1 OVERVIEW

A *program element* is a component of a program. A *teach pendant program* is a series of program elements selected and organized to perform an application.

For dispensing systems with multiple equipment, and PaintTool systems, there are two kinds of programs: process programs and job programs. A process program directs the robot to perform a specific task. A job program contains several processes.

For dispensing systems with single equipment, you can use the regular type (none type) programs.

Figure 6–1 shows some of the program elements of a typical palletizing program.

Figure 6–1. Program Example

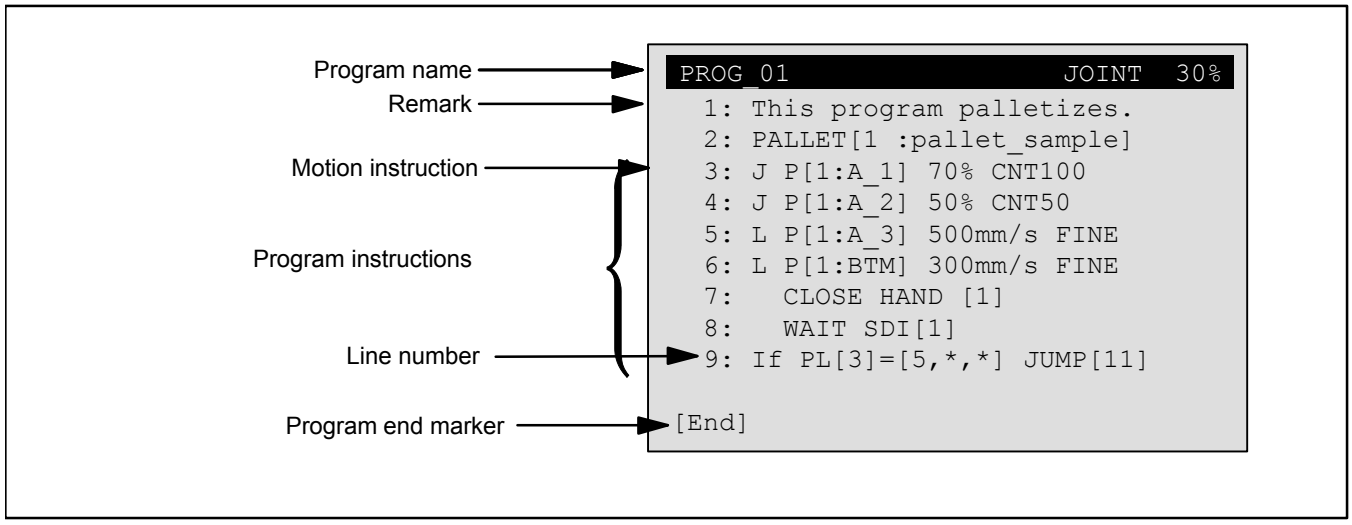
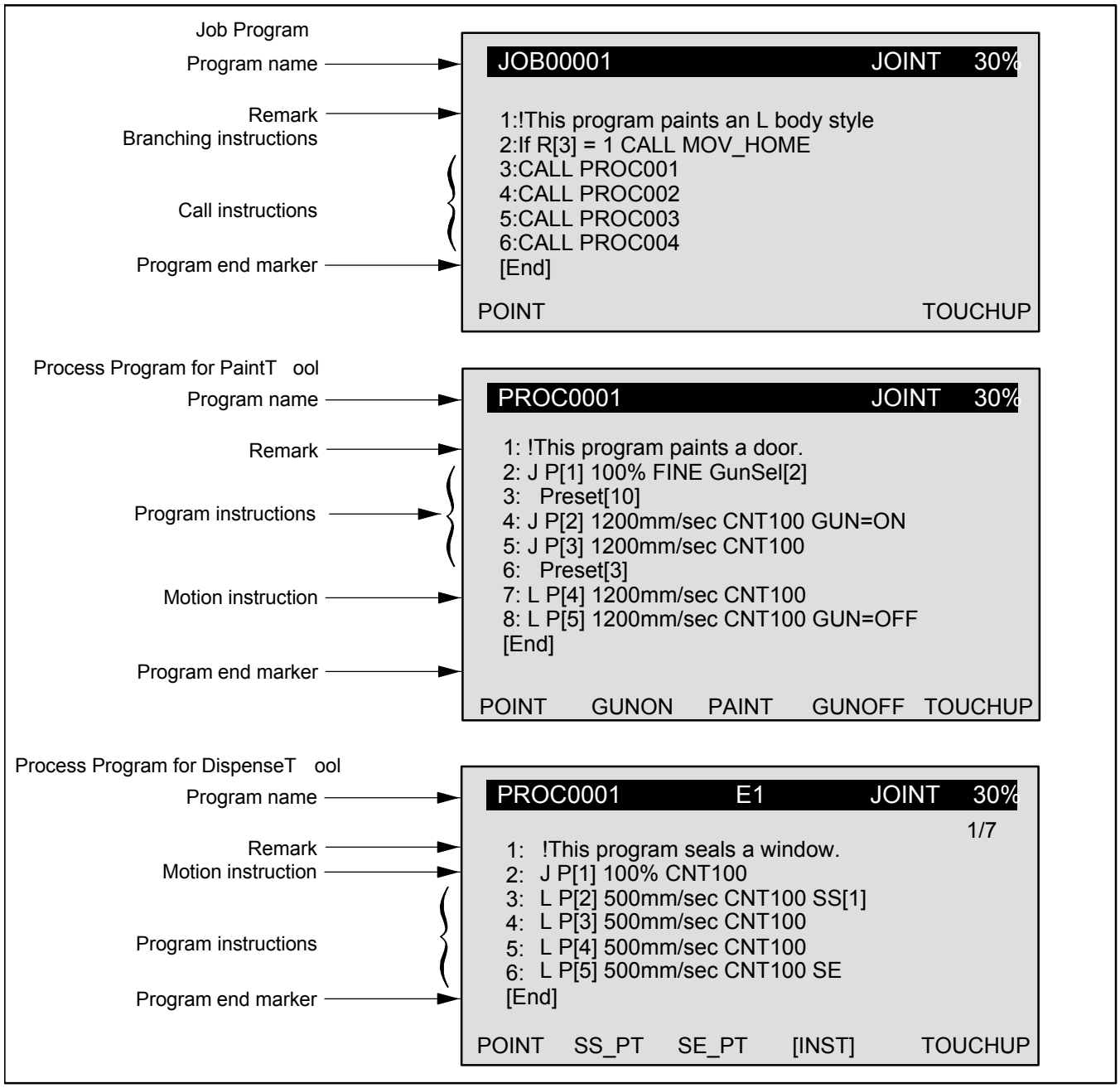


Figure 6–2 shows some of the program elements of a typical process program and a typical job program for a Painting or Dispensing application.

Figure 6–2. Program Example



A program consists of the following program elements:

- Program header information, which can include information such as program name, comment, group mask, program type, application mask, write protection setting, cycle time, gun on time, and material volume

- In PaintTool or DispenseTool, a process program contains UTOOL and FRAME number entries, tracking schedule, and boundary field when using the Line/Rail Tracking option
- Line numbers, assigned to each program instruction
- Motion instructions, which include commands that tell the robot where and how to move
- Program instructions for logic, I/O, data handling, program control, advanced functions, and so forth.
- Remarks to annotate the program.
- Program end marker, indicating that there are no more instructions in the program.

6.2 PROGRAM HEADER INFORMATION

6.2.1 Overview

Program header information is specific information that identifies and classifies the program.

Table 6–1 lists SELECT screen items and operations.

Table 6–2 lists the SELECT DETAIL screen items and operations. The items displayed on your screen will vary.

Table 6–1. Select Screen Items and Operations

ITEM	DESCRIPTION
No.	This item is the line number.
Program name	This item is the name of the program.
Attribute field	This item is a descriptive field for each program. This field can be any of the following: Comment, Protection, Last Modified, Size, or Copy Source.
CREATE	Press this key to display the Create Teach Pendant Program screen from which you can create a new program listing. When you create a new program, it will be added to the Program name listing.
DELETE	Press this key to delete a program.
MONITOR	Press this key to monitor a program that is currently running.
ATTR	Press this key to select the appropriate attribute to display.
COPY	Press this key to display the Copy Teach Pendant Program screen from which you can copy one file to another.
DETAIL	Press this key to key to display the Program DETAIL screen.

Table 6–1. Select Screen Items and Operations (Cont'd)

LOAD	Press this key to display the Load Teach Pendant Program screen from which you can load the selected program.
SAVE AS	Press this key to save the selected program. You can specify the device to which the program is saved. You can also specify a new name if you want to save the file with a different name.
PRINT	Press this key to display the Print Teach Pendant Program screen from which you can select a program to print to the selected device.

Table 6–2. Program DETAIL Screen Items and Operations

ITEM	DESCRIPTION	AVAILABLE IN JOB OR PROCESS PROGRAMS ONLY	AVAILABLE IN JOB PROGRAMS ONLY	AVAILABLE IN PROCESS PROGRAMS ONLY
Creation Date	This item is the date on which the program name was created.	Yes		
Modification Date	This item is the date when the file was last displayed in the editor.	Yes		
Copy Source	This item is the name of the file from which the file was copied.	Yes		
Positions	This item indicates whether the program contains recorded robot positions.	Yes		
Size	This item is the size of the program in bytes.	Yes		
Program name	This item is the name of the program.	Yes		
Sub Type	This item is the kind of program you want to write. This value can be one of the following: None, Macro, or Cond (Condition Handler).	Yes		
Comment	This item is a description field for the selected program.	Yes		

Table 6–2. Program DETAIL Screen Items and Operations (Cont'd)

Group Mask	This item identifies the group of axes, or motion group, that the program controls.	Yes		
Write protect	This item identifies whether the program can be modified.	Yes		
Ignore pause	This item indicates whether the program will continue to run when an error occurs, a command is issued, or the teach pendant is enabled.	Yes		
Stack Size	When a program is executed, a stack of 300 long words (1200 bytes) is allocated unless you specify a stack size. The stack is allocated from available user RAM. The minimum and default stack size is 300. The maximum stack size is 4000.	Yes		
Cycle Time	This item displays the cycle time for the most recent execution of the currently selected job or process.		***	
Last Cycle Time	This item displays the cycle time for the second most recent execution of the currently selected job or process.		*	
Gun On Time	This item displays the total amount of time the gun was on for the most recent execution of the currently selected job or process.		*, **	

Table 6–2. Program DETAIL Screen Items and Operations (Cont'd)

Last Gun On Time	This item displays the total amount of time the gun was on for the second most recent execution of the currently selected job or process.		*	
Material Volume	This item displays the volume of the material dispensed in the most recent execution of the currently selected job or process.		*	
Last Material Volume	This item displays the volume of the material dispensed in the second most recent execution of the currently selected job or process.		*	
Default User Frame	This item displays the name of the user frame to be used by the process program.			*, **
Default Tool Frame	This item displays the name of the tool frame to be used by the process program.			*, **
Part ID	This item displays the number of the job being executed.		Yes	
Equipment Number	This item allows you to specify up to two homogeneous dispensing equipments controlled by one robot controller.			*
Line Track Schedule Number	This item displays the current line tracking schedule.			****
Line Track Boundary Number	This item displays the current line tracking boundary set.			****

Table 6–2. Program DETAIL Screen Items and Operations (Cont'd)

Application Mask	This item displays the currently set application in systems with more than one application available.	***		
Program Type	The program type indicates the kind of Multi-Arm program being used. This should always be set to Normal Program.			
Multi-Arm Main	This item indicates the status of the Multi-Arm program in the Family. It can be set to No, independent program, if it is not the main program of the Multi-arm family, Yes, Head of Family if it is the head of a family program, or Yes, Family Member if it is the main program of a family program.			
Head of Family	This item indicates which robot is the head of the family for this job in a Multi-Arm application.			
Family Members	This item indicates which robots are included as family members in a Multi-Arm application.			
END	Press this key when you are finished viewing program information to return to the Select screen.	N/A	N/A	N/A
PREV	Press this key to return to the previous screen.	N/A	N/A	N/A
NEXT	Press this key to display application-specific program detail information.	N/A	N/A	N/A
1	Press this key to add a motion group to your program.	N/A	N/A	N/A

Table 6–2. Program DETAIL Screen Items and Operations (Cont'd)

*	Press this key to disable a motion group for your program.	N/A	N/A	N/A
ON/OFF	Press these keys to set the appropriate items to ON or OFF.	N/A	N/A	N/A

*These items are shown only for Dispensing systems with multiple equipment, to allow you to set the equipment number for your program.

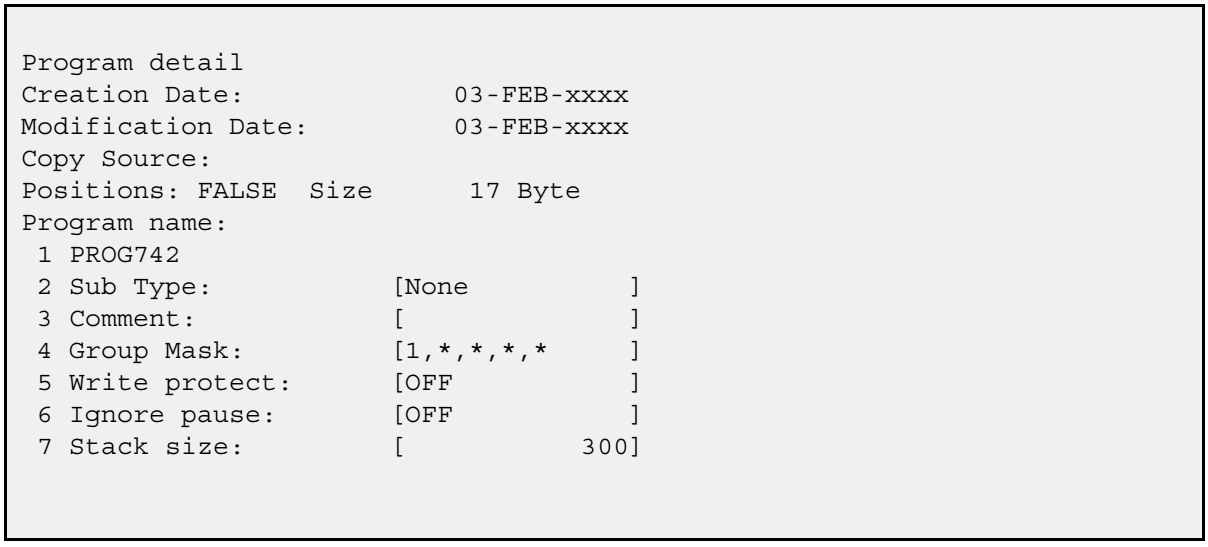
** For PaintTool, this applies to process programs only.

*** This applies to DispenseTool and SpotTool+ only.

**** This applies to DispenseTool and PaintTool only.

Program header information is displayed only the first time you create a program. If you want to view this information again, you must display it by choosing the SELECT menu and pressing the DETAIL key. See the following screen for an example.

Figure 6–3. Program Header Information



The following sections contain details on each kind of program header information.

6.2.2 Creation Date

Creation date is the date on which the program name was created.

6.2.3 Modification Date

Modification date is the date, according to the calendar in the controller, when the file was last displayed in the editor. This information can be displayed using the [ATTR] function key on the SELECT menu.

6.2.4 Copy Source

Copy source is the name of the file from which the file was copied. This field is empty if the file is an original file. This information can be displayed using the [ATTR] function key on the SELECT menu.

6.2.5 Positions and Program Size

The positions item indicates whether the program contains recorded robot positions. When you first create a program, positions is always set to FALSE.

Program size is the size of the program in bytes. The program size can be displayed using the [ATTR] function key on the SELECT menu.

6.2.6 Program Name

A program name identifies the program. When you create a new program, you must give it a unique program name. The program name differentiates the programs stored on the controller.

Length

The name can be from one to 36 characters long.

Available Characters

Letters : A through Z. A program name must start with a letter.

Numbers : 0 through 9; a program name cannot start with a number.

Symbols : _ (underscore) only; **do not use** @ (at), * (asterisk), or space.

Content

The name should be descriptive and it should tell you what the program does.

Note Any program name that is longer than 12 characters is abbreviated on the SELECT screen and displayed with a > at the end. When the program name is selected, the entire program name is displayed at the bottom of the screen.

Note If you have common characters in long program names, you can set $UI_CONFIG.$PROG_COMMON[1–3] to contain the common characters. For instance, if your program name is TEST111_PLANT1_ROBOT15467_A and TEST111 is a common prefix, then set $UI_CONFIG.$PROG_COMMON[1] = TEST111. When the system needs to truncate the program name it will remove the common portion and replace it with a tilde (~) character. See the following screen for an example.

```
Select
               702566 bytes free   7/7
No.  Program name           Comment
 1   -BCKEDT-         [                   ]
 2   HOME_IO          [SET AT HOME I-O    ]
 4   TEST1            [TEST1              ]
 5   ~_PLANT1_ROBOT1>[                    ]
 6   UOP_ROBOT_12>    [                   ]




TEST111_PLANT1_ROBOT15467_A
```

Note If you are writing a program for production operation using RSR or PNS, name the program as follows:

- An **RSR** program must be **RSRnnnn** , where**nnnn** is a four-digit number. For example, if you want your program numbered 23, you would enter RSR0023.
- A **PNS** program must be **PNSnnnn** , where**nnnn** is a four-digit number. For example, if you want your program numbered 23, you would enter PNS0023.

For RSR and PNS programs, use the program comment to indicate what the program does. Refer to Section 6.2.8 .

6.2.7 Sub Type

Sub type identifies the kind of program you want to write. These are:

- None
- Macro
- Cond
- Job *
- Process*

* Process and Job sub types are available on DispenseTool software. They are also available on SpotTool+ software only if the Dispense application has been enabled.

Table 6–3 summarizes the program instructions that can be used in a **DispenseTool** program (or a **SpotTool+** program if the Dispense application has been enabled) that is either a job, process, none, macro, or cond sub type.

Table 6–3. Program Instruction Summary for DispenseTool and SpotTool+ with Dispense Application Enabled

Program Instruction	Available when Sub Type = Job	Available when Sub Type = Process	Available when Sub Type = None	Available when Sub Type = Macro	Available when Sub Type = Cond
Call			•	•	•
I/O	•	•	•	•	
If/Select	•	•	•	•	
Jmp/Lbl	•	•	•	•	
Lock Preg	•	•	•	•	
Macro	•	•	•	•	
Miscellaneous	•	•	•	•	
Monitor/Mon.End			•	•	
Multiple Control	•		•	•	
Offset/Frames		•	•	•	
Payload	•	•	•	•	
Point (Motion)	•	•	•	•	
Program Control	•	•	•	•	
Registers (R[] AND PR[])	•	•	•	•	
Sealing (SS[] AND SE[])		•	•	•	
SKIP	•		•	•	

Table 6–3. Program Instruction Summary for DispenseTool and SpotTool+ with Dispense Application Enabled (Cont'd)

Program Instruction	Available when Sub Type = Job	Available when Sub Type = Process	Available when Sub Type = None	Available when Sub Type = Macro	Available when Sub Type = Cond
Tool_Offset		•	•	•	
Wait	•	•	•	•	
When					•

Table 6–4 summarizes the program elements that can be used in a program that is either a job, process, none, or macro sub type.

Table 6–4. Job and Process Program Element Summary for PaintTool

Program Element	Available when Sub Type = Job	Available when Sub Type = Process	Available when Sub Type = None or Macro
Painting		•	•
Motion		•	•
Offset		•	•
Registers	•	•	•
I/O	•	•	•
Wait	•	•	•
Miscellaneous	•	•	•
Branching (JMP/LBL, IF/SELECT, and CALL)	•		•
Skip		•	•
Macro	•	•	•
Multiple Control	•	•	•
Program Control			•

None

If you select *none*, the program will be created as a .TP program, which can include any instructions in your teach pendant program.

Macro

A *macro program* created as a .MR program can contain any instruction and function as a normal .TP program. However, only macro programs can be set up to be executed in a variety of ways including

from operator panel buttons, teach pendant keys, and the Manual Functions menu. They can also be assigned a name in the macro table and be called with this name in a macro program.

Macro programs can also be called by a program when the MACRO instruction is used. Refer to the "Program Elements" chapter of this manual for more information.

Cond

A "ch" program has a Cond (Condition Handler) sub type. Refer to the "Advanced Functions" chapter of this manual for more information on the condition monitor function.

Job

A job is a program that includes one or more processes. When you run production, you run the job that corresponds to the task you want to perform. This job contains program calls to each of the processes that will be performed during production.

Note In PaintTool, when you write a job program, you can include only the following kinds of program instructions:

- CALL program
- Conditional branching (IF and SELECT)
- I/O and register instructions
- Program end

Typically, a job program is used as the main program. The job program then calls the process program(s) to do the application processing. For any type of dispenser you must use job and process programs in this manner. This is because the job and process header contains some information that is necessary to run the program properly.

Process

A process is a program that directs the robot to perform a specific task.

In PaintTool, you can include any program instructions except branching instructions in a process program.

Eight predefined process programs are created for you in PaintTool:

- Home
- Gun Clean In (CLNIN)
- Gun Clean Out (CLNOUT)
- Bypass

- Purge
- Zero
- Special1
- Special2

These programs are created for you so that you can move to the positions you define in these programs using the Move Menu. Refer to the "Planning and Creating a Program" chapter for details on how to define and use predefined positions.

6.2.8 Program Comment

When you create a new program, you can also add a *program comment* to the name. A program comment includes additional information that you want to further identify the program. This information can be displayed using the [ATTR] function key on the SELECT menu, and also from the DETAIL function key.

Length

One to 16 characters

Available Characters

Letters : a through z and A through Z

Numbers : 0 through 9

Symbols : _ (underscore), @ (at), * (asterisk)

Blank spaces

Punctuation: ; (semicolon),**:** (colon), **"** (quotation marks),**( )** (left and right parentheses), **.** (period)

Content

Should be descriptive and provide additional information, if necessary.

6.2.9 Group Mask

When you create a program, you define the *group mask* that identifies the group of axes, or *motion group*, that the program will control. Motion groups define different groups of axes that can be used for independent pieces of equipment, positioning tables, opening devices, and other axes.

There are eight motion groups available. The controller can operate a maximum of 40 axes, however, only nine axes can belong to a single motion group.

Note In HandlingTool and ArcTool, multiple motion groups must be set up before they can be used. Refer to the *Software Installation Manual*.

If a system has only one motion group, the default motion group is 1. An asterisk indicates the group is not used. You can specify a program to use all eight motion groups, but only two motion groups can perform Cartesian interpolated motion within a single program. If you disable all groups, you cannot add motion instructions to your program.

Note In HandlingTool and SpotTool+, when you add a motion instruction that has a motion type, the following appears on the screen:

```
Group mask:  1, *, *, *, *, *, *, *
```

You cannot change the group mask after you have added motion instructions to your program. You will not be able to select group mask in the program DETAIL screen.

Note When using SpotTool+ with dispense plug-in and multi-group motion all programs that use motion while dispensing are required to use group mask [1,*,*,*,*,*,*,*].

Note For system level macro programs, the group mask cannot be changed.

6.2.10 Write Protection

Write protection allows you to specify whether the program can be modified.

- **When write protection is set to ON,** you cannot add or modify any element in the program. When you have finished creating a program and are satisfied with how it works, you should set write protection to ON so that you or someone else does not modify it.
- **When write protection is set to OFF,** you can create the program and add or modify any element in the program. By default, write protection is set to OFF.

This information can be displayed using the [ATTR] function key on the SELECT menu.



Caution

When write protection is set to ON, you cannot modify any program header information (program name, sub type, program comment, group mask, ignore pause) except write protection.

6.2.11 Ignore Pause

If the program type has been set to NONE, you can use *ignore pause* to specify whether the program will continue to run even when an error occurs, a command is issued (such as pushing EMERGENCY STOP or HOLD), or the teach pendant is enabled. Ignore pause is allowed only in programs that do not have motion groups specified such as an I/O monitoring program. This means that programs that use ignore pause cannot contain any motion instructions.



Warning

If ignore pause is set to ON, the program MUST NOT issue any motion instructions; otherwise, you could injure personnel or damage equipment.

- **When ignore pause is set to ON**, the program continues to run even when an error occurs, a command is issued, or the teach pendant is enabled. This allows the program to continue any monitoring function, such as monitoring I/O.
- **When ignore pause is set to OFF**, the program pauses when an error occurs, a command is issued, or the teach pendant is enabled.

6.2.12 Stack Size

When a program is executed, a stack of 300 long words (1200 bytes) is allocated unless you specify a stack size. The stack is allocated from available user RAM.

If a stack overflow alarm occurs, the program needs more stack. Options such as TCPMate and Cell Finder require that teach pendant programs have increased stack size.

The minimum and default stack size is 300. The maximum stack size is 4000.

6.2.13 Application Mask

If you have enabled more than one application (such as spot welding, dispensing, or material handling) during software installation, you must set the application mask item to the application you want to use in your program. After you specify the application to use in your program, the program instructions and function key labels will change to reflect the application you specified.

Set the application mask as follows:

- To select an application to use in your program, set the application you want to TRUE on the application process DETAIL screen.
- To write a program that includes *no* application instructions, set all applications to FALSE on the application process DETAIL screen. This sets the application process to NONE.

```
1   Spot weld                   False
2   Handling                    False
3   Dispense                    False
```

Note You can use Spot Welding and Material Handling in a program by selecting Spot Welding.

The available applications are defined at Controlled start during software installation. You can enable up to two applications at one time if you have purchased the Multi Application Enable option and have enabled it at Controlled start. Refer to the *Software Installation Manual* for more information.

All teach pendant keys and LED indicators correspond to the **currently selected** application. The currently selected application is displayed in the middle of the screen name line on the teach pendant screen. If more than one application has been enabled during a controlled start, you can toggle between them by pressing FCTN and selecting Change APPL-Tool. The character abbreviations are as follows: DI for dispensing, HT for material handling, and SW for spot welding. Refer to Section for more information on defining the currently selected application. Refer to Section for more information on the teach pendant keys and LED indicators for the currently selected application.

Note If a software application is using the multiple equipment option, then you will see only the first letter of the application in its character abbreviation, and the number of the equipment assigned to it. D1 corresponds to "Dispensing equipment 1," and S2 corresponds to "Spot welding equipment 2."

Calling Programs when Application Masks are Used

The following rules dictate the kinds of programs you can CALL (using the program CALL instruction) when application masks are used:

- A program with an application mask set to NONE can CALL a program that has an application mask set to an application or NONE.
- A program with an application mask set to an application can CALL a program that has an application mask set to NONE.
- A program with an application mask set to an application cannot CALL a program that has a different application mask.

6.2.14 Program Type

The program type indicates the kind of Multi-Arm program being used. This should always be set to Normal Program.

6.2.15 Multi-Arm Main

This item indicates the status of the Multi-Arm program in the Family. It can be set to No, independent program, if it is not the main program of the Multi-arm family, Yes, Head of Family if it is the head of a family program, or Yes, Family Member if it is the main program of a family program.

6.2.16 Head of Family

This item indicates which robot is the head of the family for this job in a Multi-Arm application.

6.2.17 Family Members

This item indicates which robots are included as family members in a Multi-Arm application.

6.3 LINE NUMBER AND PROGRAM END MARKER

A *line number* is automatically inserted next to each instruction you add to a program. If you remove an instruction or move an instruction to a new position in the program, the program instructions will be renumbered automatically so that the first line is always line 1, the second line 2, and so forth. You use line numbers to identify which lines to move, remove, and mark when you modify a program.

The *program end marker* (**[End]**) automatically appears after the last instruction in a program. As new instructions are added, the program end marker moves down on the screen, retaining its position as the last line in the program.

6.4 MOTION INSTRUCTION

6.4.1 Overview

A *motion instruction* directs the robot to move in a specified way to a specific location in the workcell using a specified speed. A motion instruction includes:

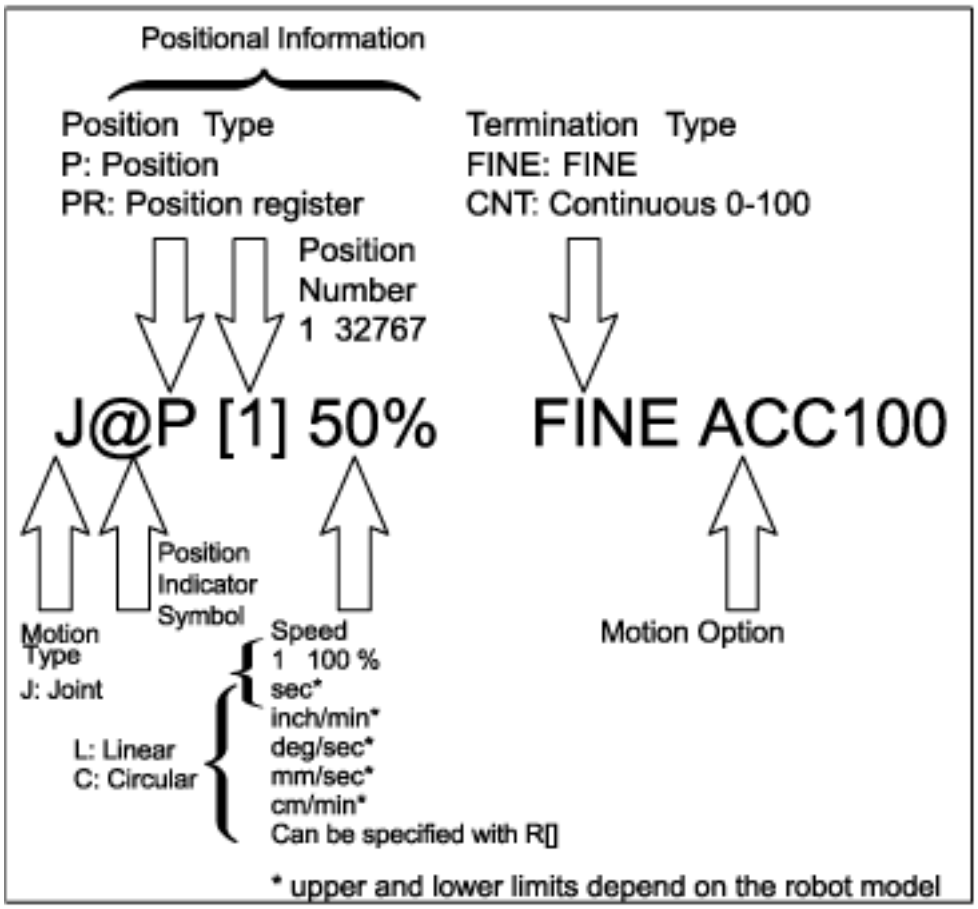
- **Motion type** - How the robot moves to the position
- **Position indicator symbol** - Indicates that the robot is at the taught position
- **Positional information** - Where the robot moves
- **Termination type** - How the robot ends the move to the position

- **Speed** - How fast the robot moves to a position
- **Motion options** - Additional commands that perform specific tasks during robot motion

A typical example motion instruction is shown in Figure 6–4 .

Note If you are using HandlingTool, for information on the correspondence between teach pendant program motion and KAREL program motion, refer to the section called “Correspondence Between Teach Pendant Program Motion and KAREL Program Motion” in the *HandlingTool Setup and Operations Manual*.

Figure 6–4. Typical Motion Instruction Example



Note Wrist Joint (W/JNT) is a motion option that allows the robot to move only in Linear or Circular.

6.4.2 Motion Type

Motion type defines how the robot will move to the destination position. There are three motion types:

- Joint
- Linear
- Circular

Joint Motion

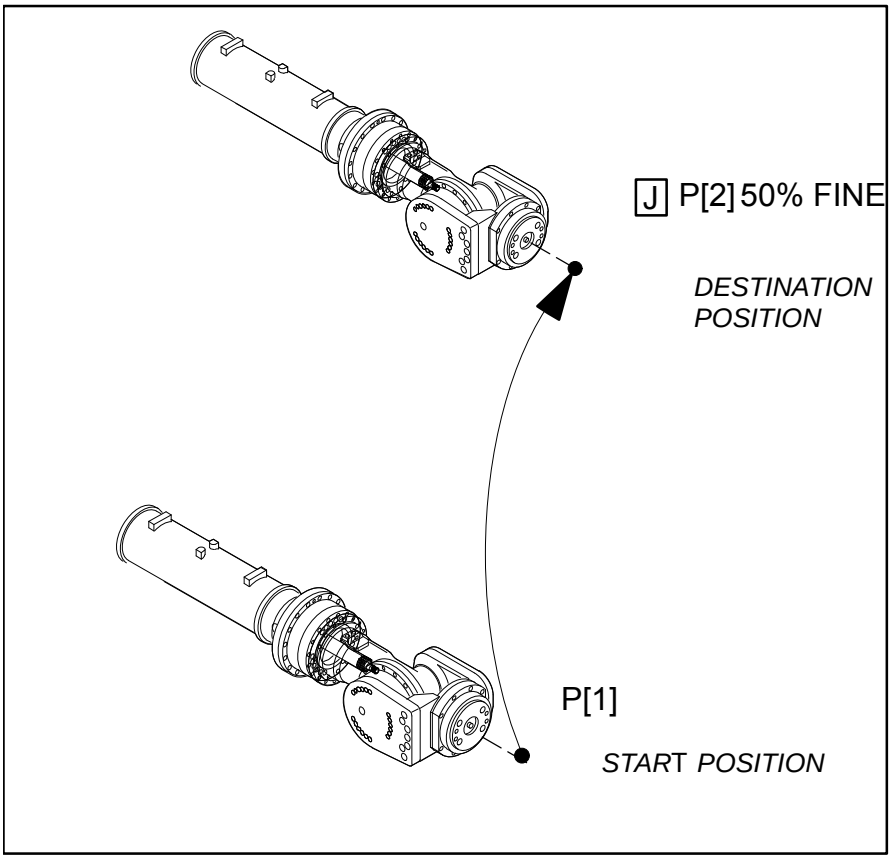
```
J P[2] 50% FINE
```

Joint motion

- Causes the robot to move all required axes to the destination position simultaneously. The motion of each axis starts and stops at the same time.
- Is programmed at the destination position.
- Speed is specified as a percentage of the total default speed, or in seconds. The actual speed of the move is dependant on the speed of the slowest axis. Refer to Section 6.4.8 .

Figure 6–5 shows an example of joint motion.

Figure 6–5. Joint Motion Type



Linear Motion

```
L P[2] 100mm/sec FINE
```

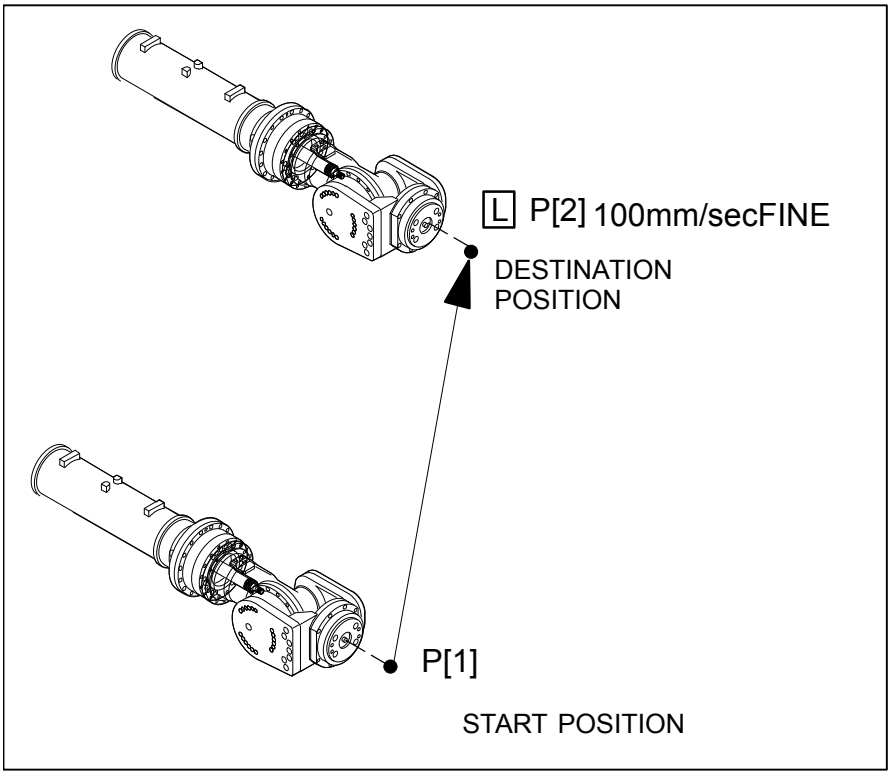
Linear motion

- Causes the robot to move the tool center point in a straight line from the start position to the destination position.
- Is programmed at the destination position.
- Speed is specified in millimeters per second, centimeters per second, inches per minute, degrees per second, or seconds. Refer to Section 6.4.8 .

During a linear move, the orientation of the tool changes gradually as the robot moves from the start position to the destination position, depending on how the destination position is programmed.

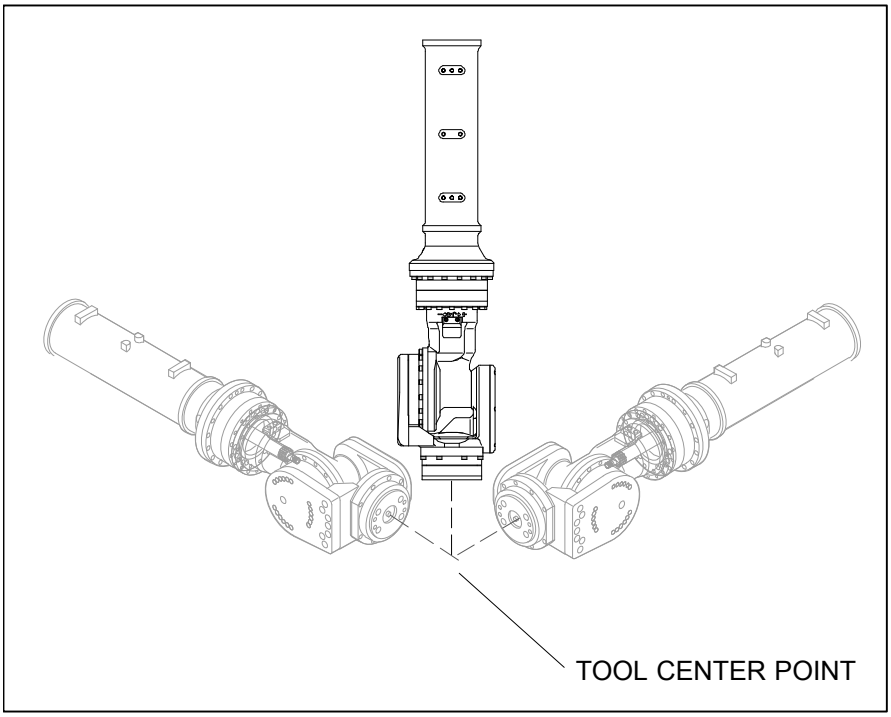
Figure 6–6 shows an example of linear motion.

Figure 6–6. Linear Motion Type



Linear motion type can also be used to rotate about the tool center point while maintaining that position. The speed for this type of motion is in degrees per second. Figure 6–7 shows an example of linear motion used for rotation about the tool center point.

Figure 6–7. Linear Motion Type Used to Rotate About the Tool Center Point



Circular Motion

```
C P[2] P[3] 100mm/sec FINE
```

Circular motion

- Causes the robot to move the tool center point in an arc from the start position through an intermediate to the destination position.
- Is programmed at the intermediate position.
- Speed is specified in inches per minute, millimeters per second, and centimeters per minute. Refer to Section 6.4.8 .

When you add a motion instruction that has circular motion type, the following appears on the screen:

```
C P[2] P[3] 100 mm/sec FINE
```

The first position, P[2] in the example, is the*intermediate* position. The intermediate position is automatically recorded as the current robot position when you add the motion instruction. The second position, P[3] in the example, is the *destination* position. You must record the destination position, after you add the circular motion instruction, using the TOUCHUP function key, F5.

If you change an existing point to "C", that position becomes the "via" or intermediate position.

To program a complete circle, add two circular motion instructions, which will generate two intermediate positions and two destination positions. The circular motion instructions can be added by:

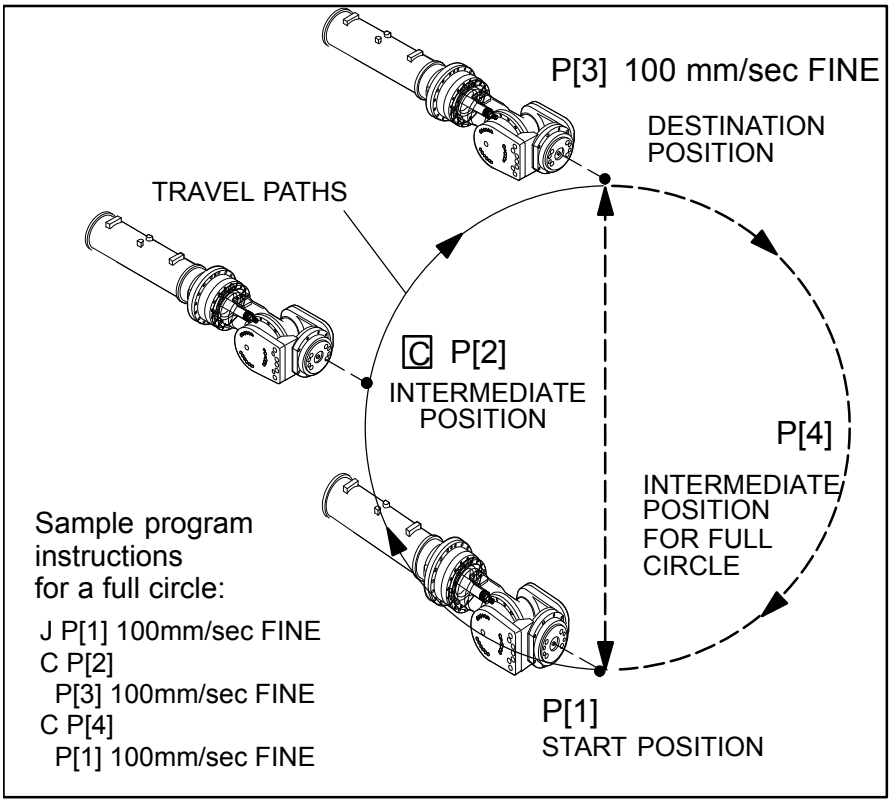
- Inserting a line.
- Returning to DEFAULT
- Selecting [INST].
- Editing a default instruction to add the circular motion instruction.
- Teaching a point with the current default and then modifying the line to become a circular motion statement.

Circular Orientation Control at Intermediate (Via) Point

Circular orientation control at the intermediate "via" point ensures that the robot will go through the "via" point at the taught orientation point. Orientation is smoothly changed between the *start, via,* and *end* points.

Figure 6–8 shows an example of circular motion.

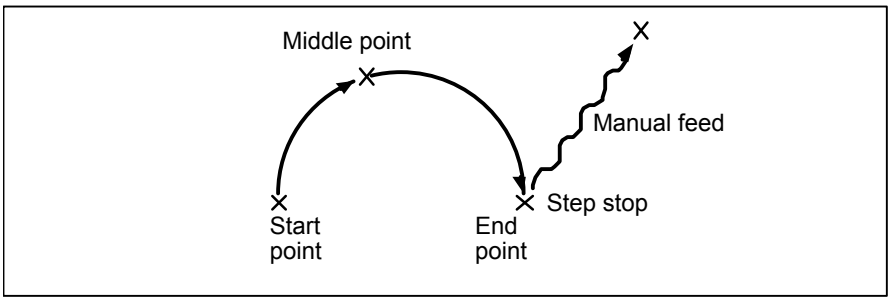
Figure 6–8. Circular Motion Type



Restart of Circular Motion

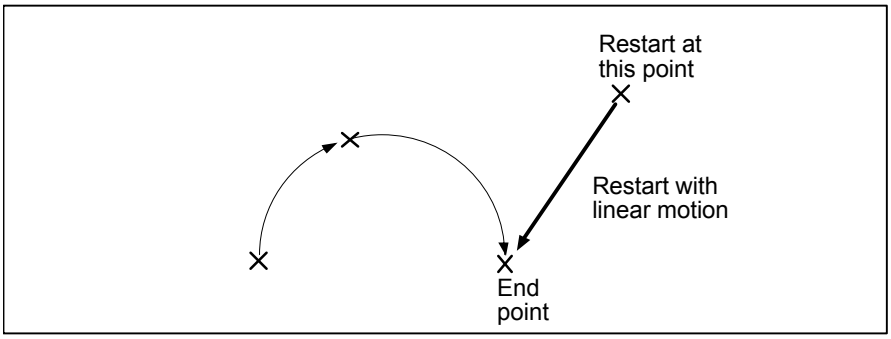
In Figure 6–9 a single-step stop occurs at the destination position of a circular motion instruction. You can then jog the robot.

Figure 6–9. Restart of Circular Motion Instruction



In Figure 6–10 when program execution is restarted after a single-step stop and jogging, the robot moves, using linear motion to the end point of the previous circular motion.

Figure 6–10. Restart of Circular Motion Instruction



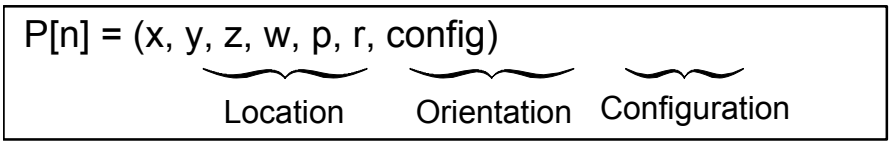
6.4.3 Positional Information

Positional information describes the **location, orientation, and configuration of the tool center point** when a motion instruction is added to a program. Positional information is *recorded* when the motion instruction is added to the program. Refer to the "Planning and Creating a Program" chapter in the Setup and Operations Manual for more information on adding motion instructions.

J P[1] 50% FINE

Positional information is made up of seven components, as shown in Figure 6–11 . These components are represented by the*position command*, P[n].

Figure 6–11. Positional Information



- **Location** components, (x,y,z), describe the three-dimensional location of the position.
- **Orientation** components, (w,p,r), describe rotation about x, rotation about y, and rotation about z.
- The**configuration** component describes the condition of the axes when the robot arrives at the destination position. Orientation of the wrist axes at the destination position remains the same, but the orientation of the other axes might change.

In the motion instruction, positional information is represented as a position command, P[n], or*position register*, PR[x]. The **n** is the *position number*. The **x** is the position register number. A position command stores positional information with the motion instruction in the program. A position register stores positional information in a storage location separate from the motion instruction. Refer to Section 6.6 .

The position number identifies the position. Position numbers are automatically assigned when a motion instruction is added to a program. The first number assigned is [1], the second [2], and so forth.

If you add a position before an already existing position, the position number is incremented from the last numbered position regardless of its place in the program. You can request that positions be renumbered so that the position numbers are sequential in your program.

When you delete positions, all other taught positions keep their current numbers unless you request that they be renumbered.

Positions can also have comments of one to 16 characters. You specify these when you add or modify positional information.

Refer to the “Planning and Creating a Program” chapter in the Setup and Operations Manual for more information on modifying the positions in your program.

6.4.4 Position Confirmation

Position Confirmation provides a visual indicator (an @ symbol) on a motion line when the robot is near the position. To configure the Position Confirmation feature, set the system variable $MNDSP_POSCF using a setting in Table 6–5 .

Table 6–5. $MNDSP_POSCF Configuration Settings

$MNDSP_POSCF	Description	Example Program
0 (Disabled)	The “@” symbol is not displayed in the editor.	1:J P[1] 100% FINE 2:J P[2] 40% FINE : Arc Start E1[1] 3:L P[3] WELD_SPEED FINE : Arc End E1[3] 4:J P[1] 100% FINE
1 (Multiple @)	The "@" symbol is displayed next to all positions if the robot is currently near those positions.	1:J @P[1] 100% FINE 2:J P[2] 40% FINE : Arc Start E1[1] 3:L P[3] WELD_SPEED FINE : Arc End E1[3] :J @P[1] 100% FINE
2 (Single @)	The "@" symbol is displayed next to the position of the current line if the robot is near that position.	1:J @P[1] 100% FINE 2:J P[2] 40% FINE Arc Start E1[1] 3: P[3] WELD_SPEED FINE : Arc End E1[3] 4:J P[1] 100% FINE

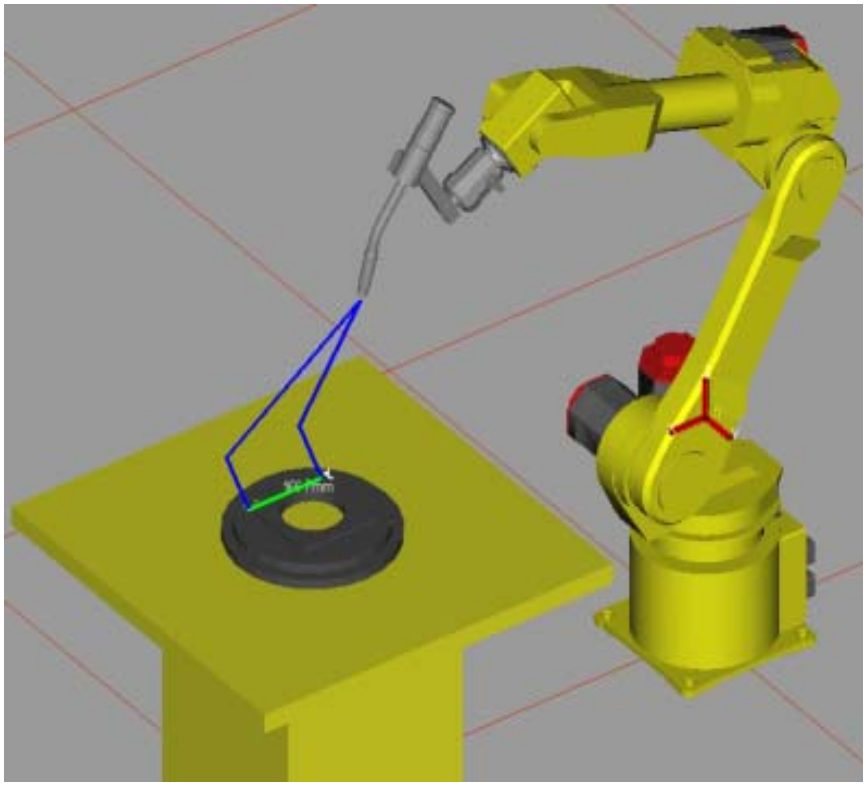
6.4.5 Motion Status Display

The teach pendant editor can display motion status information on each motion line. The example program shown in Figure 6–12 displays the distances from all the positions in the program to the current tool center point (TCP) in millimeters.

Figure 6–12. An Example Program in Display Mode 1

```
TEST1              LINE 0     T2  ABORTED
TEST1                         WORLD 100 %
        P[mm to TCP G1]                1/7
   1:J @P[1:     0.000] 100% FINE
   2:J  P[2:   373.559] 100% CNT100
   3:J  P[3:   418.852] 100% FINE
    :  Arc Start[1]
   4:L  P[4:   461.159] 20.0inch/min
    :  FINE
    :  Arc End[1]
   5:J  P[5:   420.448] 100% CNT100
   6:J @P[1:     0.000] 100% FINE
[End]

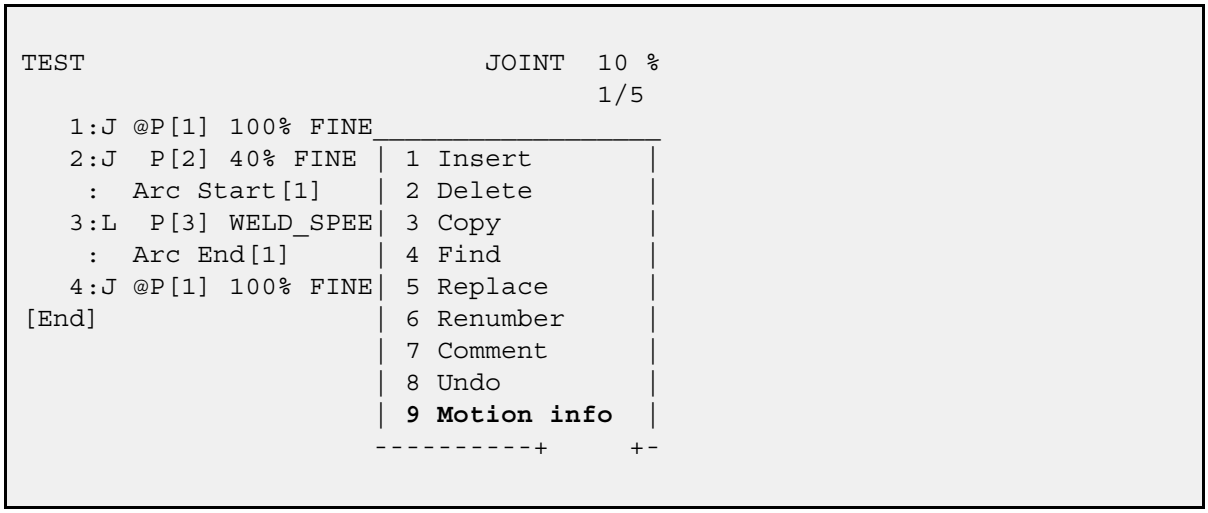
 POINT  ARCSTRT WELD_PT  ARCEND TOUCHUP>
```



The distances are updated continuously as the robot moves. This feature can be helpful while teaching or testing programs. This is display mode number 1. In this mode, the distance from the position to the TCP is displayed on multiple lines. Other display modes are available. The other modes display data only on the current motion line.

This feature is enabled and configured by setting the system variable $MNDSP_MST. After the feature is enabled and configured, it can easily be toggled ON or OFF by selecting the “Motion info” entry in the list of EDCMD commands. See Figure 6–13 .

Figure 6–13. Motion Info is Toggled ON and OFF with EDCMD



Setup

The behavior of the motion status display feature can be adjusted using the following fields in the system variable $MNDSP_MST. Refer to the *System Reference Manual* for more information about $MNDSP_MST.

$disp_enable turns the motion status display ON or OFF in the Editor. It is toggled by the motion status element in the EDCMD menu.

$disp_edcmd enables the display of the motion status menu choice in the EDCMD pullup menu.

$disp_inauto enables the motion status display during AUTO mode. $disp_enable must also be TRUE.

$disp_is_on indicates the motion status is displayed in the editor.

$disp_rsmdis enables display of the resume distance in the editor.

$mode_grp[1..7] is the display mode for each group. Valid modes are listed in Table 6–6 . See Table 6–7 for an example program display for each mode.

Table 6–6. Display Modes

Mode	Description	Header
1	Distance on ONE line	P[mm to TCP G1]
2	Distance on Multiple lines	P[mm to TCP G1]
3	COMMAND PERCENT	P[Cmnd % G1]
4	COMMAND TIME	P[Cmnd Time G1]
5	COMMAND DISTANCE	P[Cmnd Dist G1]
6	COMMAND DISTANCE UP	P[Cmnd D UP G1]
7	COMMAND DISTANCE DOWN	P[Cmnd D DN G1]
8	COMMAND PROGRESS	P[Cmnd PBar G1]
9	COMMAND PERCENT AND DISTANCE UP	P[Cmnd % UP G1]
10	COMMAND PERCENT AND DISTANCE DOWN	P[Cmnd % DN G1]
11	COMMAND PERCENT AND TIME	P[Cmnd % T G1]
19	Travel Angle in degrees	P[Travel dg G1]
20	Work Angle in degrees	P[WorkXZ dg G1]
21	Current X location in world frame	P[X (1) mm G1]
22	Current Y location in world frame	P[Y (2) mm G1]
23	Current Z location in world frame	P[Z (3) mm G1]
24	Current W orientation in world frame	P[W (4) deg G1]
25	Current P orientation in world frame	P[P (5) deg G1]
26	Current R orientation in world frame	P[R (6) deg G1]

Comments

Positions can have comments up to 16 characters in length. When the motion status is displayed, only the first 10 characters of the comments will be visible.

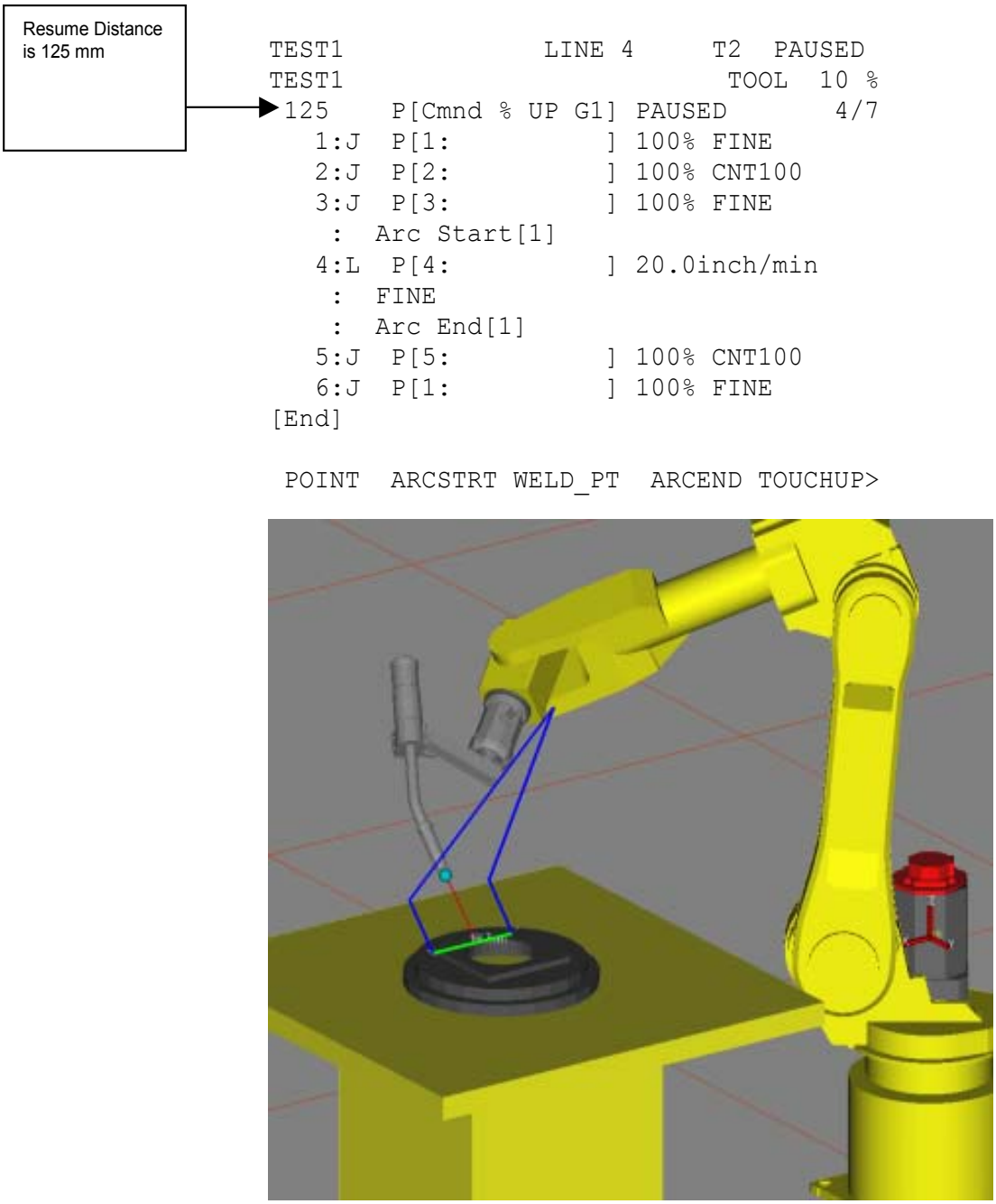
Multiple Groups

When a system has multiple groups, the motion status display can display information for any of the groups. The group number is displayed in the header with a short description of the display mode. To change the selected group number, use the FCTN menu entry CHANGE GROUP. If the selected group is not in the program group mask, the lowest group number in the mask is selected for display. Each group can display data in a different mode. You can set the mode for each group using the array $mndsp_mst.$mode_grp[1..6].

Resume Distance

When a running program pauses the robot stops and the position of the robot is recorded. If the robot is moved from the recorded stop position it may return to this position when the program is resumed, depending on what options are installed. The distance the robot has moved from the stop position can be displayed in the upper left corner of the editor if the variable $mndsp_mst.$disp_rsmdis = TRUE.

Figure 6–14. Resume Distance Display



Distance Modes

Modes 1 and 2 display the distance between the program positions and the robot tool center point. The distance is along a straight line, not along the programmed path. The actual distance traveled by the robot could be greater if it involves Joint or Circular motion.

Motion Command Modes

The motion command modes 3 – 11 display the output from the motion system as it executes the current line. The timing of this data slightly precedes the actual robot position. The times and distances displayed reflect the motion command, not the real time feedback data from the motor encoder. When a motion is paused the data will remain displayed. When a motion is resumed the data displayed will reflect the motion from the stop position to the destination position.

Note Modes 3 - 11 only display distance information for L, C, and A motion types. In modes 3 - 11, Joint motions will display an increasing percentage and total time like Mode 11.

Table 6–7. Display Mode Examples

Mode	Example	Description
0	P[Inv. Mode G1] \\\1:J @P[1:] 100% FINE 2: P[2:] 40% FINE : Arc Start[1] 3:L P[3:] WELD_SPEED FINE : Arc End[1] 4:J @P[1:] 100% FINE	No motion status displayed. This is an invalid mode for this group. In this example, group 1 is selected for display. The display mode for group 1 is specified in $MNDSP_MST.$MODE_GRP[1]. If the mode is 0, or some other mode not listed in this table, the header will indicate "Invalid Mode," and data will not be displayed in the motion lines.
1	P[mm to TCP G1] \\\1:J @P[1: .000] 100% FINE 2:J P[2:] 40% FINE : Arc Start[1] 3:L P[3:] WELD_SPEED FINE : Arc End[1] 4:J @P[1:] 100% FINE	The distance of this position from the TCP is displayed, for this line only. In this example, $MNDSP_MST.$MODE_GRP[1] = 1. The cursor is on line 1 and the robot is at P[1]. The distance from P[1] to the robot Tool Center Point (TCP) in millimeters is displayed as 0.000. If you jog the robot away from P[1], the distance to the tool will be displayed dynamically.

Table 6–7. Display Mode Examples (Cont'd)

Mode	Example	Description
2	P[mm to TCP G1] \\\1:J @P[1: .000] 100% FINE 2:J P[2: 26.795] 40% FINE : Arc Start[1] 3:L P[3: 11.520] WELD_SPEED FINE : Arc End[1] 4:J @P[1: .000] 100% FINE	Distance on MULTIPLE lines. Mode 2 is similar to mode 1. In mode 2 a distance is displayed in all the motion lines, not just the cursor line. The distance is from the recorded position to the current tool center point.
3	P[Cmnd % G1] \\\3:L P[3: 28%] WELD_SPEED FINE : Arc End[1]	COMMAND PERCENT Mode 3 displays the percentage of the motion completed while the move progresses. It is updated dynamically. If you pause execution before reaching the position and continue, the percentage will be reset to 0 and then increase to reflect the remaining motion.
4	P[Cmnd Time G1] \\\3:L P[3: 2.9s] WELD_SPEED FINE : Arc End[1]	COMMAND TIME Mode 4 displays the total time required for the motion to complete. If does not change dynamically, unless the speed override is changed. If the speed override is changed, the time is updated to reflect the total time for the entire motion at the new override, not the remaining time.
5	P[Cmnd Dist G1] \\\3:L P[3: 24.19] WELD_SPEED FINE : Arc End[1]	COMMAND DISTANCE Mode 5 displays the total move distance. It does not change during the motion. If you pause execution and then resume the display will change to show the distance to P[3] from the pause position.

Table 6–7. Display Mode Examples (Cont'd)

Mode	Example	Description
6	P[Cmnd D UP G1] \\\3:L P[3: 6.758] WELD_SPEED FINE : Arc End[1]	COMMAND DISTANCE UP Mode 6 is similar to mode 3. It is dynamic. It shows the distance increasing (UP) while moving to P[3]. If you pause and resume, the distance is reset to zero. If single step is active, stepping through the entire motion on line 3 to P[3], without stopping, will result in the total move distance displayed when the robot reaches P[3].
7	P[Cmnd D DN G1] \\\3:L P[3: 17.43] WELD_SPEED FINE : Arc End[1]	COMMAND DISTANCE DOWN Mode 7 is just like mode 6 except the distance is shown decreasing (DN - down) to zero.
8	P[Cmnd PBar G1] \\\3:L P[3:--] WELD_SPEED FINE : Arc End[1]	COMMAND PROGRESS Mode 8 displays a progress bar. It behaves just like the percentage of Mode 3, only graphically.
9	P[Cmnd % UP G1] \\\3:L P[3: 28% 6.758] WELD_SPEED FINE : Arc End[1]	COMMAND PERCENT and DISTANCE UP Mode 9 displays both the increasing percentage of Mode 3 and the increasing distance of Mode 6.
10	P[Cmnd % DN G1] \\\3:L P[3: 28% 17.43] WELD_SPEED FINE : Arc End[1]	COMMAND PERCENT and DISTANCE DOWN Mode 10 displays both the increasing percentage of Mode 3 and the decreasing distance of Mode 7.

Table 6–7. Display Mode Examples (Cont'd)

Mode	Example	Description
11	P[Cmnd % T G1] \\\3:L P[3: 28% 2.9s] WELD_SPEED FINE : Arc End[1]	COMMAND PERCENT and TIME Mode 11 displays both the increasing percentage of Mode 3 and the total time of Mode 4.
19	P[Travel dg G1] \\\1:J @P[1: —10.277] 100% FINE	Travel Angle in degrees A negative number indicates a push angle. A positive number indicates a drag angle. Refer to Section 6.4.10.19 .
20	P[WorkXZ dg G1] \\\1:J @P[1: 36.832 R] 100% FINE	Work Angle in degrees The header WorkXZ indicates the work angle is relative to the XZ plane. WorkXY means relative to the XY plane. The letter L or R after the angle indicates the torch is to the left or right of vertical. Refer to Section 6.4.10.19 .
21	P[X (1) mm G1] \\\1:J @P[1: 2070.261] 100% FINE	Current X location in world frame
22	P[Y (2) mm G1] \\\1:J @P[1: 25.242] 100% FINE	Current Y location in world frame
23	P[Z (3) mm G1] \\\1:J @P[1: 1226.707] 100% FINE	Current Z location in world frame

Table 6–7. Display Mode Examples (Cont'd)

Mode	Example	Description
24	P[W (4) deg G1] \\\1:J @P[1: -2.027] 100% FINE	Current W orientation in world frame
25	P[P (5) deg G1] \\\1:J @P[1: 19.972] 100% FINE	Current P orientation in world frame
26	P[R (6) deg G1] \\\1:J @P[1: 2.375] 100% FINE	Current R orientation in world frame

Limitations

- Joint motions do not display distances for the Command Distance modes. Percentage and time are displayed instead.
- Command times and distances are approximate.
- Incremental motions do not display distance properly for modes 1 and 2.
- Not all motion options and formats are supported.
- The units for distance are always mm and cannot be changed.
- Extended axes are not supported.

6.4.6 Frame Number of Positional Data

The User Frame (UF) and User Tool frame number (UT) are displayed at the top of the Position Detail screen. See the following screen for an example.

```
P[1]  UF:0 UT:1  CONF: N 00
X    100.000    mm  W     12.555    deg
Y    100.000    mm  P      3.123    deg
Z    100.000    mm  R       .014    deg
```

These fields indicate the current frame number.

UF: User Frame number

- 0 = world coordinate
- 1-10 = normal UFRAME number
- F = current $MNUFRAMENUM

UT: User Tool frame number

- 0 = not valid
- 1-10 = normal UTOOL number
- F = current $MNUTOOLNUM

Note These values cannot be modified directly from the teach pendant.

Note The position register screen has UF and UT in the same area, and this value is always "F" for both.

6.4.7 Switch Frame Check Type

This function provides an easy way to move forward or backward past a frame change in a teach pendant program. You must set $FRM_CHKTYP to move through the frame change correctly.

The system variable $FRM_CHKTYP allows you to switch the frame check type. Refer to Table 6–8 .

Table 6–8. $FRM_CHKTYP Values

$FRM_CHKTYP	DESCRIPTION
-1	Prohibits FWD or BWD motion between two points which have different frame numbers.
-2	The system does not check frame number at FWD and BWD. System does not change the frame number $MNUFRAME_NUM, $MNUTOOL_NUM when you FWD/BWD execute past a frame change.
2	System does not check frame number at FWD and BWD. System changes the frame number ($MNUFRAME_NUM, $MNUTOOL_NUM) when you FWD/BWD execute past a frame change.

Note If a different frame is used for circular motion, the robot will not move, even if $FRM_CHKTYP is 2 or -2.

Figure 6–15 shows the frame numbers used for the positional data in a program example.

Figure 6–15. Frame Number of Positional Data Example Program

```
1: UTOOL_NUM = 1 2: J P[1] 100% FINE

/* P[1] UT=1 */
3: J P[2] 100% FINE

/* P[2] UT=2 */
```

Table 6–9 describes how the example program in Figure 6–15 operates, depending on the setting of the $FRM_CHKTYP system variable.

Table 6–9. Example Program Operation

$FRM_CHKTYP Value	FWD Execution	BWD Execution
-1	The system posts a book keeping error at line 3.	You manually set UTOOL_NUM = 2 and backward execute from line 3. The system posts book keeping error at line 2.
-2	The system does not post a book keeping error and the robot moves correctly.	You bwd execute from line 3 to line 2. The system does not post a book keeping error.
2	The system does not post a book keeping error at line 3 and the robot moves correctly. Also, the system automatically changes UTOOL_NUM to 2 at line 3.	You backward execute from line 3 to line 2. The system does not post a book keeping error at line 2. But the system automatically changes UTOOL_NUM to 1 when backward executing from line 3 to line 2.

6.4.8 Speed

Speed defines how fast the robot moves to a position.

The motion type used determines the units of speed. Depending on the motion type you want, you can specify speed in millimeters per second, centimeters per minute, inches per minute, rotational degrees per second, or seconds.

When a program is running, you can change the speed override using the +% and -% keys on the teach pendant. The value ranges from .01% (very fine) to 100 percent of the *programmed speed*. Programmed speed is the speed specified in the program.

Note The programmed speed cannot exceed the capability of the robot. If programmed speed cannot be met, an error will occur.

Note If you are using HandlingTool, for information on the correspondence between teach pendant program motion and KAREL program motion, refer to the section called “Correspondence Between Teach Pendant Program Motion and KAREL Program Motion” in the *HandlingTool Setup and Operations Manual*.

Joint motion uses

- A percentage (%) of the total default speed. Joint motion speed can have a value of 1% to 100% of the maximum joint speed limit.

```
J P[1]
50%

FINE
```

- Seconds (sec), the length of time the motion lasts. Seconds can have a value of .1 to 3200. This is used for motion that requires an exact time span. If a program is paused and then resumed during execution of motion that uses seconds, the controller will be held in a busy and running state until the portion of time that had been executed elapses again. Then, the robot will complete the motion using the remaining amount of time. See Figure 6–16 .

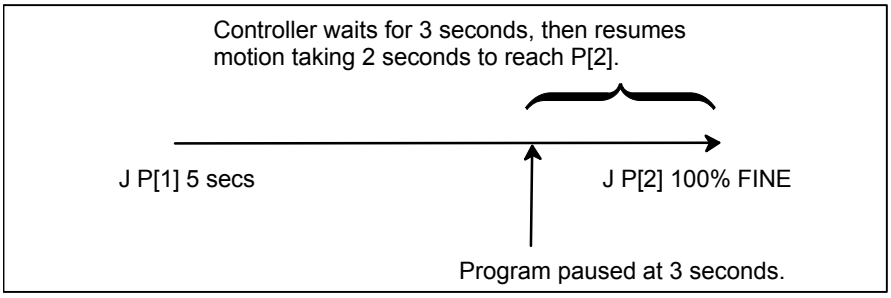
```
J P[1]
 2 sec

FINE


L P[2]
2 sec

FINE
```

Figure 6–16. Example of the Sec Speed Feature



Linear and circular motions use

- Millimeters per second (mm/sec), with a range of values from 1 to 2000 millimeters per second.
- Centimeters per minute (cm/min), with a range of values from 1 to 12000 centimeters per minute.
- Inches per minute (inch/min), with a range of values from 0.1 to 4724.41 inches per minute.
- Seconds (sec), the length of time the motion lasts. This is used for motion that requires an exact time span. If a program is paused and then resumed during execution of motion that uses seconds, the controller will be held in a busy and running state until the portion of time that had been executed elapses again. The robot will then complete the motion using the remaining amount of time. See Figure 6–16 .

```
L P[1]
100mm/sec

FINE
```

or

```
C P[1]
100mm/sec

FINE
```

Warning

If you change the motion type of a positional instruction from linear to joint, the speed value can change from mm/sec to a default value as high as 100%. Be sure to check the speed value before you execute the instruction; otherwise, you could injure personnel or damage equipment.

Note When the speed is specified in mm/sec, cm/min, or inch/min, then the value entered will represent the speed at the tool center point.

Rotational control of axes around the tool centerpoint uses rotational degrees per second (deg/sec), with a default range of values from 1 to 500 degrees per second.

```
L P[1]
90 deg/sec
```

```
FINE
```

Variable Motion Speed

You can specify motion speed by using a register in a motion instruction. The value of the specified register defines motion speed. This is called *variable motion speed*.

Warning

Before you run a program, make sure you are aware of any register values used to set speed in a motion instruction. Otherwise, unexpected motion could occur that could injure personnel or damage equipment.

Note A program will stop pre-execution of subsequent instructions when it reaches a motion instruction with the register speed type. This ensures the motion instruction uses the register speed type value. See Figure 6–17 .

Figure 6–17. Variable Motion Speed Program Execution Example

```
9:  J P[2] 100% FINE
10: R[1] = 100
11: J P[1] R[1]% FINE      (program stops pre-execution and takes the value of
12: R[1] = 10                the register in line 10 as the speed.)
13: J P[2] R[1]% FINE      (program stops pre-execution and takes the value of
                            the register in line 12 as the speed.)
```

This feature is enabled when the system variable $RGSPD_PREXE = FALSE. You can disable this feature by setting $RGSPD_PREXE = TRUE. However, the robot will not be able to move at the speed specified by the register value.

The following examples show various motion type instructions that take their speed value from a register (R[ ]).

- Joint motion type

```
J P[2] R[1]% CNT100
```

- Linear motion type

```
L P[1] R[2]mm/sec FINE
```

- Circular motion type

```
C P[2] P[3] R[3]cm/min FINE
```

- Palletizing

```
PALLETIZING-B_1
L PAL_1[A_1] R[4]mm/sec CNT100
L PAL_1[BTM] R[4]mm/sec FINE
L PAL_1[R_1] R[4]mm/sec CNT100
```

The following features are changed to accommodate variable motion speed:

- **Default motion instructions** include an instruction that uses variable motion speed.
- The **Motion Modify** screen, displayed when you select REPLACE for a motion instruction on the [EDCMD] menu, includes items for specifying variable motion speed.
- Specific **motion speed values** are valid for variable motion speed. Refer to Table 6–10 for valid variable motion speed values. If the specified register value is not a valid speed value (exceeds the speed limit or is out-of-range), an error will occur during the execution of the motion instruction.

Table 6–10. Range of Register Values to Specify a Variable Motion Speed

Unit	Range of Register Values	
%	1 to 100	Integer type
sec	0.1 to 3200.0	Float type (*1)
mm/sec	1 to 2000	Integer type (*2)
cm/min	1 to 12000	Integer type (*2)
inch/min	0.1 to 4724.41	Float type (*3)
deg/sec	1 to 500	Integer type (*4)

*1 : Valid one decimal point.

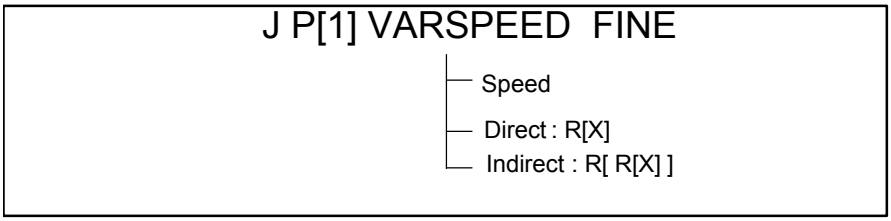
*2 : The speed limit is the value of $MRR_GRP.$SPEEDLIM.

*3 : Valid one decimal point. The limit is the value of $MRR_GRP.$SPEEDLIM/25.4 * 60.

*4 : The limit is the value of $MRR_GRP.$ROTSPEEDLIM * 180/3.141.

See Figure 6–18 for the syntax for changing the motion speed from a specific motion speed value to a variable (register) speed value or for changing the motion speed from a variable (register) speed value to a specific motion speed value.

Figure 6–18. Syntax for Changing the Motion Speed



Use Procedure 6-1 to replace speed values using the Motion Modify screen in [EDCMD] REPLACE.

Procedure 6-1 Replacing Speed Values (using Motion Modify in [EDCMD] REPLACE)

Conditions

- You are currently editing a teach pendant program that contains motion instructions.

Steps

1. Move the cursor to the line number of the instruction in which you want to replace speed values and press F5, [EDCMD].
2. Select Replace.

```
Select Replace menu 1
 Register          5 Motion modify
2 Call             6
3 I/O              7
4 JMP/LBL          8
```

3. Select Motion modify. See the following screen for an example.

```
Modify motion menu
1 Replace speed     5
2 Replace term      6
3 Insert option     7
4 Remove option     8
```

4. Select Replace speed. See the following screen for an example.

```
Select interpolate
1 Unspecified type  5
2 J                 6
3 L                 7
4 C                 8
RSR0001                            10%
   1: L P[1] 20.0sec FINE
2: L P[2] 500mm sec FINE
3: L P[3] R[1]mm sec FINE
Select source type
```

5. Specify the **interpolation type (motion type)** of the motion instruction for which you want to search:

- Unspecified type - searches for joint, linear, and circular motion instructions
- Joint - searches for joint motion instructions only
- Linear - searches for linear motion instructions only
- Circular - searches for circular motion instructions only

6. Specify the **speed type** of the motion instruction you want to search for:

- All type - searches for motion instructions that use a speed value, a variable (register) value, or an indirect variable (register) speed value.
- Speed value - searches only for motion instructions that use a speed value.
- R[ ] - searches only for motion instructions that use a variable (register) speed value.
- R[R[ ]] - searches only for motion instructions that use an indirect variable (register) speed value.

See the following screen for an example.

```
Speed type menu
1 All type          5
2 Speed value       6
3 R[ ]              7
4 R[R[ ]]           8
```

7. Specify the **units** of the replacement motion instruction.

See the following screen for an example.

```
Select motion item
1 %                 5 deg/sec
2 mm/sec            6 sec
3 cm/min            7
4 inch/min          8
```

8. Select the **speed type** of the replacement motion instruction:

- Speed value - changes the speed of the found (searched) motion instruction to a speed value.
- R[ ] - changes the speed of the found (searched) motion instruction to a variable (register) speed value.
- R[R[ ]] - changes the speed of the found (searched) motion instruction to an indirect variable (register) speed value.

See the following screen for an example.

```
Select motion item
1 Speed value       5
2 R[ ]              6
3 R[R[  ]]          7
4                   8
```

9. **If you selected R[ ] or R[R[ ]],** type a register number.

10. Select how you want the found motion instruction to be replaced:

- F2, ALL - changes all found motion instructions below the current line to the specified speed type and value.
- F3, YES - changes only the found motion instruction on the current line to the specified speed type and value.
- F4, NEXT - skips the found motion instruction on the current line and searches the next motion instruction.
- F5, EXIT - ends the motion modify operation.

See the following screen for an example.

```
RSR0001
Modify OK ?
```

11. Continue the search and replace operations as desired.

12. **When you are finished with all search and replace operations,** press F5, EXIT.

LP[1] WELD_SPEED CNT100

Motion instructions used **during welding** use the WELD_SPEED parameter. WELD_SPEED is defined in the weld schedule specified by an ArcStart instruction.

You can use WELD_SPEED only for linear or circular motion. If you change the motion type of an instruction that uses WELD_SPEED from circular or linear to joint, the speed will change to 100%.

When a motion instruction that contains WELD_SPEED is executed, the speed used depends on certain conditions:

- If the Arc START instruction is executed before executing the WELD_SPEED motion instruction, the weld speed defined in the corresponding weld schedule is used.
- If the Arc Start instruction is **not** executed before executing the WELD_SPEED motion instruction, the default weld speed is used as the value of WELD_SPEED. The default weld speed is defined on the SETUP Weld System screen.
- If the program is resumed from a WELD_SPEED motion instruction, the WELD_SPEED in effect when the program was paused is used.
- If the following sequence is executed while the program is paused and then the program is restarted, the default weld speed is used:
    3. You step the program backward through some instructions.
    4. You move the cursor to another line in the program.
    5. You abort the program.

6.4.9 Termination Type

Termination type defines how the robot ends the move in the motion instruction. The following termination types are available:

- Fine
- Continuous
- Corner Region — available only if you have the constant path motion option and corner region option

The fine and continuous termination types are described in this section. Refer to Section 6.4.10.18 for information on the corner region termination type.

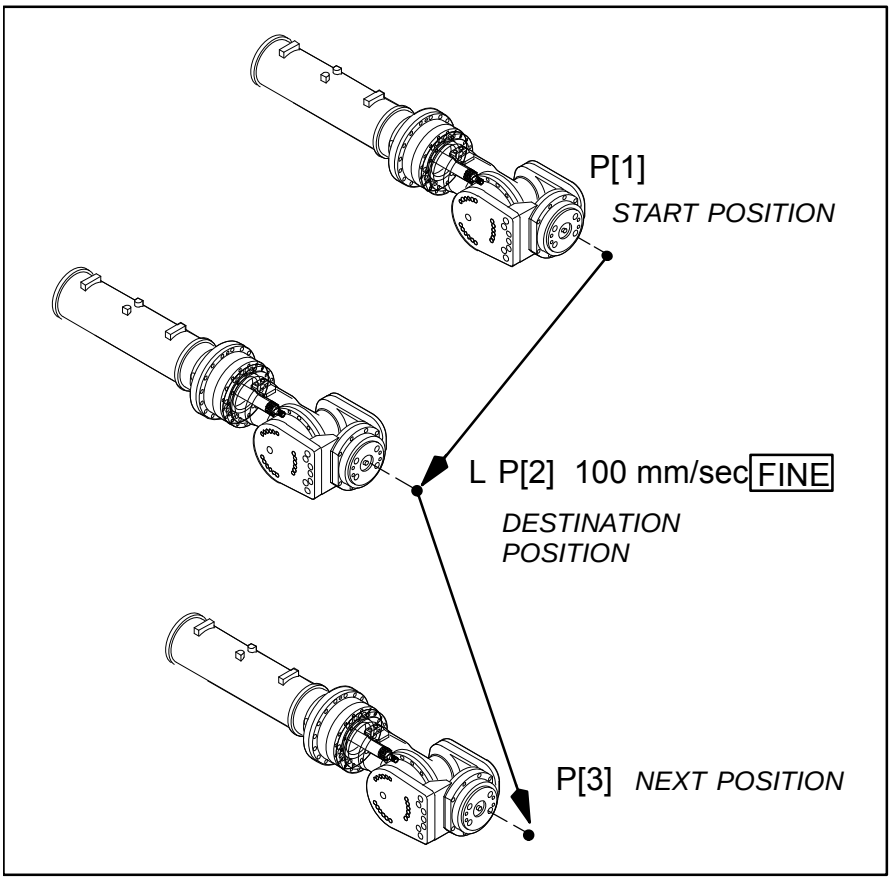
Fine Termination Type

```
J P[1] 50%
FINE
```

Fine termination type causes the robot to **stop at the destination position** before moving to the next position.

Figure 6–19 shows how the robot will move when you specify the fine termination type.

Figure 6–19. Robot Motion with Fine Termination Type



Continuous Termination Type

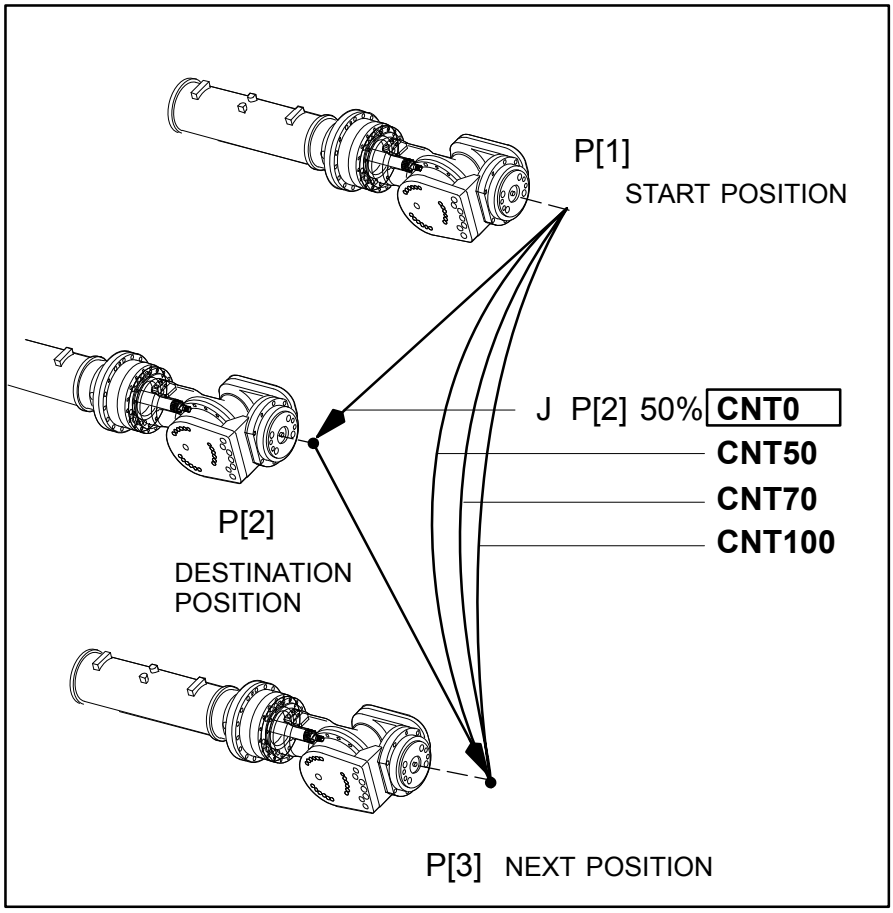
```
J P[1] 50%
CNT50
```

Continuous termination type allows the robot to **decelerate as it approaches the destination position but does not stop** at it before it accelerates toward the next position. A value from 0 to 100 defines how close the robot comes to the destination position. At CNT0 the robot is closest, with maximum deceleration. At CNT100 the robot is farthest, with minimum deceleration.

Note Programming certain instructions, such as WAIT, causes the robot to stop at the destination position and execute the instruction before it executes the next instruction.

Figure 6–20 shows how the robot will move with different continuous termination type values.

Figure 6–20. Robot Motion with Continuous Termination Type



6.4.10 Motion Options

Motion options can be used to provide additional information to perform specific tasks during robot motion. Motion options include

- Wrist joint

  Refer to Section 6.4.10.1
- Coordinated motion

  Refer to Section 6.4.10.2
- Acceleration override

  Refer to Section 6.4.10.3
- Minimal rotation with joint motion

  Refer to Section 6.4.10.4
- Skip label

  Refer to Section 6.4.10.5
- Offset

  Refer to Section 6.4.10.6
- Offset position register

  Refer to Section 6.4.10.7
- Tool offset

  Refer to Section 6.4.10.8
- Tool offset position register

  Refer to Section 6.4.10.9
- Incremental motion

  Refer to Section 6.4.10.10
- Search [ ]

  Refer to Section 6.4.10.11
- EV (extended velocity)

  Refer to Section 6.4.10.12
  - — Simultaneous EV
  - — Independent EV
- PTH

  Refer to Section 6.4.10.13

- Time before/Time after

  Refer to Section 6.4.10.14
- Remote TCP

  Refer to Section 6.4.10.15
- Break

  Refer to Section 6.4.10.16
- Constant Path

  Refer to Section 6.4.10.17
- Advanced Constant Path

  Refer to Section 6.4.10.18
  - — Linear Distance
  - — Corner Region
  - — Process Speed
  - — Max Speed

6.4.10.1 Wrist Joint

```
L P[1] 50% FINE
W/JNT
```

The wrist joint option is used during linear or circular moves. It causes the wrist orientation to change during moves, permitting the tool center point to move along the programmed path without flipping the wrist axes due to axis singularity positions.

6.4.10.2 Coordinated Motion

```
J P[1] 50% FINE
 COORD
```

The coordinated motion option describes motion for multiple motion groups. When this option is used, multiple motion groups move together to maintain the same position relative to each other.

Motion speed which is specified in the line is relative speed for coordinated motion. This option is effective on linear and circular motion.

6.4.10.3 Acceleration Override

```
J P[1] 50% FINE
 ACC50
```

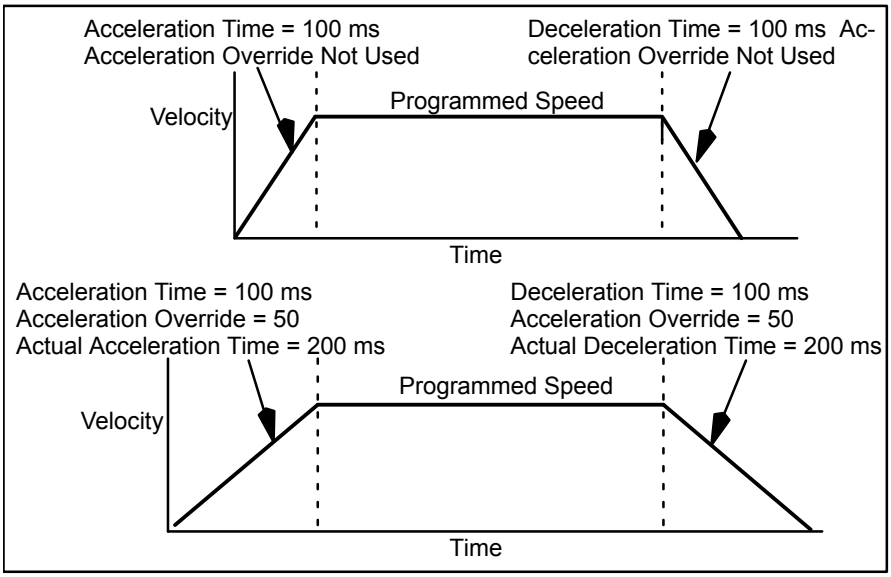
The acceleration override motion option specifies the acceleration/deceleration override value for each axis during motion. Acceleration override shortens or lengthens the acceleration time when the robot moves from a starting position to the destination position. Acceleration override is programmed at the destination position.

The acceleration override value ranges from 20 to 150%. This value is a percentage of the acceleration. For example, an acceleration override of 50 means the robot will take twice as long to accelerate or decelerate. Figure 6–21 shows how the acceleration override is used.

The acceleration override was created to allow the user to make specific moves slower or more conservative for cases when extra care is needed. The usage of acceleration override over 100% could allow more aggressive motion, but may also cause jerky motion and, if the Collision Guard option is loaded, false collision alarms could occur.

In general, the usage of acceleration override over 100% should be limited. This setting may reduce the life of the mechanical unit because the default tuned accelerations are being overridden by more aggressive values.

Figure 6–21. Acceleration Override



6.4.10.4 Minimal Rotation (HandlingTool Only)

```
J P[1] 50% FINE
MROT
```

Minimal Rotation (MROT) is a motion option to be used with the Joint motion type, or Cartesian motion with the WJNT option. It generates the shortest joint angle move for the wrist axes within axes limits during Joint and WJNT motion, and ignores the turn numbers of the wrist. You should add MROT to Joint and WJNT motion instructions for motion that requires the shortest joint angle movement.

This option is useful when you do not know the correct turn number for a destination position that is calculated in Cartesian space, and which requires the shortest joint angle move.

When you are using the MROT option for Joint motion, or for linear motion with WJNT, if an axis limit error occurs on one of the wrist axes only, the system will

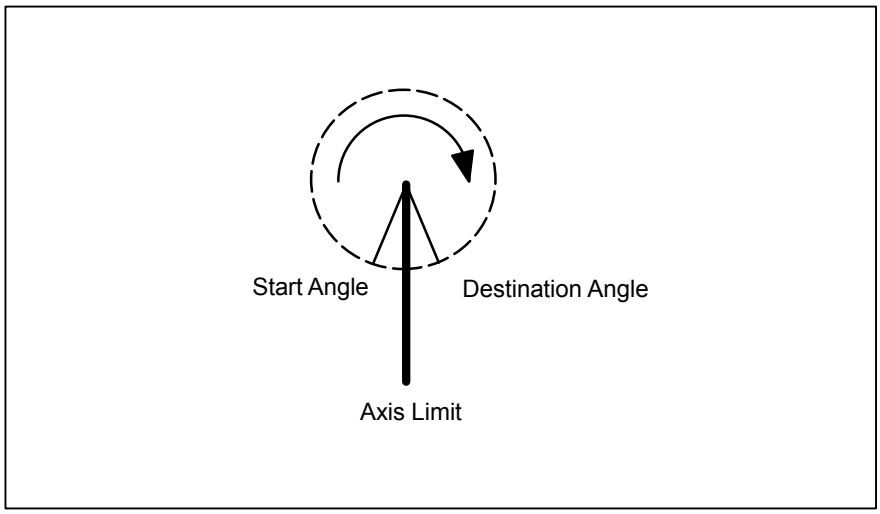
- Post a warning message similar to MOTN-330 MROT Limit Warn (G:1, A:20 Hex).
- Attempt to move in the opposite direction for this axis, which generates the shortest motion **within** the axis limit.

For example, the upper and lower limits of a wrist axis are 270 degrees and –270 degrees, and the start and destination angles are 260 degrees and –80 degrees, respectively. In this case, the shortest move should be the motion from 260 degrees to 280 degrees.

Note Physically, 280 degrees and –80 degrees are the same position for the axis.

However, a limit error occurs for this motion because it strokes the upper limit. In this case, a warning message will be posted and the axis will move from 260 degrees to –80 degrees with a 340 degree change in joint angle. See Figure 6–22

Figure 6–22. Shortest Motion Within Axis Limit



Note For M–410i series robots (4-axis robots), users can choose one of the two different motion behaviors by setting the system variable $group[].$ornt_mrot as follows:
(1) if $group[].$ornt_mrot = TRUE, the robot tries to move so that the change in the part orientation in Cartesian space is minimized within the axis limit;
(2) if $group[].$ornt_mrot = FALSE, robot tries to move so that the individual joint angle change is minimized within the axis limit.

The MROT motion option has the following limitation:

- Before MROT can take effect, the destination position of the Joint or WJNT motion must be represented in Cartesian space (XYZWPR form). Otherwise, if the destination position is represented in Joint angles, the system will attempt to reach the specified destination joint angles regardless of the MROT option.
- For a motion line with the MROT instruction, if the taught point is represented in Cartesian space (XYZWPR from), the actual joint angles may be different from the taught joint angles. As a result, Single Step Backward motion may not be the same as forward motion. For example, in a TP program similar to the following:

```
J P[1] 100% FINE
J P[2] 100% FINE MROT
J P[3] 100% FINE
```

if the actual reached joint angles at P[2] are different from the taught joint angles (in terms of turn number), then backward motion from P[3] to P[2] would be different from the forward motion from P[2] to P[3].



Caution

Be extremely cautious when executing motion backward to the destination position of a motion line with MROT option.

6.4.10.5 Skip Label

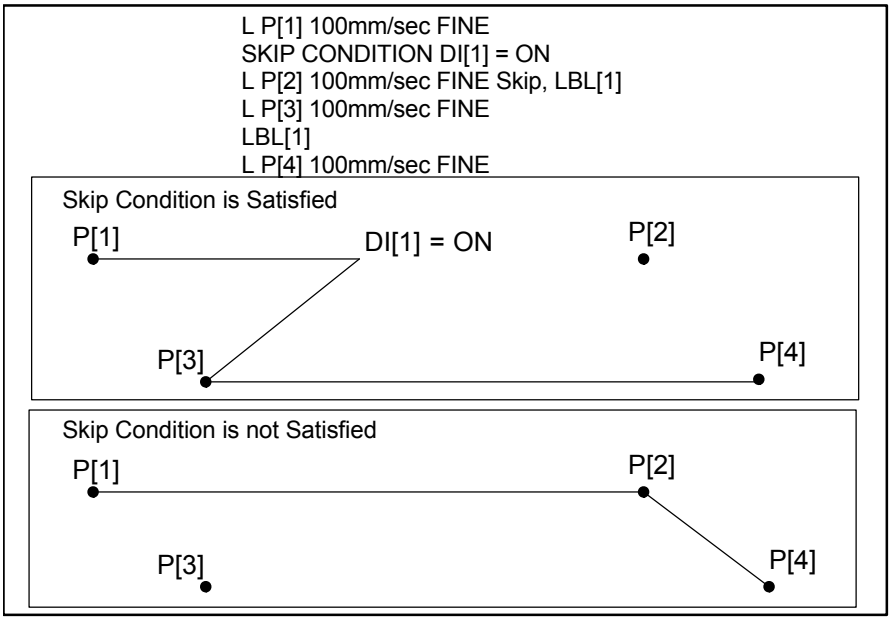
```
SKIP CONDITION [I/O] = [value]
J P[1] 50% FINE
Skip, LBL[3]
```

The Skip, LBL[x] motion option redirects program execution based on whether a predefined*SKIP CONDITION* is true. A SKIP CONDITION instruction defines an I/O condition. The execution of the motion instruction that contains the Skip, LBL[x] motion option is affected depending on status of the SKIP CONDITION, as follows:

- **If the SKIP CONDITION is satisfied,** the motion defined in the motion instruction that contains the Skip, LBL[x] motion option terminates and the next program instruction is executed.
- **If the SKIP CONDITION is not satisfied,** the motion defined in the motion instruction that contains the Skip, LBL[x] motion option is executed. After the robot reaches the destination position and the condition is still not satisfied, the program branches to the label, LBL[x].

Refer to Section 6.9 for more information on branching. Refer to Section 6.12 for more information on the SKIP CONDITION instruction. Refer to Figure 6–23 for an example of the Skip, LBL[x] motion option.

Figure 6–23. SKIP LBL[x] Motion Option Example



6.4.10.6 Offset

```
OFFSET CONDITION PR[x]
J P[1] 50% FINE
Offset
```

The offset motion option is used with the OFFSET CONDITION instruction to alter positional information programmed at the destination position by the offset amount specified in a position register. The OFFSET CONDITION instruction defines the position register that contains the offset information. The OFFSET CONDITION instruction must be added to the program before the offset motion instruction.

The OFFSET CONDITION instruction shown uses the offset in position register 1, PR[x]. The offset motion instruction sets the positional information to position (P[1] + PR[x]) with the orientation of P[1]. When the offset condition is set, any time the offset motion option is used, that offset will be used. Refer to Section 6.14 for more information on offset instructions.

6.4.10.7 Offset Position Register

```
J P[1] 50% FINE
Offset, PR[x]
```

The Offset, PR[x] motion option alters positional information by the offset amount specified in the position register PR[x]. This offset affects only the motion instruction where it appears. It does not apply to any other motion instructions. The offset user frame number is the currently selected user frame number.

If $OFFSET_CART is TRUE, offsets for Cartesian positions are treated as frames and used to pre-multiply positions. If this is FALSE, offsets for XYZQPR positions are added field by field (for example, target.w=pos.w+offset.w).

The OFFSET calculation depends on the position register representation specified in the OFFSET motion option:

- **If PR[x] is Cartesian representation,** the system adds each element of the position register to each element of the position to yield the position that is offset. If the position does not have Cartesian representation, the system internally converts the representation of the position to Cartesian before the offset is calculated.
- **If PR[x] is JOINT representation,** the system adds each element of the position register to each element of the position to yield the position that is offset. If the position does not have JOINT representation, the system internally converts the representation of the position to JOINT before the offset is calculated. If PR[x] is JOINT representation, an offset user frame is not used.

```
J P[1] 50% FINE Offset, PR[x] Inc
J P[1] 50% FINE Offset Inc
```

- **If the incremental motion option is specified with the OFFSET motion option,** the position and position register MUST have the same representation, either Cartesian or JOINT. Before you define an offset in a motion instruction that also includes the INC motion option, make sure that the representations of the position register and position are the same. For example, if the position register is JOINT representation, the position must also be JOINT representation.

6.4.10.8 Tool_offset

```
TOOL_OFFSET_CONDITION PR[x] (UTOOL[1])
J P[1] 50% FINE
Tool_offset
```

The Tool_offset motion option is used with the TOOL_OFFSET_CONDITION instruction to alter positional information programmed at the destination position by the tool offset amount specified in a position register. The TOOL_OFFSET_CONDITION instruction defines the position register that contains the offset information and the tool frame that will be used during the tool offset. The TOOL_OFFSET_CONDITION instruction must be added to the program before the tool offset motion instruction.

A tool offset condition instruction specifies the offset condition used in a tool offset instruction. Execute a tool offset condition instruction before executing the corresponding tool offset instruction. After you specify the tool offset condition, it remains effective until the program terminates or the next tool offset condition instruction is executed.

When you specify tool offset conditions, be aware of the following:

- The position register specifies the direction in which the target position shifts, as well as the amount of shift.
- The tool coordinate system is used to specify offset conditions.
- When the number of a tool coordinate system is omitted, the currently selected tool coordinate system is used.
- When a motion statement which includes a tool offset instruction is taught or a certain position is modified, the position from which the offset is subtracted can be taught.
- When a motion statement which includes a tool offset instruction is taught or a certain position is modified, you will be asked to answer the following questions:
  - **Subtract tool offset data?** Press YES to subtract the tool offset data from the position data and accept the new position. Press NO to store the current position as the position data.
  - **Enter PR index of tool offset data?** Enter the position-register number specified by the tool offset condition instruction.
  - **Enter tool no. of tool offset data?** Enter the number of the tool coordinate system in which the offset is to be specified.
- If you manually modify the position data using the numeric keys, the position is taught without subtracting the offset.
- If you teach the position from which the offset is subtracted, the current position is stored in the following cases.
  - The specified position register has not yet been initialized
  - The tool offset instruction ignore function is enabled (see other setting.)
- If you enable the ignore function for the tool offset instruction, the current position is taught as position data and you will not receive any error messages. The robot moves to the taught position, even if a tool offset instruction is executed.
- If you pause the robot during the execution of a tool offset instruction and modify the shift amount, the modified amount will be used in the resumed movement.

If you modify a position register number specified by a tool offset condition instruction, the modified number will not be used.

- In backward execution, the robot is moved to the position to which the offset has been applied. This also applies to the direct tool offset instruction, described next.

6.4.10.9 Tool offset position register

```
J P[1] 50% FINE
 Tool_Offset, PR[2]
```

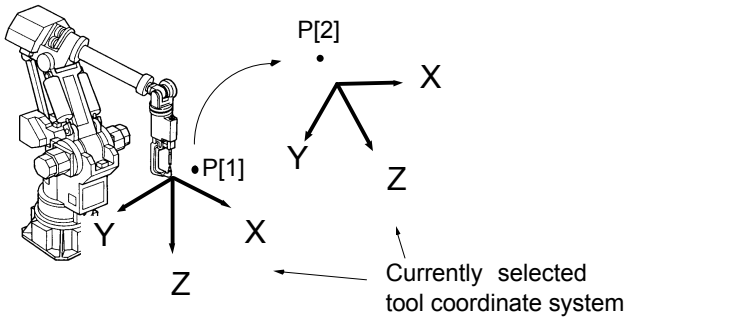
A direct tool offset instruction specifies the position register number. The robot moves according to the offset stored in the specified position register, ignoring the tool offset conditions specified by the tool offset condition instruction. The currently selected tool coordinate system is used.

When you specify tool offset position registers, be aware of the following:

- If you teach a motion statement which includes a direct tool offset instruction or you modify a certain position, you can teach the position from which the offset is subtracted. You will be asked to answer the following question.
  - **Subtract tool offset data?** Press YES to subtract the tool offset from the position data and accept the new position. Press NO to store the current position as position data.
- If you manually modify the position data using the numeric keys, the position is taught without subtracting the offset.
- If you teach the position from which the offset is subtracted, the current position is stored in the following cases.
  - The specified position register has not yet been initialized
  - The direct tool offset instruction has not specified the number of a position register
  - The tool offset instruction ignore function is enabled.
- If you enable the tool offset instruction ignore function, the current position is taught as position data (no prompt messages are output). The robot moves to the taught position even if a tool offset instruction is executed.

See Figure 6–24 for an example.

Figure 6–24. Tool Offset Instruction



OFFSET DATA

PR[1]		UF: F	UT: F
X:	0.000	W:	0.000
Y:	0.000	P:	0.000
Z:	10.000	R:	0.000

Example 1)
1: TOOL_OFFSET CONDITION PR[1]
2: J P[1] 100% FINE
3: L P[2] 500mm/sec FINE Tool_Offset

Example 2)
1: J P[1] 100% FINE
2: L P[2] 500mm/sec FINE Tool_Offset, PR[1]

6.4.10.10 Incremental Motion

```
J P[1] 50% FINE
INC
```

The incremental motion option specifies that the destination position is an incremental motion amount from the previous position. To use the incremental motion option, do the following:

Caution

If you use the incremental motion option in a motion instruction, the position or position register in that instruction will be uninitialized. Also, all instances of that same position or position register in your program will be uninitialized.

If you do not want this to happen, use a new position or position register in the motion instruction that will include the incremental motion option. If you want to use the same incremental motion elsewhere in your program, copy the entire motion instruction and paste it where you want to use it.

1. Add a motion instruction. Do not include the incremental motion option.
2. Add another motion instruction. Be sure to include the incremental motion option.
3. Move the cursor to the right of the motion instruction you just added.
4. Press F4, [CHOICE].
5. Select Incremental. You will see the message, "Position(P[n]) has been uninitialized."
6. Move the cursor to the position component of the instruction and press F5, POSITION. Each position component will be set to uninitialized and the position representation screen will be displayed. See Figure 6–25 .

Figure 6–25. Position Representation Screen

```
Position Detail
 P[2]  UF:0 UT:1      conf: N       0   0
 X ******.***  mm   W ******.*** deg
 Y ******.***  mm   P ******.*** deg
 Z ******.***  mm   R ******.*** deg
```

Note If your program is set up with multiple groups or extended axes, you must enter appropriate values in the extended axes and group position components in order for the motion instruction to be executed.

7. Move the cursor to each position component you want to change, type the increment you want the robot to move, and press ENTER. If you do not want to change a component, set it to zero.

6.4.10.11 Search [ ] Motion Option

```
J P[1] 50% FINE
Search[ ]
```

The Search [ ] motion option directs the motion of the robot (in a positive or negative x, y or z direction) to search for an object. The x, y and z vectors are defined by the touch frame assigned in the touch schedule. When contact is made with the object, the robot's current TCP position is stored. The search motion option must be used between a search start and search end statement.

Note Touch Sense is an option and might not be installed on your system. If Touch Sense is not installed, Search will not appear as a menu item.



Caution

Motion speed and direction are controlled by values set in the touch schedule assigned by the Search Start instruction. The motion and speed might be different than what is displayed on the line.

6.4.10.12 Extended Velocity EV Motion Option

In addition to the programmed robot speed, the extended velocity (EV) motion option allows the specification of the programmed extended axis speed. The EV motion option has the following two options:

- Simultaneous EV
- Independent EV

Simultaneous EV

```
J P[1] 100% FINE
 EV50%
```

The programmed simultaneous EV is defined as a percentage of the maximum extended axis speed (1% - 100%).

If the EV motion option is not specified, then the extended axis motion is planned based on the maximum extended axis speed. This means that the default motion without the EV option is equivalent to simultaneous motion with EV100%.

In simultaneous EV, the extended axis moves simultaneously with the robot axes. This means that they both start and end at the same time for each motion segment.

In order to achieve simultaneous motion, the robot motion time is compared with the extended axis segment time during planning. The longer time will be used for both the robot and the extended axis so that they both reach the destination at the same time.

In cases where the robot motion time is longer than the extended axis motion time, the actual extended axis speed will be lower than its programmed extended axis speed so that robot motion speed is maintained.

When the extended axis motion time is longer than the robot motion time, the actual robot speed will be slower than its programmed speed in order to maintain simultaneous motion.

When there is extended axis motion but no robot motion, the programmed extended axis speed will be used as specified, even if it could be the default maximum speed.

Independent EV

```
J P[1] 100% FINE
 Ind.EV50%
```

Like simultaneous EV, the programmed independent Extended Velocity is also defined as a percentage of the maximum extended axis speed (1% - 100%).

In independent EV, the extended axis moves independently of the robot axes. Both the extended axis and the robot axes start each motion segment at the same time, however, because of their independent speed rates, they might not reach the destination at the same time. The next planned motion cannot execute until both the extended axis and the robot axes have reached the destination.

6.4.10.13 PTH Motion Option

```
J P[1] 50% CNT100
PTH
```

The PTH motion option allows you to increase the robot acceleration between positions in a series of positions, or path. You can use the PTH motion option only in motion instructions that use continuous termination type.

If you have a short series of continuous positions that are relatively close together, use the PTH motion option with each motion instruction to increase the acceleration between each position. This will reduce the amount of time the robot takes to execute that portion of the program.

Caution

If motion instructions that contain the PTH motion option produce jerky motion or vibration in the robot, remove the PTH motion option from the motion instruction.

6.4.10.14 Time Before / Time After

TIME BEFORE Motion Option

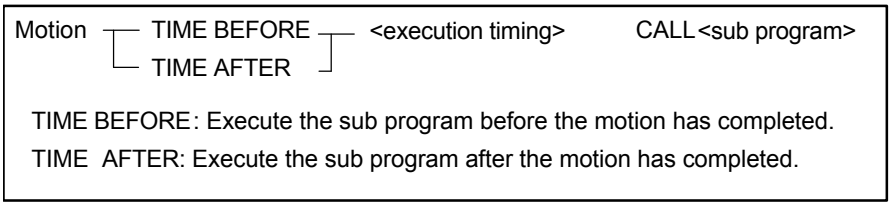
```
J P[1] 50% FINE
TIME BEFORE 2.0 sec, CALL prog
```

TIME AFTER Motion Option

```
J P[1] 50% FINE
TIME AFTER 2.0 sec, CALL prog
```

Normally, when a teach pendant program is executed, the instruction that follows a motion instruction is not executed until the motion has been completed. The TIME BEFORE/AFTER motion option instruction allows you to specify a teach pendant program that is to be called at a specified time before or after the completion of a motion instruction. See Figure 6–26 .

Figure 6–26. TIME BEFORE / TIME AFTER Motion Option Instructions



Refer to the “Advanced Functions” chapter in the Setup and Operations Manual for more information on the TIME BEFORE and TIME AFTER motion options.

6.4.10.15 Remote TCP Motion Option (optional)

```
L P[1] 100mm/sec CNT100
RTCP
```

The optional remote TCP motion option (RTCP) allows you to control the orientation of the robot in applications where the tool is fixed in the workcell and the robot manipulates the workpiece around the tool. The frame used for jogging and programming is a user frame you set up and select.

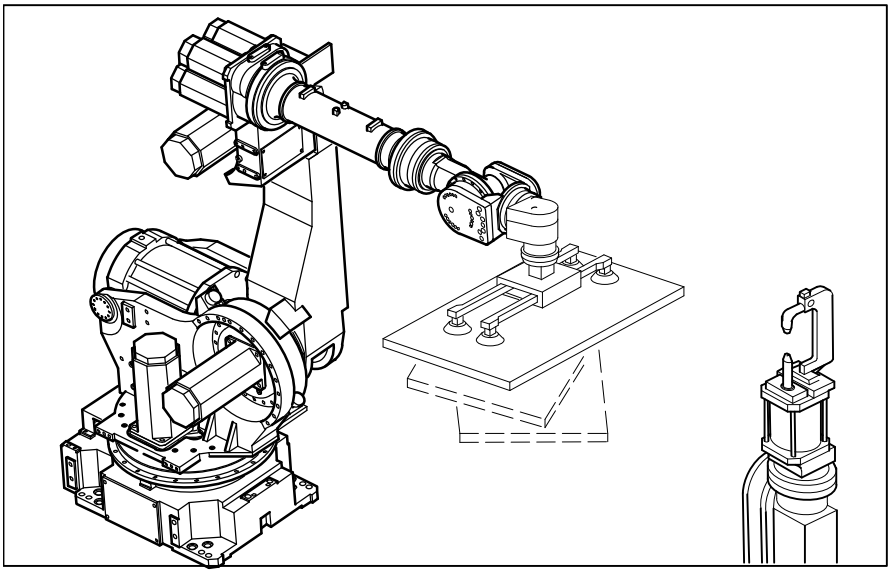
See Figure 6–27 for an illustration of the robot using remote TCP. The tool is fixed and the robot is holding the workpiece.

Note If the Coordinated Motion option is loaded on the controller, the RTCP motion option will not be available. If an existing program that already contains an RTCP instruction is executed, the following error will be posted at the RTCP instruction: RTCP-014 “RTCP not supported with COORD”. Loading the Coordinated Motion option on the controller completely disables Remote TCP.

Note Remote TCP jogging is not available with the following options:

- Automatic Voltage Control (AVC)
- Coordinated Motion
- Multi-Arm
- Scratch Start
- TAST
- RPM/Multi-Pass
- Touch Sensing
- Weaving

Figure 6–27. Remote TCP Motion Option



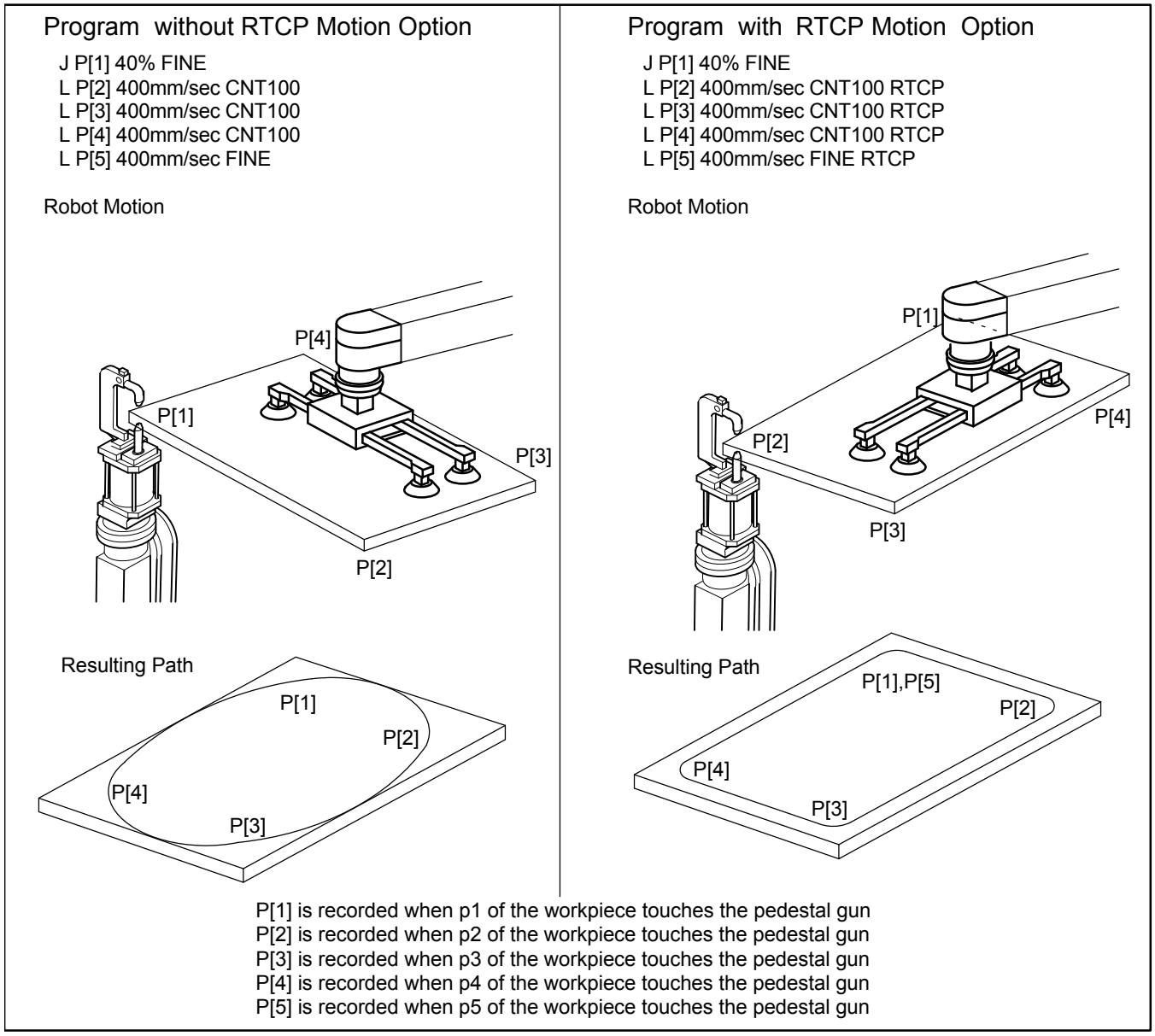
When you use remote TCP, you must first set up the user frame you will use as the remote TCP frame. When you include the remote TCP (RTCP) motion option in a motion instruction, use the UFRAME_NUM= instruction to specify the user frame you want to use; otherwise the current user frame will be used by default.

Refer to Section 6.14 for more information on the UFRAME_NUM= instruction.

Note In a motion instruction that includes RTCP, the speed specified is the relative speed between the workpiece and the tool.

Figure 6–28 contains an example of how to use the RTCP motion option.

Figure 6–28. Remote TCP (RTCP) Motion Option Example



P[1] is recorded when p1 of the workpiece touches the pedestal gun
P[2] is recorded when p2 of the workpiece touches the pedestal gun
P[3] is recorded when p3 of the workpiece touches the pedestal gun
P[4] is recorded when p4 of the workpiece touches the pedestal gun
P[5] is recorded when p5 of the workpiece touches the pedestal gun

Limitations when used with ArcTool

RTCP does not support the following ArcTool features:

- Weave
- Touch sensing
- TAST
- AVC

- MigEye
- RPM
- MultiPASS
- Multiple group program

If you attempt to program RTCP when any of these features are loaded, the following error message will be displayed:

```
RTCP+Option not supported
```

See the following examples:

Example 1

```
1. Weave Sine[1]
2. L P[]  8 mm/sec CNT100
3  L P[]  8 mm/sec CNT100   RTCP
4. Weave End
```

The error will be posted on Line 3.

Example 2

```
1. Search Start[1] PR[2]
2. L P[]  8 mm/sec CNT100
3  L P[]  8 mm/sec CNT100   RTCP Search[X]
4. L P[]  8 mm/sec CNT100
5  L P[]  8 mm/sec CNT100   RTCP Search[Y]
4. Search End
```

The error will be posted on Lines 3 and 5.

The following examples are for a multiple group program.

Example 3

```
1. L P[]  8 mm/sec CNT100
2. L P[]  8 mm/sec CNT100
3  L P[]  8 mm/sec CNT100   RTCP
```

The error will be posted on Line 3.

Example 4

```
1. Track AVC[1]
 2. L P[]  8 mm/sec CNT100
 3  L P[]  8 mm/sec CNT100   RTCP
 4. Track End
```

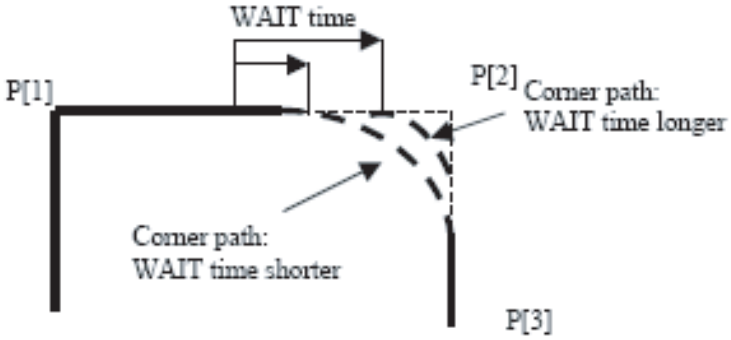
The error will be posted on Line 3.

6.4.10.16 Break

BREAK is a motion option that does not start the next motion segment until the cursor moves to the motion line in the TP program. With BREAK option in a motion statement, the constant path feature may not be maintained. BREAK can be used with the WAIT statement for applications that need to change the corner path depending on WAIT time, as shown in the following example.

Corner path changes as wait time varies: the longer the tighter

```
1 J P[1] 50% FINE
2 L P[2] 500 mm/sec CNT100 BREAK
3 R[1] = 1
4 Wait DI[1] = on
5 L P[3] 500 mm/sec CNT0
```



6.4.10.17 Constant Path

Constant Path is a motion control option that provides enhanced motion performance for all motion types in the following areas:

- **Constant path**

  With Constant Path, the robot maintains the same path regardless of static or dynamic speed override changes. A path that has been taught and tested at a low speed override will be maintained when the program is executed at 100% override.

- **Constant Path with respect to T1/T2/Auto Mode**

  With Constant Path, the robot maintains the same path in different modes. For example, a path that has been taught and tested in T1 mode will be maintained when the program is executed in Auto mode.

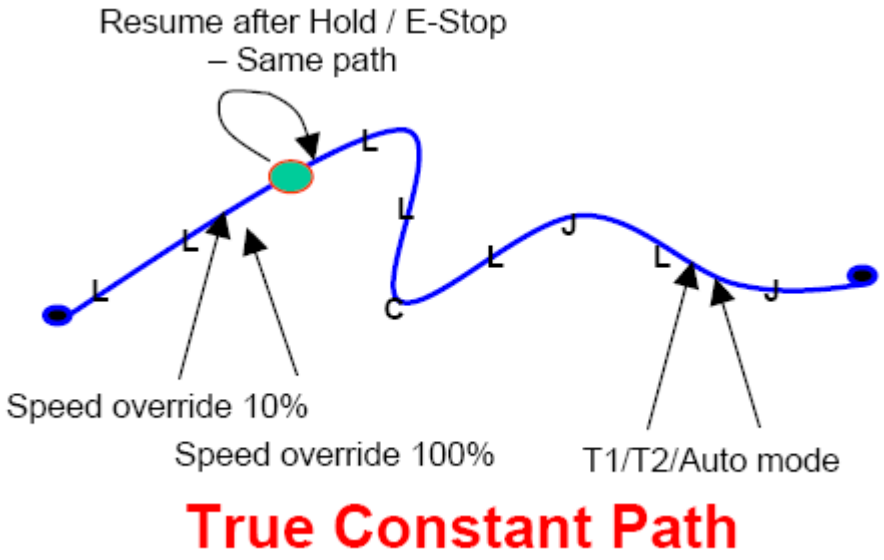
- **Hold/Resume and Emergency Stop/Resume** (HandlingTool, PaintTool, and SpotTool+ only)

  After the HOLD or EMERGENCY STOP buttons have been pressed, the robot can resume execution along the same path that was being executed prior to the HOLD or EMERGENCY STOP. The location and orientation of the resumed path will be along the original path.

For exceptions, refer to the Limitations section.

- **Enhanced path accuracy**

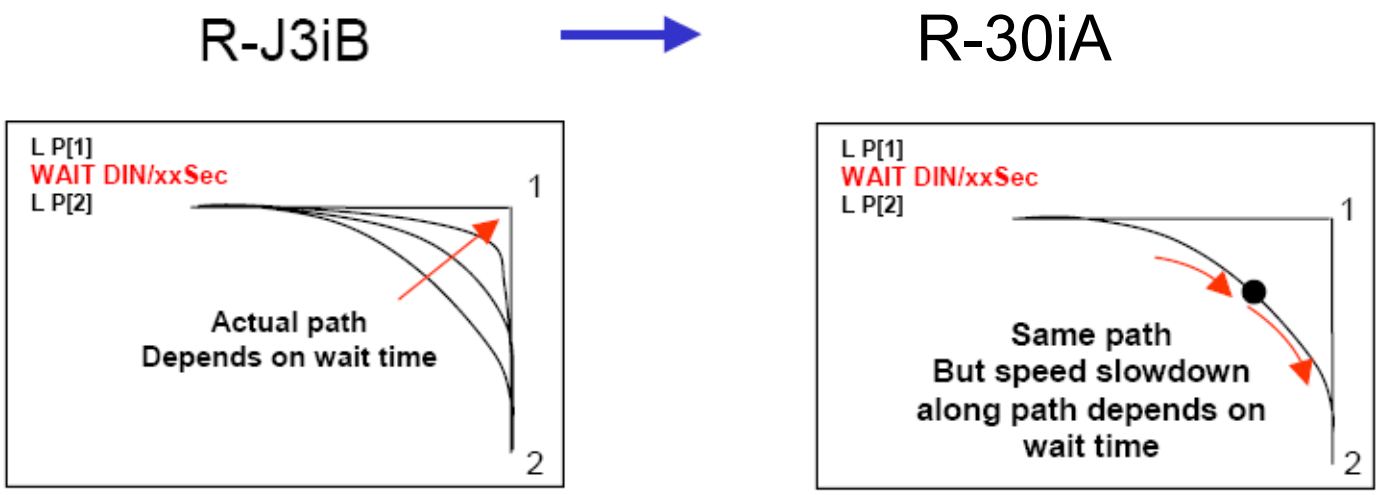
  The path will be executed as taught, using a straight line or circular motion.



True Constant Path

- **Constant Path Regardless of WAIT Statements**
  - — Maintains same path regardless of duration of Wait I/O instruction
  - — Maintains same path regardless of duration of Wait xx sec instruction. The robot will decelerate along the path until the WAIT instruction expires. If the WAIT duration is long enough, the robot will decelerate to a stop. After the WAIT instruction expires, the robot will accelerate and resume the original path.

Figure 6–29.



To get R-J3iB behavior, use the BREAK motion option:

```
1: L P[1] 2000 mm/s CNT100 BREAK
2: WAIT DIN/xxSec
3: L P[2] 2000mm/s CNT100
```

In this example, Line 3 will not affect motion until the WAIT expires. The path will shift toward P[1], depending on WAIT duration.

Semi-Hot Start Limitation

Constant path cannot be maintained through a Semi-Hot Start cycle. When the program is resumed, the robot will move toward the taught position of the paused line without blending of previous lines. If original path resume is enabled, the robot will move to the stop position before moving toward the taught position.

Teach Pendant Instruction Limitations with respect to Hold, Stop, Resume, and Override Along the Path

Warning

Some instructions cannot assure Constant Path motion because they dynamically change program execution. These instructions do not necessarily result in path variation (with respect to modes, WAITs, and overrides), but path variation is possible.

The instructions in this category are as follows:

- Frame instructions: UFRAME_NUM, UFRAME, UTOOL_NUM, UTOOL
- Branching instructions: IF, SELECT, CALL
- Miscellaneous instruction: $PARAMETER = ...
- Program control instructions: PAUSE, ABORT
- Macro program instruction
- SKIP instruction
- TRACK instruction
- Variable motion speed instructions
- Sensor instructions: RCV, SENSOR_ON, SENSOR_OFF
- Palletizing instruction: PALLETIZING-B, PL[ ]
- BREAK motion option

The following instructions are constant path when the position registers are locked. When position registers are unlocked, the path may vary.

- Position register instructions: PR[ ], PR[ ] INC
- Offset instructions: OFFSET, TOOL_OFFSET

Constant Path Look Ahead Limitation



Caution

The amount of segment look ahead available for determining the path is limited. If there are not enough segments available to identify the path, the path could deviate toward the taught point of the last available segment. Limiting the number of segments that blend together at the same time helps to avoid this problem.

Programming Guideline

To allow the best constant path functionality, avoid teaching a path with several of these characteristics:

- ACC< 100
- High CNT values
- High processor loading
- High speed
- Many segments in a short distance
- Short segment lengths

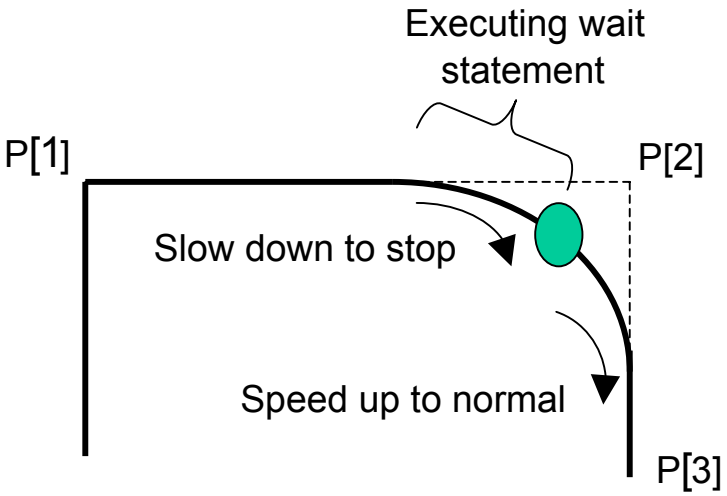
Constant path behavior for motions with WAIT statement

1. **Normal execution**

   With R-30*i*A motion system loaded and enabled, while executing WAIT statement, when possible the robot will decelerate along the path of the motion lines that follow the WAIT statement. If the WAIT duration is long, the robot will decelerate to a stop on the path. Some teach pendant instructions do not support constant path during WAIT: refer to Section 6.10 for teach pendant instruction limitations.

   After the WAIT statement completes, the robot will accelerate to normal speed and continue the rest of the path. The path will remain the same regardless of WAIT time. For example,

```
1 J P[1] 50% FINE
2 L P[2] 500 mm/sec CNT100
3 R[1] = 1
4 Wait DI[1] = on
5 L P[3] 500 mm/sec CNT0
```

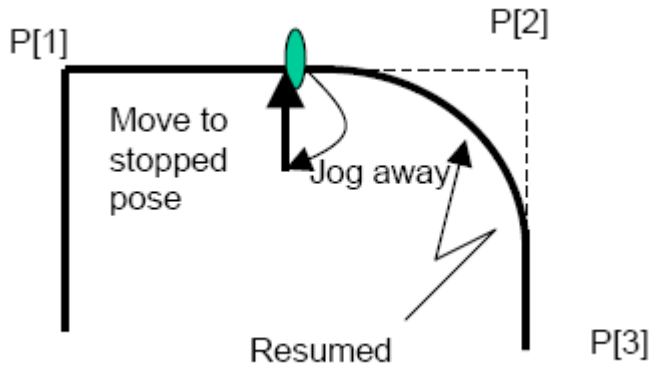


2. **Hold/resume**

   The R-30*i*A motion system will maintain the same path regardless of WAIT time when possible, even when Hold/resume interrupts the teach pendant motions near a WAIT statement.

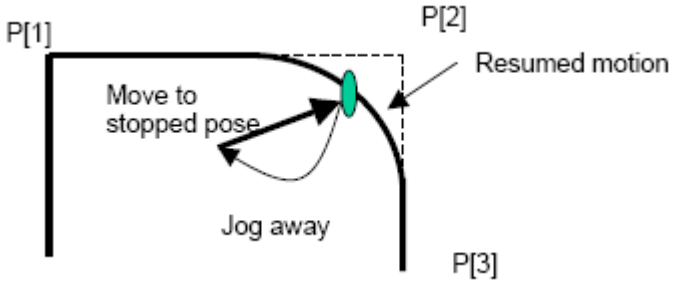
   a. case 1: Hold/E-stop at the motion line prior to WAIT statement

```
          1 J P[1] 50% FINE
cursor -> 2 L P[2] 500 mm/sec CNT100
          3 R[1] = 1
          4 Wait DI[1] = on
          5 L P[3] 500 mm/sec CNT0
```



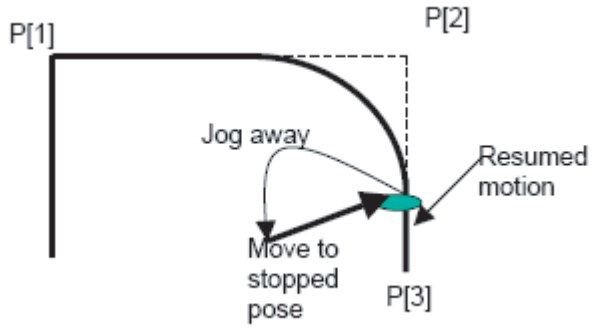
b. case 2: Hold/E-stop at WAIT statement (cursor at non-motion line)

```
           1 J P[1] 50% FINE
           2 L P[2] 500 mm/sec CNT100
           3 R[1] = 1
cursor ->  4 Wait DI[1] = on
           5 L P[3] 500 mm/sec CNT0
```



c. case 3: Hold/E-stop at motion line after WAIT statement

```
           1 J P[1] 50% FINE
           2 L P[2] 500 mm/sec CNT100
           3 R[1] = 1
           4 Wait DI[1] = on
cursor ->  5 L P[3] 500 mm/sec CNT0
```



3. **Single step execution after Hold/E-stop**

   a. **Single step Forward execution after Hold/E-stop**

   Assume that Hold/E-stop occurs when single step execution is not enabled. If single step forward execution is enabled after Hold/E-stop occurs, then users will observe (1) the first single step forward: robot moves to stopped pose (may be zero distance move if not jogging away) and then posts "CPMO-069 can't resume with STEP (G:1)" is posted (2) single step forward again: robot moves to the destination position of the motion line at cursor

   Caution

   Single step execution (FWD and BWD) is relative to the TP cursor line, not current robot position. After Hold or E-stop the current robot position may be far from the taught point indicated by the TP cursor (especially in T1 mode or low override). Single step execution after Hold/E-stop will move to the taught point indicated by the TP cursor, not on the original program path. A collision could occur if the single step execution path is obstructed.

   i. case 1: Hold/E-stop at the motion line prior to WAIT statement

   Single step forward execution will move robot to the destination of the cursor motion line.

   After HOLD/E-STOP

```
           1  J  P[1] 50% FINE
cursor ->  2  L P[2] 500 mm/sec CNT100
           3  R[1] = 1
           4  Wait DI[1] = on
           5  L P[3] 500 mm/sec CNT0
```

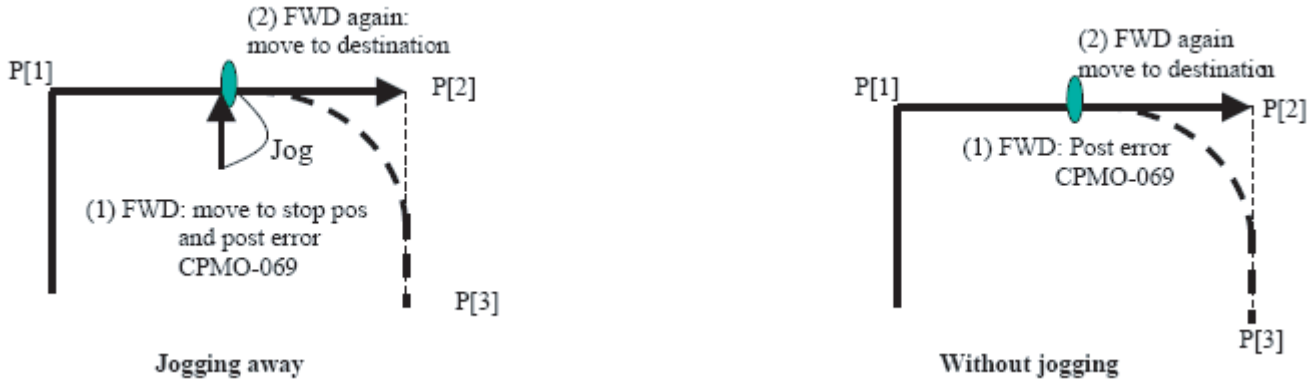
After 1st SSTEP FWD

```
           1  J  P[1] 50% FINE
cursor ->  2  L P[2] 500 mm/sec CNT100
           3  R[1] = 1
           4  Wait DI[1] = on
           5  L P[3] 500 mm/sec CNT0
```

After 2nd SSTEP FWD

```
           1  J  P[1] 50% FINE
cursor ->  2  L @P[2] 500 mm/sec CNT100
           3  R[1] = 1
           4  Wait DI[1] = on
           5  L P[3] 500 mm/sec CNT0
```

Figure 6–30.



ii. case 2: Hold/E-stop at WAIT statement (cursor at non-motion line)

Single step forward execution will move the robot to the destination of the motion line prior to the WAIT statement.

After HOLD/E-STOP

```
     1  J  P[1] 50% FINE
     2  L P[2] 500 mm/sec CNT100
     3  R[1] = 1
->   4  Wait DI[1] = on
     5  L P[3] 500 mm/sec CNT0
```

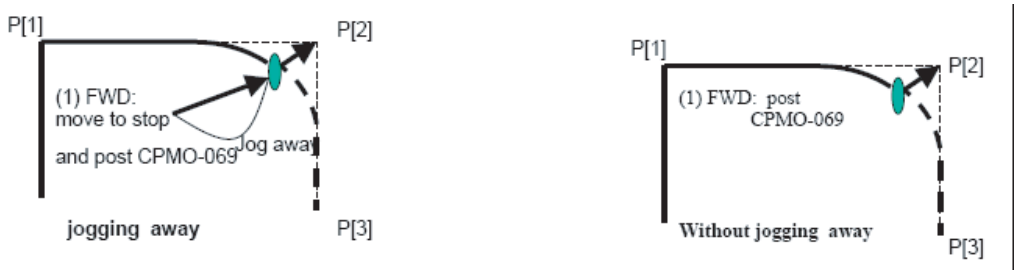
After 1st SSTEP FWD

```
     1  J  P[1] 50% FINE
->   2  L P[2] 500 mm/sec CNT100
     3  R[1] = 1
     4  Wait DI[1] = on
     5  L P[3] 500 mm/sec CNT0
```

After 2nd SSTEP FWD

```
     1  J  P[1] 50% FINE
->   2  L @P[2] 500 mm/sec CNT100
     3  R[1] = 1
     4  Wait DI[1] = on
     5  L P[3] 500 mm/sec CNT0
```

Figure 6–31.



Note As shown in this example, the robot position is more likely to be close to P[2] than P[3] because the previous motion line defines the corner location.

iii. case 3: Hold/E-stop at the motion line after WAIT statement. Single step forward execution will move robot to the destination of the cursor motion line.

After HOLD/E-STOP

```
     1  J  P[1] 50% FINE
     2  L P[2] 500 mm/sec CNT100
     3  R[1] = 1
     4  Wait DI[1] = on
->   5  L P[3] 500 mm/sec CNT0
```

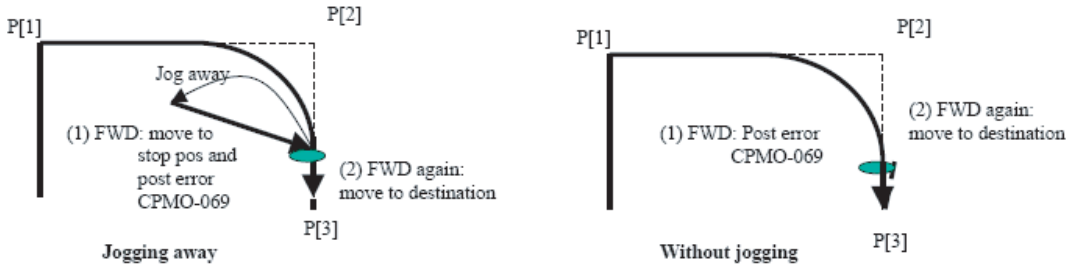
After 1st SSTEP FWD

```
     1  J  P[1] 50% FINE
     2  L P[2] 500 mm/sec CNT100
     3  R[1] = 1
     4  Wait DI[1] = on
->   5  L P[3] 500 mm/sec CNT0
```

After 2nd SSTEP FWD

```
     1  J  P[1] 50% FINE
     2  L P[2] 500 mm/sec CNT100
     3  R[1] = 1
     4  Wait DI[1] = on
->   5  L P[3] 500 mm/sec CNT0
```

Figure 6–32.



b. **Single step Backward execution after Hold/E-stop**
Assume that single step execution is not enabled at the time of the Hold. Single step backward execution after Hold/E-stop will cause the robot to move back to the destination position of the previous motion line.

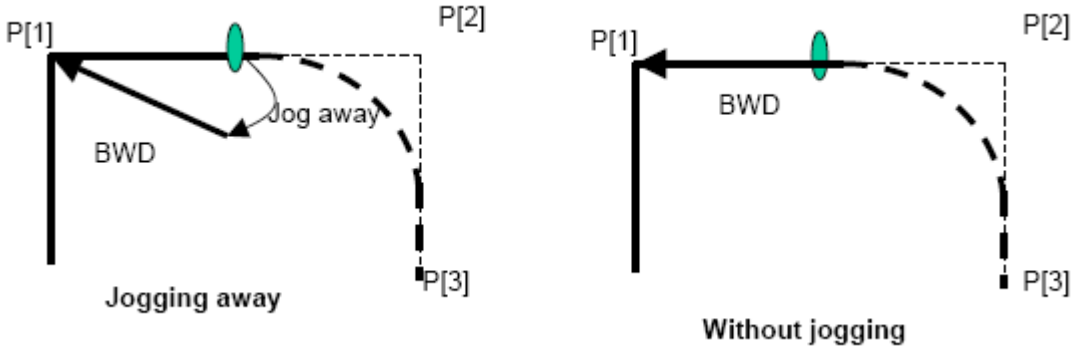
i. case 1: Hold/E-stop at the motion line prior to WAIT statement. Single step backward goes to destination of previous motion line. After HOLD/E-STOP

```
    1  J  P[1] 50% FINE
->  2  L P[2] 500 mm/sec CNT100
    3  R[1] = 1
    4  Wait DI[1] = on
    5  L P[3] 500 mm/sec CNT0
```

After BWD

```
 -> 1  J  @P[1] 50% FINE
    2  L P[2] 500 mm/sec CNT100
    3  R[1] = 1
    4  Wait DI[1] = on
    5  L P[3] 500 mm/sec CNT0
```

Figure 6–33.



ii. case 2: Hold/E-stop at WAIT statement (i.e. cursor at non-motion line)

Caution

Single step backward goes to destination of second previous motion line. (Single step forward goes to destination of previous motion line.)

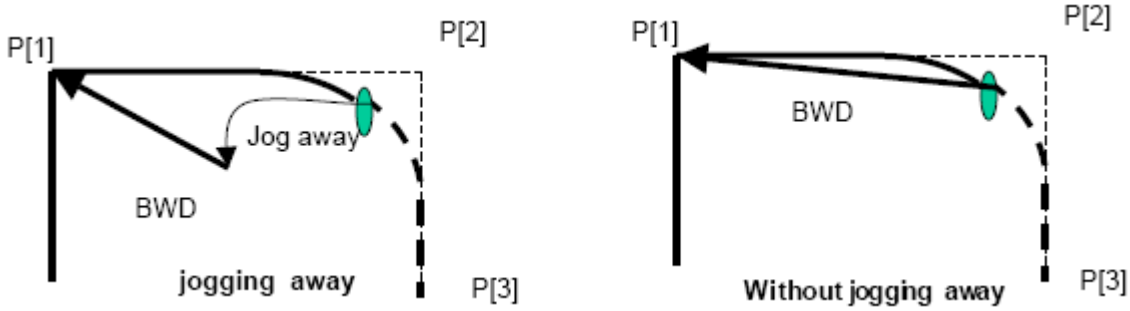
After HOLD/E-STOP

```
    1  J  P[1] 50% FINE
    2  L P[2] 500 mm/sec CNT100
    3  R[1] = 1
->  4  Wait DI[1] = on
    5  L P[3] 500 mm/sec CNT0
```

After BWD

```
->  1  J  @P[1] 50% FINE
    2  L P[2] 500 mm/sec CNT100
    3  R[1] = 1
    4  Wait DI[1] = on
    5  L P[3] 500 mm/sec CNT0
```

Figure 6–34.



iii. case 3: Hold/E-stop at the motion line after WAIT statement. Single step backward moves to destination of previous motion line.

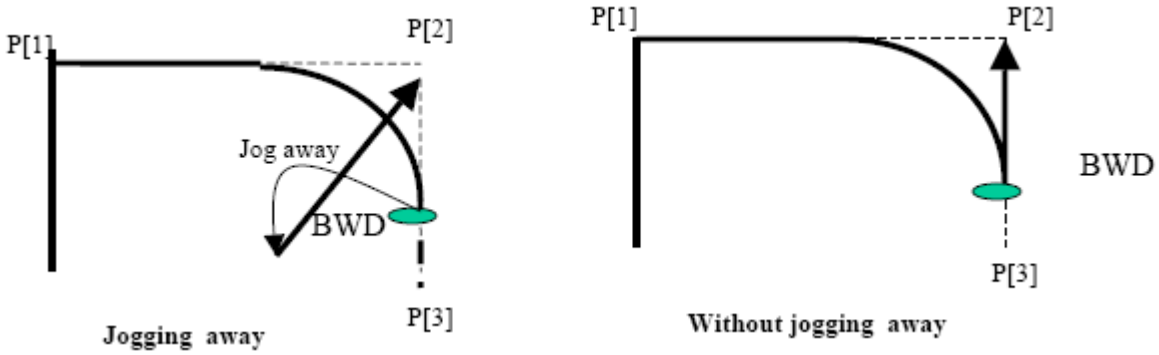
After HOLD/E-STOP

```
    1  J  P[1] 50% FINE
    2  L P[2] 500 mm/sec CNT100
    3  R[1] = 1
    4  Wait DI[1] = on
->  5  L P[3] 500 mm/sec CNT0
```

After BWD

```
     1  J  P[1] 50% FINE
->   2  L @P[2] 500 mm/sec CNT100
     3  R[1] = 1
     4  Wait DI[1] = on
     5  L P[3] 500 mm/sec CNT0
```

Figure 6–35.



6.4.10.18 Advanced Constant Path

In addition to the Constant Path feature the Advanced Constant Path package consists of the following functions for easy teaching and cycle time reduction.

- Linear Distance
- Corner Region
- Process Speed
- Max Speed

It requires that the Constant Path option be loaded and enabled.

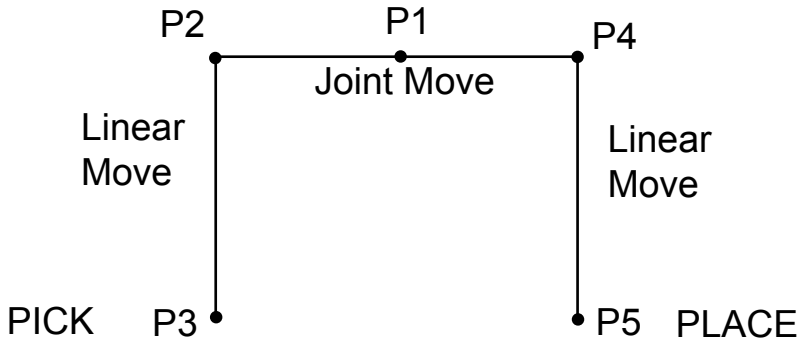
Linear Distance

Overview

Linear Distance is a robot motion feature that is useful for pick and place material handling applications such as palletizing.

Figure 6–36 shows a typical pick and place application. If all termination types are FINE or CNT0, then the pick and place path would be as shown.

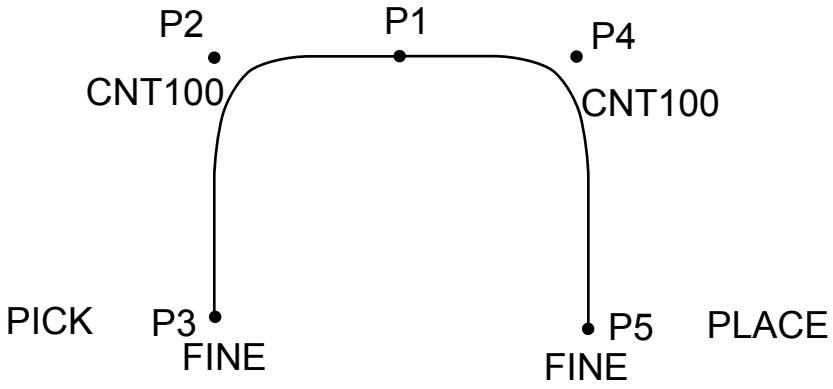
Figure 6–36. PICK and PLACE Application



The robot starts at P1 and goes through P2 and picks up a part at P3. It then goes through P2 to P4 and then to P5 where it places the part.

Typically, however, FINE and CNT0 is used only to reach P3 for PICK and to PLACE at P5. For all other motions, high CNT values are used. For instance, the actual path might look like Figure 6–37 .

Figure 6–37. PICK and PLACE Application with CNT100

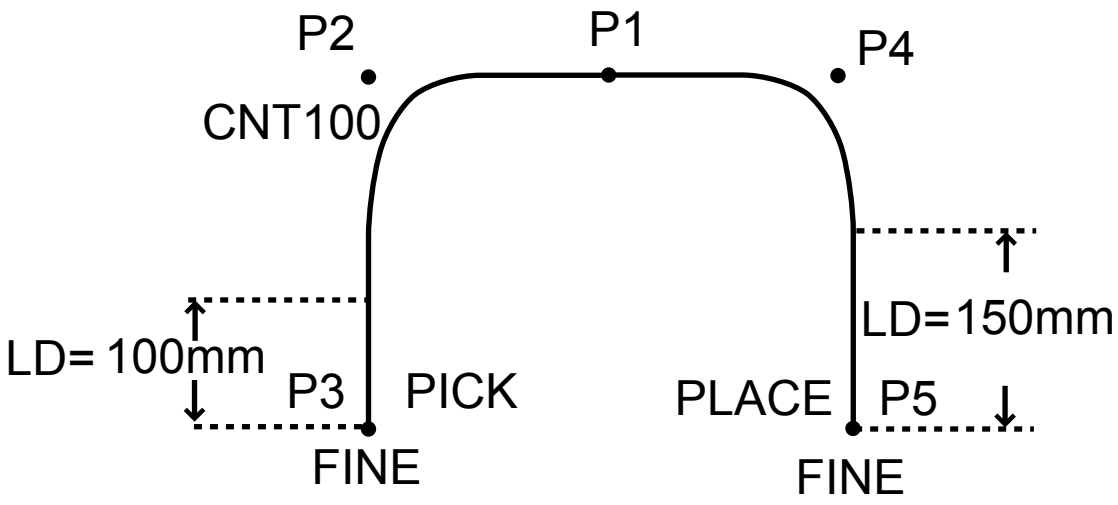



```
Robot is at P1
J P[2] 100% CNT100
L P[3] 2000 mm/s FINE
L [P2] 2000 mm/s CNT100
J [P4] 2000 mm/s CNT100
L [P5] 2000 mm/s FINE
```

This kind of a path will give you better cycle time. However, you do not know how much linearity you will get going from P2 to P3 (for pick) or from P4 to P5 (for place). In order to get the desired linearity, you would adjust either the positions P2 or P4, or you would experiment with different CNT values. For example, you might shift P2 or P4 higher or lower, or you might use CNT50 at P2 or P4.

With Linear Distance, you do not have to guess and experiment. If you want the last 100mm before pick and the last 150mm above place to always be straight above P3 and P5 respectively, you can use Linear Distance for specifying these amounts. Refer to Figure 6–38 .

Figure 6–38. Adjusting P3 and P5 with Linear Distance



The program would be (from P1):

```
J P[2] 100% CNT100
L P[3] 2000 mm/s FINE AP_LD100
L P[2] 2000 mm/s CNT100 RT_LD100
J P[4] 2000 mm/s CNT100
L P[5] 2000 mm/s FINE AP_LD150
```

This is a convenient way to design motions for pick and place. By only adjusting the linear distance, you can determine the appropriate tradeoff between cycle time and approach/retract linearity.

Note Linear Distance is implemented on the controller to control the linearity immediately after a pickup at the infeed and just before placement on the pallet. Linear Distance guarantees the distances you specify. Due to ITP resolution, Linear Distance cannot match your value exactly. However, the linear distance that you specify will be the minimum value that system will try to achieve. For example, if you specify 100 mm linear distance, the system might provide 103 mm linear distance, but it will never provide linear distance less than what you specified.

How it Works

Linear Distance uses two motion option elements:

- AP_LD: Approach Linear Distance

  AP_LD is used for a place motion.

```
L P[2] 2000mm/s CNT0 AP_LD100
L P[2] 2000mm/s CNT0 AP_LDR[1]
```

In the second example, the distance is specified indirectly via register #1.

- RT_LD: Retract Linear Distance

RT_LD is used for a pick-up motion.

```
L P[3] 2000mm/s CNT100 RT_LD100
L P[3] 2000mm/s CNT100 RT_LDR[1]
```

In the second example, the distance is specified indirectly via register #1.

Figure 6–39. RT_LD: Effect of CNT Value

WITHOUT USING LINEAR DISTANCE

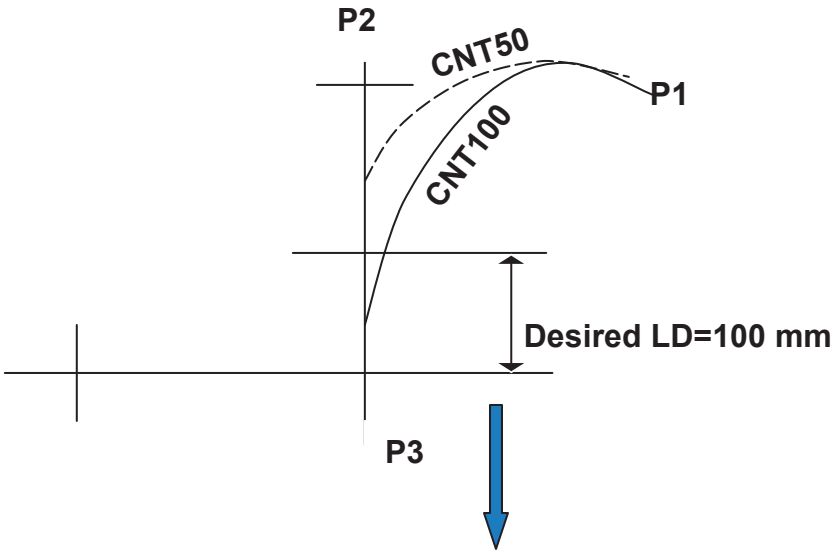
Robot is at P3

Dashed line example:

```
L P[2] 2000 mm/s CNT50
J P[1] 100% CNT100
```

Solid line example:

```
L P[2] 2000 mm/s CNT100
J P[1] 100% CNT100
```



USE OF LINEAR DISTANCE

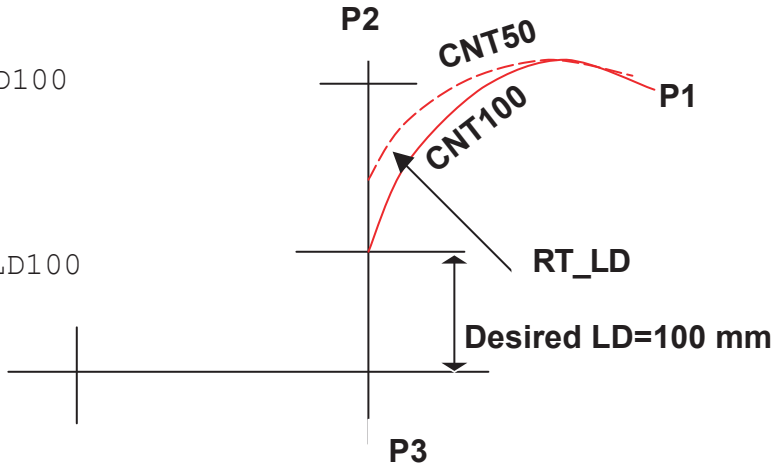
Robot is at P3

Dashed line example:

```
L P[2] 2000 mm/s CNT50 RT_LD100
J P[1] 100% CNT100
```

Solid line example:

```
L P[2] 2000 mm/s CNT100 RT_LD100
J P[1] 100% CNT100
```



The RT_LD value affects the corner of P3-P2-P1 in Figure 6–39 . The higher the value of RT_LD, the smaller the corner will be. When the RT_LD value is greater than or equal to the distance between

P3 to P2, the corner will become 0. The motion will automatically become FINE regardless of the CNT value you specify.

Figure 6–40. Place Motion: Two Possible Traces

WITHOUT USING LINEAR DISTANCE

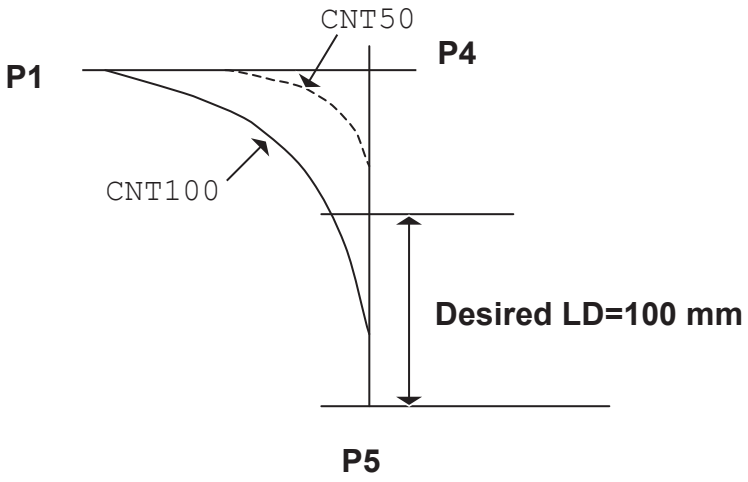
Robot is at P1

Dashed line example:

```
J P[4] 100% CNT50
L P[5] 2000 mm/s CNT100
```

Solid line example:

```
J P[4] 100% CNT100
L P[5] 2000 mm/s CNT100
```



USE OF LINEAR DISTANCE

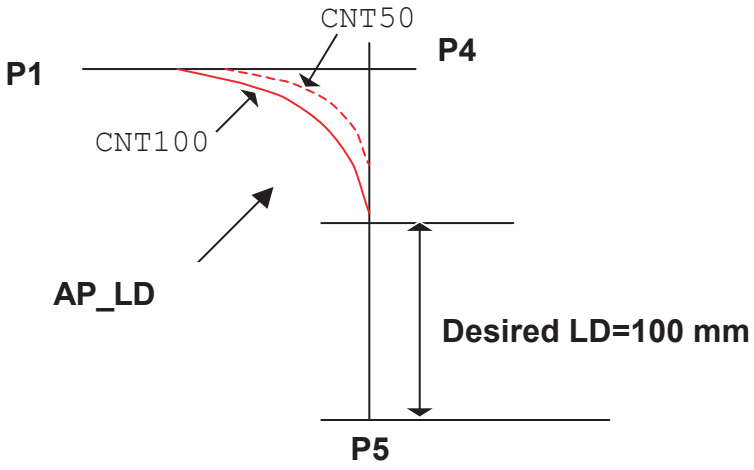
Robot is at P1

Dashed line example:

```
J P[4] 100% CNT50
L P[5] 2000 mm/s CNT0 AP_LD100
```

Solid line example:

```
J P[4] 100% CNT100
L P[5] 2000 mm/s CNT0 AP_LD100
```



The AP_LD value affects the corner of P1-P4-P5 Figure 6–40 . The higher the value of AP_LD, the smaller the corner will be. When the AP_LD value is greater than or equal to the distance between P4 to P5, the corner will become 0. The preceding motion (motion from P1 to P4) will become FINE regardless of the CNT value for that move.

Limitations

- The linear distance function only supports motion group 1 by default. Set the system variable $LDCFG.$group_msk to 3 to support a second group.
- The linear distance function only supports the linear motion type.
- The linear distance function only supports position (x, y, and z) and not orientation (w, p, and r).
- The linear distance function only supports articulated robots and does not support Independent Axes or Positioners.
- The local condition trigger time might be different than without Linear Distance. However the timing is repeatable.
- When multiple group motion is used, the motion will be synchronized. However, if more than one group has linear distance enabled, all the groups will have linear distance satisfied.
- When using max_speed (refer to max speed section below), the specified Linear Distance may not be guaranteed.
- Linear Distance function does NOT support Coordinated Motion (Linear Distance function will be automatically disabled for motions with COORD option).
- Linear Distance function does NOT support Continuous Motion Types such as Weave, Continuous Turn, and Robot Link. (The Linear Distance function will be automatically disabled with the continuous motion type.)
- Linear Distance function might not work with TCP speed prediction function (TCPP). With Linear Distance specified, TCPP results might not be correct.

How to Use Linear Distance

Note Due to ITP resolution, Linear Distance cannot match your value exactly. However, the linear distance that you specify will be the minimum value that system will try to achieve. For example, if you specify 100 mm linear distance, the system might provide 103 mm linear distance, but it will never provide linear distance less than what you specified.

Procedure 6-2 Using Linear Distance

Conditions

- You have created a teach pendant program.
- Your teach pendant program contains at least one linear motion instruction.

Steps

1. Press SELECT.
2. Move the cursor to the name of the program you want to modify and press ENTER.
3. Continuously press the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.
4. To touch up and modify motion instructions, move the cursor to the line number of the motion instruction you want to modify.

Note To use Linear Distance, you must modify a linear motion instruction.

5. Move the cursor to the empty space at the end of the linear motion instruction that you want to modify and press F4, [CHOICE]. You will see a screen similar to the following.

```
Motion Modify                JOINT  10 %
 1 Retract_LD       5 TIME BEFORE
 2 Approach_LD      6 TIME AFTER
 3 Tool_Offset      7 DISTANCE BEFORE
 4 Tool_Offset,PR[  8 ---next page---
LD
                                    5/8
   4:J  P[2] 100% CNT100
   5:L  P[2] 2000mm/sec CNT10
    :  Offset,PR[1] AP_LDR[1]
   6:L  P[2] 2000mm/sec CNT100
    :  RT_LDR[1]
   7:J  P[1] 100% CNT100
Select item
```

6. If you want to use Linear Distance on an approach point, select Approach_LD.

   If you want to use Linear Distance on a retract point, select Retract_LD.

7. Type the number of millimeters that you want the tool center point (TCP) to approach or retract using Linear Distance.

   Note The default value is “direct” which means that the value is a specific number in millimeters.

   To use a value stored in a register, press F3, INDIRECT, and type the register number.

Corner Region Termination Type (CRy)

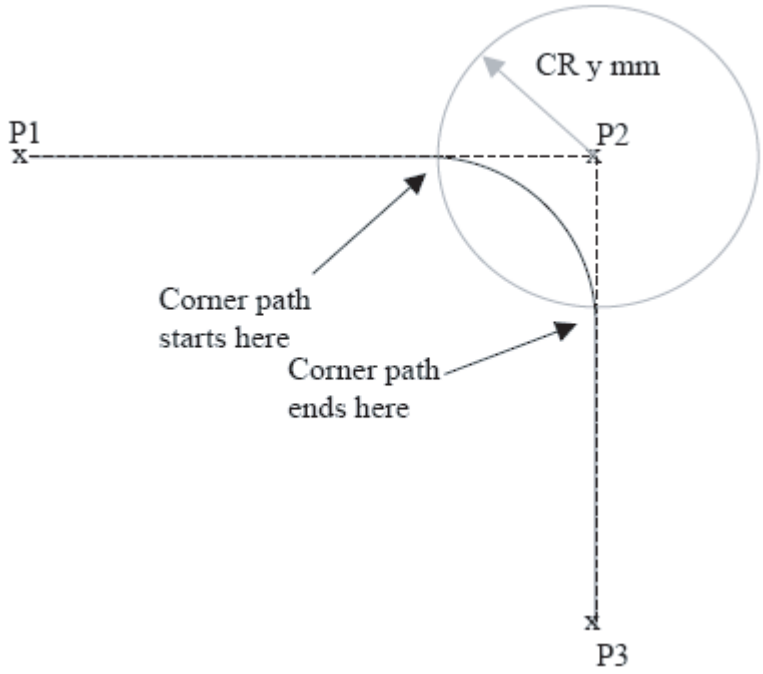
```
L P[2] 100 mm/sec CRy
```

CRy is an optional termination type that can be used to adjust the corner rounding for Cartesian motions. When you use the CR termination type, you must specify the corner region value, y, (in millimeters).

Corner Region value is the distance from the starting of a corner path to the taught position, as shown in the following figure. When CRy is specified, TCP path will maintain the corner path within the

specified region, meaning that the actual distance from the starting/ending of a corner path to the taught position is less than or equal to the specified corner region value (y).

Figure 6–41. Corner Path



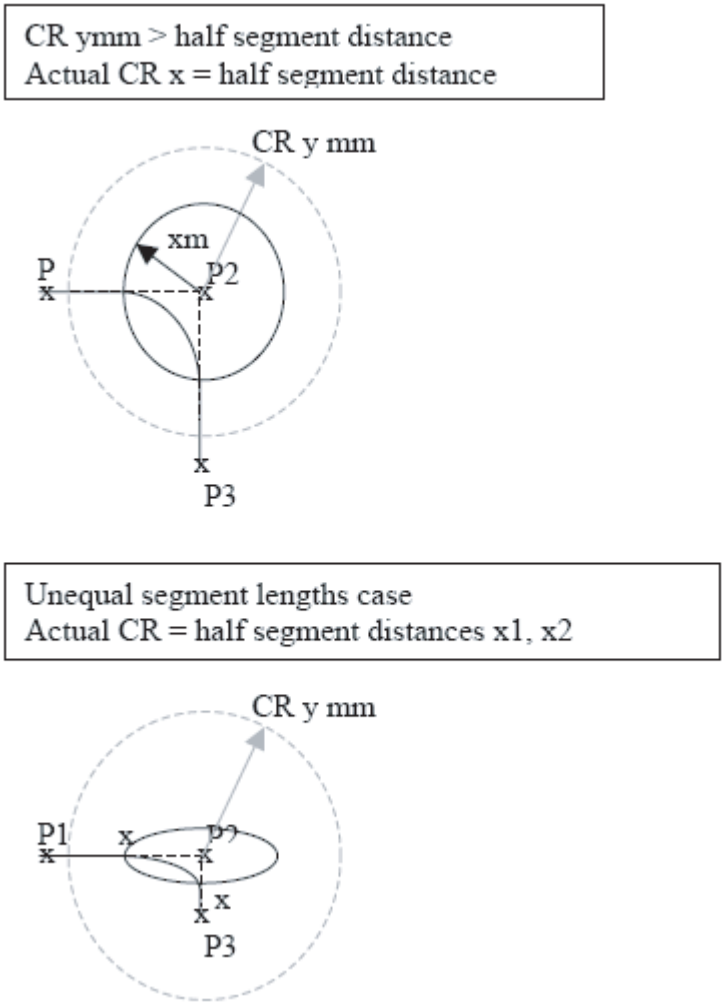
When you set corner region, use the following guidelines:

- Specify the corner region in millimeters
- Corner region value can range in value from 0 mm to 1000 mm
- The *smaller* the corner region value, the *closer* the robot will get to the taught position, and the *less* the corner rounding
- With a *larger* corner region value specified, the robot will not get as close to the position and the *more* corner rounding

CRy versus segment distance

If the specified corner region value, y, is greater than half the segment distance, then the actual value used is limited to half segment distance as shown below

Figure 6–42. Half Segment Length



Teaching Techniques

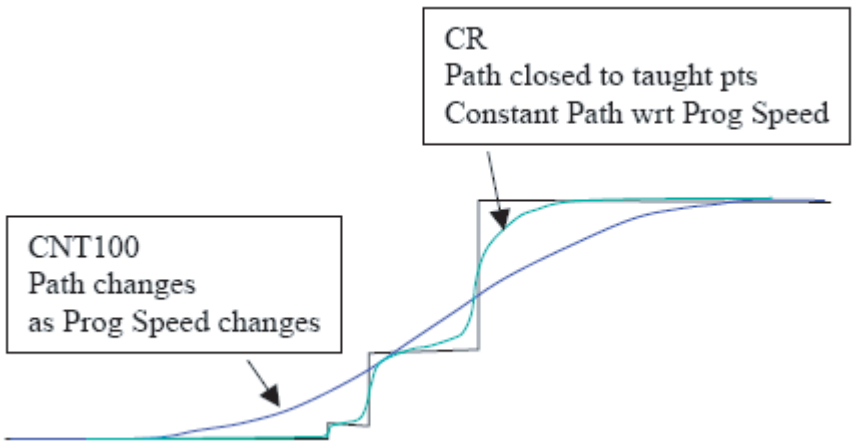
You must be careful about the half distance rule stated above. Keep in mind that because of the half distance rule, with large specified corner region value, corner path may still be close to the taught position when the segment distance is short. Use the following guidelines when you teach a path:

- Minimize the number of taught positions
- Reteach positions using the CR termination type to fit the path instead of adding positions

Constant Path with respect to Program Speed Changes

Program section with consecutive **CR**y – path could be maintained much better compared to other termination types even when program speed changes, as shown in the following figure.

Figure 6–43. Program Speed Changes



Compatibilities and Limitations

- **Motion Type**
  - — CRy supports Cartesian motion types such as LINEAR and CIRCULAR
  - — J motion type is **not** supported
- **Multi Group Motion**

  For multi group motions, motion will be synchronized. The corner path is generated as a result of the synchronization of applicable CRy for all involved groups;
- **CRy termination type** only supports articulated robots and does not support INDEPENDENT AXES or POSITIONERs
- **Motion Options**

  CRy supports the following motion options:
  - — Group motions
  - — RTCP
  - — Line Tracking

  CRy has no obvious geometric meaning (corner path would be naturally generated by motion blending) during the following transitions
  - — between COORD and non-COORD (if supported)
  - — between RTCP and non-RTCP

— between Tracking and non-tracking .

The CRy termination type may not work with TCP speed prediction function (TCPP). That is, with CRy termination type, TCPP results may not be correct.

If you specify both Linear Distance and CRy termination type, then Linear Distance has preference over CRy in determining a corner path, as shown in the following figures.

Figure 6–44. Corner Path determined by CRy if Linear Dist is satisfied

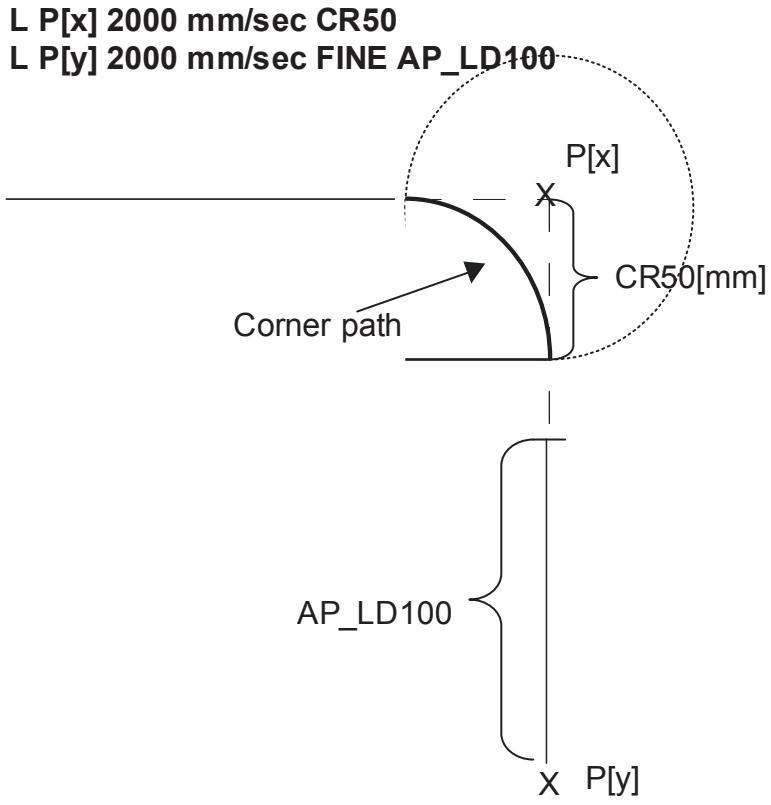
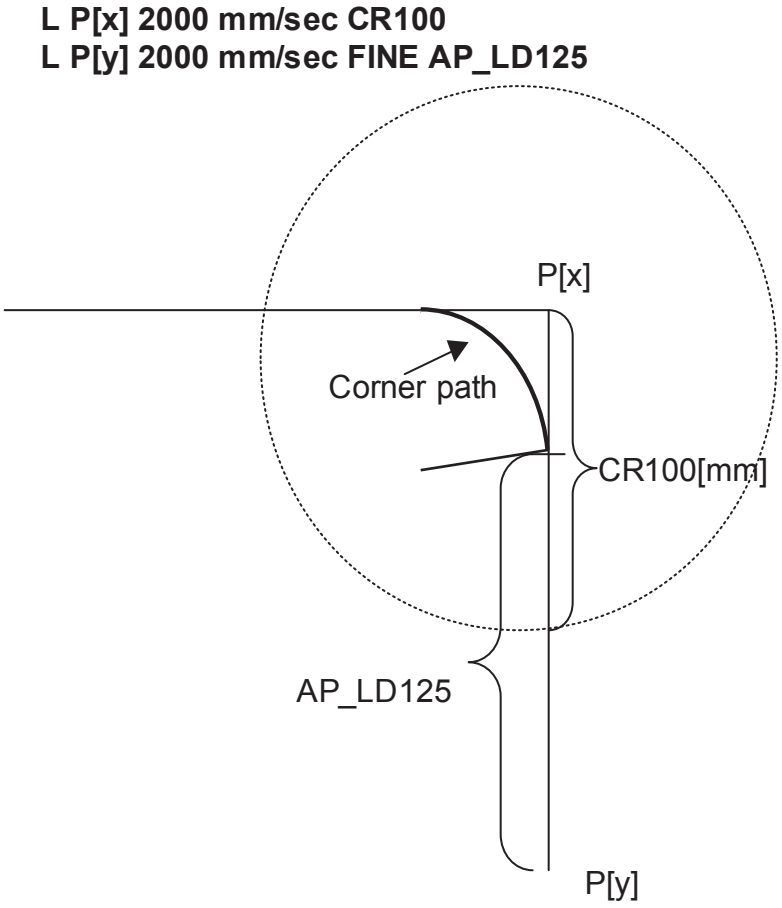


Figure 6–45. Corner path is determined by Linear Distance



Process Speed

```
PSPD xxx
```

Process speed is a motion option control feature that allows you to adjust robot speed to be faster or slower along a given path (if applicable, the path would be maintained the same regardless of xxx), where xxx is an integer you specify. The larger the value of xxx is, the faster the robot will move along the given path.

Process speed is useful for applications with continuous path motion that don't normally use maximum program speed; for example, sealing and waterjet cutting. Typically, the process controls the program speed: how fast the sealing gun can dispense, and how fast the waterjet can cut.

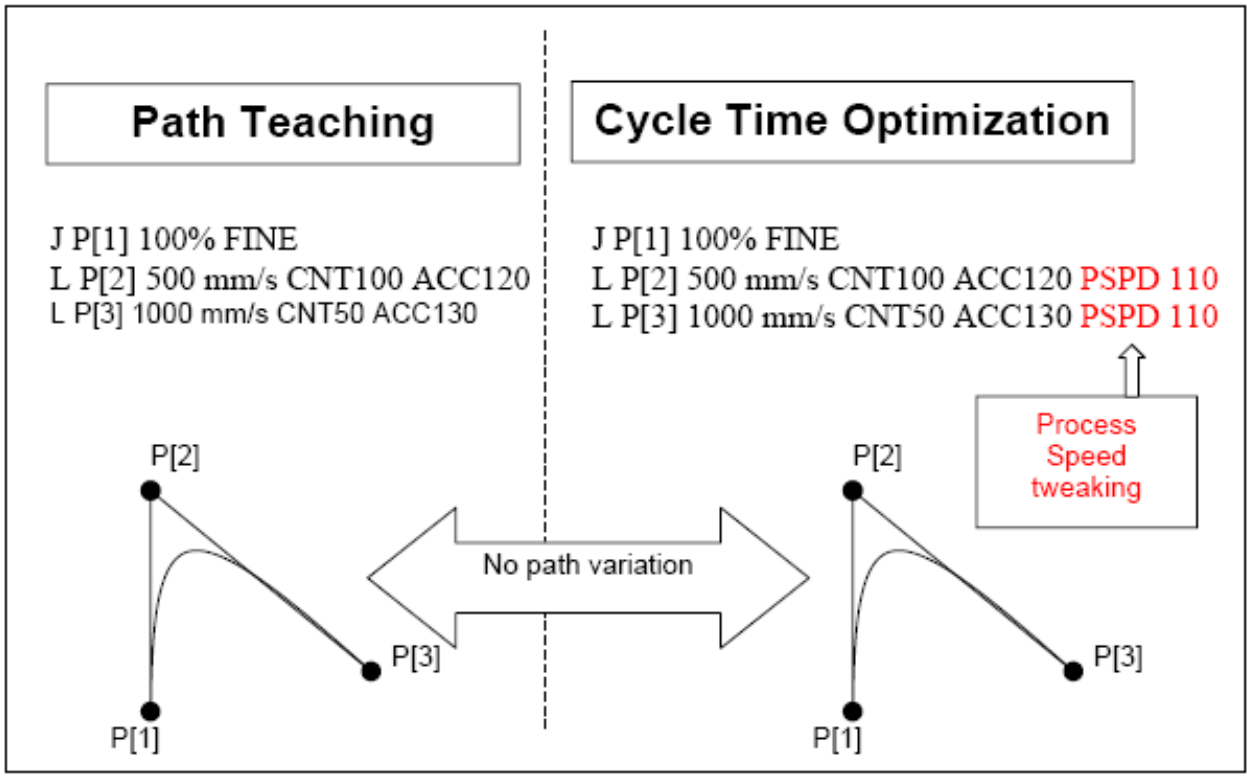
For these applications, teach the desired path using normal methods, tweaking taught position, speed, termtype and ACC.

After the path is taught, if you want to adjust its process speed from the nominal taught value, but do not want to change the path, you can use the Process Speed feature.

Add this motion option to the range of motion lines where adjustment is required.

- PSPD 100 is equivalent to the default cases without PSPD option.
- PSPD greater than 100 means faster process speed, while maintaining the same path.
- PSPD less than 100 means slower process speed, while maintaining the same path.
- You can still change other fields in the motion for further tweaking, but the same original rules apply; that is, the path will change. This allows you to adjust the path easily, even though PSPD is used.
- For PSPD greater than 100, the system internally limits the achievable (but higher) process speed, based on the jerk/acceleration margin available from the default case.

Be careful to use the PSPD option to reduce cycle time while maintaining the same path since the jerk/acceleration value will be higher. An example is palletizing, where additional factors such as vibration, duty cycle, reducer life, and so forth, affect cycle time.



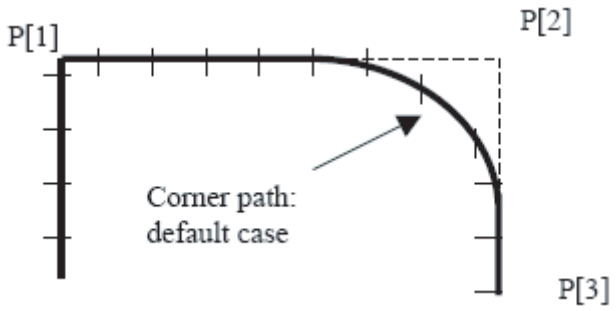
Caution

Process Speed can cause jerky motion if applied too aggressively. To avoid jerky motion, use a reduced speed.

PSPDxxx can be added to any selective motion line in a TP program, and is applicable to all motion types. For examples, see the following:

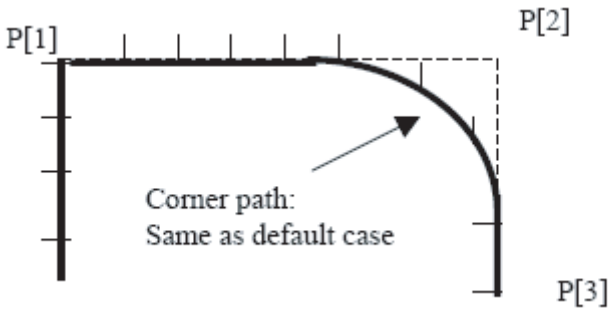
- **Case 1**: if xxx = 100, motion behavior is exactly the same as 100% speed override in the default case, as though there were no PSPD100
- **Case 2**: if xxx > 100, the speed will be faster than 100% speed override in the default case
- **Case 3**: if xxx < 100, the speed will be slower than 100% speed override in the default case
- Default case:

```
1  J  P[1] 50% FINE
2  L P[2] 500 mm/sec CNT100
3  L P[3] 500 mm/sec CNT0
```



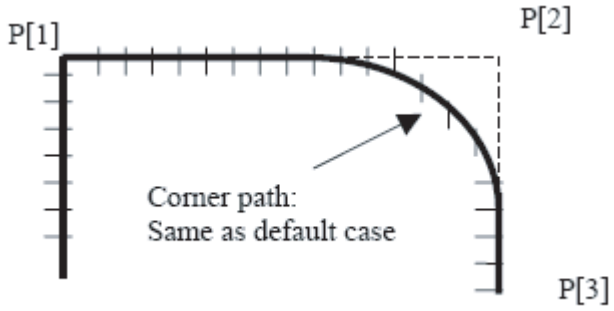
- Faster motion case (path is the same, cycle time is shorter):

```
1  J  P[1] 50% FINE PSPD110
2  L P[2] 500 mm/sec CNT100 PSPD110
3  L P[3] 500 mm/sec CNT0 PSPD110
```



- Slower motion case (path is the same, cycle time is longer):

```
1  J  P[1] 50% FINE PSPD50
2  L P[2] 500 mm/sec CNT100 PSPD50
3  L P[3] 500 mm/sec CNT0 PSPD50
```



Note The system will internally limit the speed override such that the resulting motion performance is within mechanical capabilities. As a result, a large value of xxx may not take effect in some cases and the actual speed override may be smaller than the specified value.

Limitations

- Under T1 mode, PSPDxxx (with xxx>100) will not take effect
- PSPDxxx (with xxx>100) might not take effect for the motion line that has max_speed as programmed speed in a TP program
- The PSPD option does NOT support TCP speed prediction function (TCPP) in the first release. That is, for the motions with PSPD option, TCPP might not result in correct results;
- The PSPD option does not support
  - — Continuous Turn
  - — Coordinated motions

— Robot Link

- With large PSPD value or very short segments, the actual corner path might deviate from the one without PSPD option.

Max Speed

```
L P[1]
max_speed

CNT100
```

In some applications, the desired speed is the maximum speed that the robot can deliver. For joint motion moves, the system delivers the maximum capability of the robot; that is, one of the axes reaches its maximum speed. For linear moves, the system delivers the speed that is specified in the teach pendant instruction. However, the maximum linear speed of 2000mm/sec imposes a limit on the capability of the motor to reach higher speeds. The robot can move faster than the speed specified in the motion instruction.

The max speed option allows you to specify a linear motion that will use the maximum speed capability of the robot. It improves cycle times in Load/Unload applications by speeding up long linear motions. When this option is loaded, the choice of max_speed will be displayed in the speed field of the teach pendant motion instruction for a linear motion.

The max_speed option affects **only** the motions for which the speed is specified as max_speed.

Note When you load this option, the itp_time will be set to at least 12ms.

- If you change the motion type from Linear to Joint, the speed field will change to 100%.
- When the speed field changes from max_speed to another choice, the speed value will return to the original speed value..



Warning

When you specify max_speed, the robot will run at high speed. Be sure any loose parts are firmly attached and that the workpiece is secured. Otherwise, you could injure personnel or damage equipment.

Limitations

- If unsupported options are used, max_speed will be disabled automatically. No warning or error message will be displayed. This option does not support the following:

— Any tracking option, such as line tracking, TAST, Mig-Eye, Coordinated motion, and so forth. When these options are used, the max speed option will default to 2000 mm/sec.

— Multiple group motion

— RTCP function

- If you run a program with an override speed different than 100%, the system will drive the robot such that one of its axes will reach the override value of its maximum joint speed.
- The local condition trigger time might have some variation.
- If the path becomes too aggressive, you might need to use ACC to smooth it.
- If you are using Dry Run, max speed will be disabled and the speed specified in dry run will be used.
- If you are using Org path resume, max speed will be disabled for the motion line that is resumed.
- If T1 is selected, the T1 speed will be used.
- In single step mode (FWD/BWD) max speed will be disabled and the maximum speed value will be used.
- Max speed will be disabled automatically for a circular motion.
- The max speed option will still apply when the Miscellaneous teach pendant instruction LINEAR_MAX_SPEED is used.

  The robot will try to attain the maximum speed capability of at least one of its axes. It determines the maximum speed for the current move by comparing the teach pendant instruction LINEAR_MAX_SPEED with the maximum linear speed of 2000 mm/sec. The ratio of these two speeds is the percentage of the maximum axis speed that the axis will reach.

  For example: the maximum linear speed is 2000 mm/sec.

```
1. LINEAR_MAX_SPEED = 1200
2. L P[1] max_speed CNT1000
```

  The ratio of 1200 to 2000 is 60%. The system will drive the robot such that one of its axes will reach 60% of its maximum joint speed for line 2 of the program above.

- Max speed does not work with TCP speed prediction function (TCPP). That is, with Max speed, TCPP results may not be accurate.
- Max speed does not support the following functions (the system will automatically disable the Max speed feature):

  — Line Tracking

  — RTCP function

  — Multi-group motion

6.4.10.19 Torch Angle

Overview

The Torch Angle option provides users with multiple methods to display the welding torch Work and Travel Angles. This eliminates the need for manual Torch Angle measurement, assists in the audit of weld torch angles, and helps optimize welding quality.

There are two primary methods of obtaining the torch angles:

- As a TP program executes, the angles are continuously calculated based on the executing motions and frame definitions. The resultant data is available in system variables and can be displayed in the editor or in a graphical status screen
- A report can be generated for any of the TP programs from the robot's web page without running the TP program.

Torch Angle system can be used by several applications including:

- Gas Metal Arc Welding and Gas Tungsten Arc Welding
- Plasma Cutting
- HandlingTool Laser Cut
- DispenseTool Dispense Gun

Torch Angle supports multiple arm group motion and is compatible with many robot options including:

- Touch Sensing
- Thru Arc Seam Tracking
- RPM and Multi-Pass
- Coordinated motion
- Constant Path motion

The following sections will address these topics:

- Torch Angle Definitions
- Teaching a Reference Position
- Programming Considerations and Limitations
- Torch Angle Web Page Display
- Torch Angle Status Display
- Torch Angle System Variables

- Torch Angle Graphical Status Display

Torch Angle Definitions

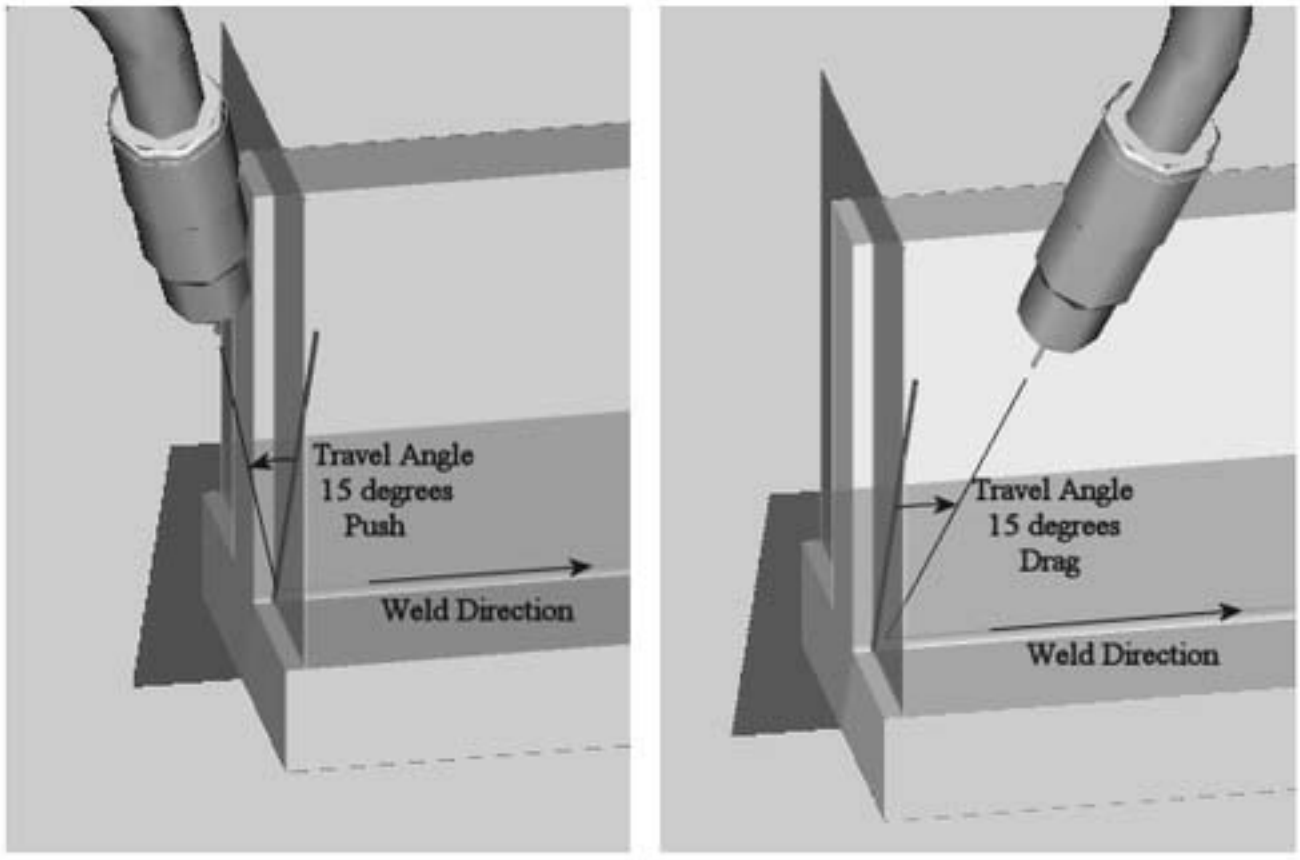
The Torch Angle is specified by the Work Angle and the Travel Angle.

Travel Angle

Travel Angle Definition: The angle less than 90 degrees between the torch axis and a line perpendicular to the weld direction, in the plane determined by the torch axis and the weld direction

In the two examples shown below the welding torch is approaching the Arc Start position of a weld in a T-joint. In both examples the Travel Angle is 15 degrees. In the left figure, the torch is pointing in the direction of the weld, so this is considered a push angle. In the right figure, the torch is pointing in the direction opposite to the weld progression, so this is a drag angle.

Figure 6–46. Travel Angle Definitions (Push and Drag)



For curved surfaces or pipe welds, the weld direction will be the tangent line to the weld path at the weld point.

Work Angle

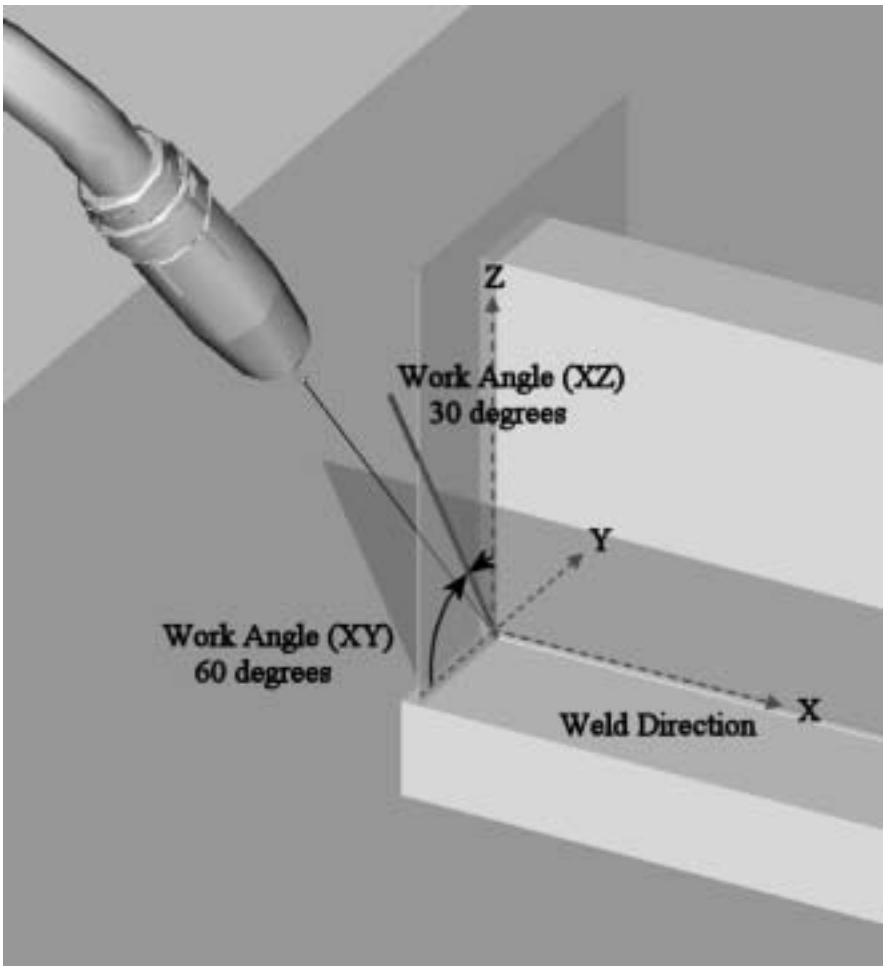
There are two common ways to measure the work angle. One references the vertical XZ plane and the other references the horizontal XY plane. The X axis is defined as the vector along the weld axis. The Z direction is defined by a taught TA_REF reference position or by the Z axis of uframe. The Y axis is perpendicular to the X and Z axes.

Work Angle (XZ) Definition: The angle less than 90 degrees between the torch axis and a plane determined by the weld direction and a surface normal vector perpendicular to the workpiece. The surface normal at a point on a non-flat surface will be a vector perpendicular to the tangent plane to that surface at that point.

Work Angle (XY) Definition: 90 degrees minus the Work Angle (XZ).

This figure depicts the welding torch approaching the Arc Start position of a weld in a T-joint. The travel angle is 15 degrees push. The Work Angle (XZ) is 30 degrees. The Work Angle (XY) is 60 degrees (90-30).

Figure 6–47. Work Angle Definitions (XZ and XY)



A reference position is not needed in this example. It is sufficient to use uframe.

The user can choose between the XZ and the XY definitions in the Torch Angle Status Display using the "Work Ref" function key. This display is described in a later section.

Teaching a Reference Position

When the workpiece is not suitably aligned with uframe it is useful to teach a reference position and specify it in a TP program. This allows the work angle to be calculated relative to the workpiece

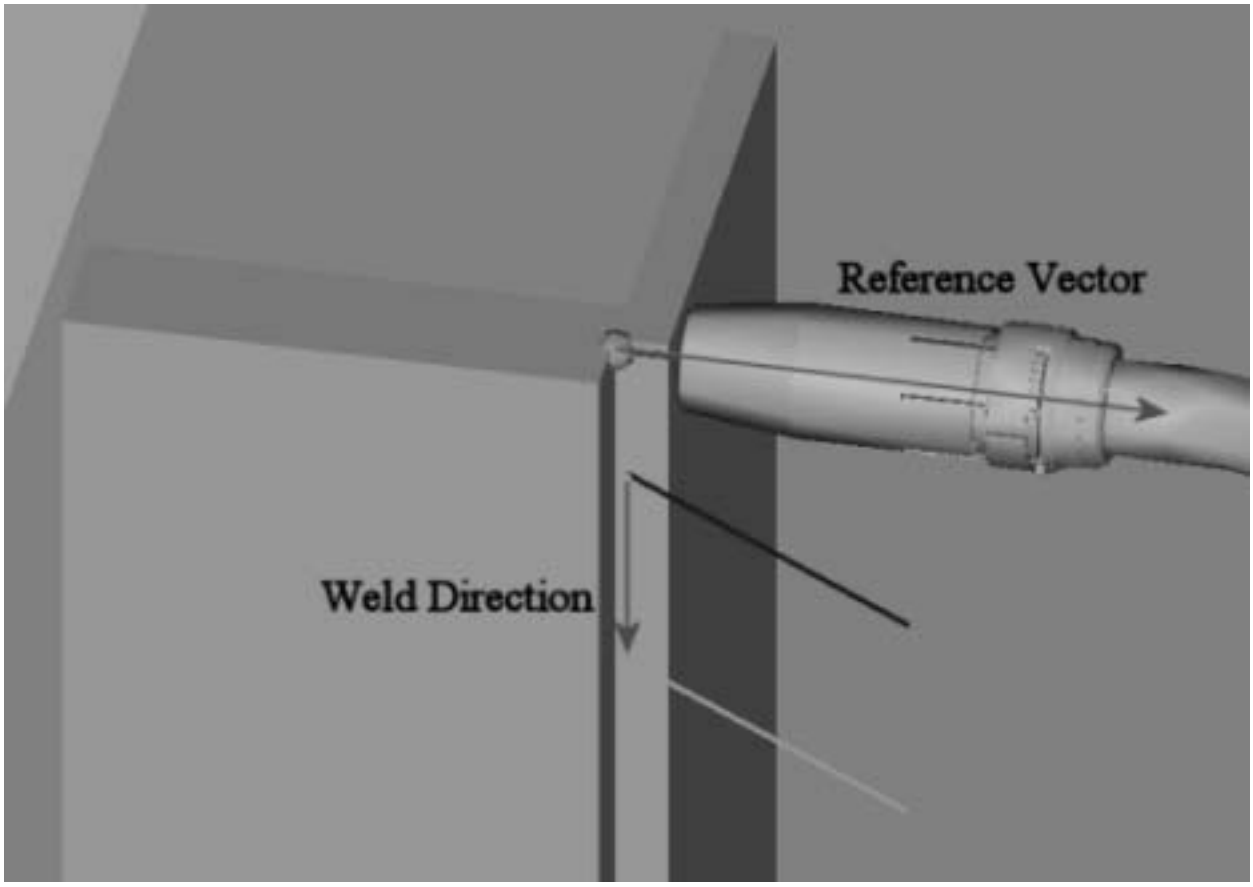
The reference position can be recorded in a TP program with the TA_REF motion modifier or it can be recorded in a position register and referenced in the TP program with the TA_REF PR[] motion modifier.

The figure below depicts a vertical-down weld in an outside corner joint. The work angle is approximately 45 degrees and the travel angle is approximately 15 degrees push. To assure the

work angle is calculated relative to the workpiece, a reference position is recorded normal to the workpiece as shown.

- The reference position is used to calculate the Work Angle; it is not used to calculate the Travel Angle.
- TA_REF or TA_REF PR[] can be attached to each weld point to support various weld joints.
- The tool frame must be taught such that the tool frame Z+ corresponds to the torch direction.

Figure 6–48. Teaching a Reference Position



Procedure 6-3 Recording the reference position with TA_REF

1. Move the cursor to the end of the motion line.
2. Press F4 [CHOICE] and select the motion modifier TA_REF. The prompt "Align torch with ref axis & record." will be displayed as in Figure 6–49
3. Move the robot to an area with enough space to permit adjusting the torch orientation.
4. Align the torch such that it is perpendicular to the workpiece or so it is aligned with the reference axis, i.e. the torch is at zero degree work angle.

5. Press Shift-F3 [RECORD] to record Torch Angle Reference position.

Figure 6–49. Torch Angle TA_REF Instruction

```
        P[Travel dg G1]              1/2
   1:J @P[1:          ] 100% FINE
    :  TA_REF
[End]









Align torch with ref axis & record.
        MOVE_TO RECORD [CHOICE]
```

Programming Considerations and Limitations

When using the Torch Angle option please consider the following items:

- The tool frame must be taught such that tool frame Z+ corresponds to the torch direction.
- The “TA_REF” motion modifier can be used in a TP program to record the reference positions of up to two robot arms. “TA_REF PR[]” is required if there are more than two robot arms in the program.
- When coordinated motion is specified, “TA_REF PR[]” is required to record the reference positions of both the leader and the follower.
- A position register does not include uframe data. “TA_REF PR[]” uses the currently active uframe to calculate the reference position during program execution.
- The Torch Angle reference position is set to uframe when a program is aborted.
- A run-time motion alarm “Uninitialized TA_REF ref frm” will be posted for TP motion instructions with TA_REF or TA_REF PR[] if the reference position is not recorded.
- If there isn’t a torch angle motion modifier (TA_REF or TA_REF PR[]) on the current welding motion, the previous reference position defined by TA_REF or TA_REF PR[] in the TP program will be used. If neither TA_REF or TA_REF PR[] have been executed in the program, the User frame approach vector will be used to define the reference position.
- A sub-program inherits the reference position from the main program.
- A reference position set inside a sub-program remains valid when execution returns to the main program.

Torch Angle Definition

Procedure 6-4 Recording the reference position with TA_REF PR[ ]

1. Move the cursor to the end of the motion line.
2. Press F4 [CHOICE] and select the motion modifier TA_REF PR[]. The prompt "Enter PR number." will be displayed as shown in Figure 6–50
3. Enter the position register number.
4. Move the robot to an area that has enough space to permit adjusting the torch orientation.
5. Align the torch such that it is perpendicular to the workpiece or so it is aligned with the reference axis, i.e. the torch is at zero degree work angle.
6. Go to DATA Position Reg screen, and move the cursor to specified PR number.
7. Press Shift-F3 [RECORD] to record Torch Angle Reference position. After the reference position has been recorded, Shift-F2 [MOVE_TO] can be used to verify the recorded position.

Figure 6–50. Torch Angle TA_REF PR[ ] Instruction

```
        P[Travel dg G1]              1/2
   1:J @P[1:          ] 100% FINE
    :   TA_REF PR[...]
[End]




Enter PR number.
        DIRECT INDIRECT[CHOICE]
```

For pipe weld, the weld direction will be the tangent line to the weld path at the weld point.

Torch Angle Web Page Display

A new link "TP Torch Angle files available on MD:" is available on the controller's main web page when Torch Angle option is loaded.

Figure 6–51. Web Page Main Display

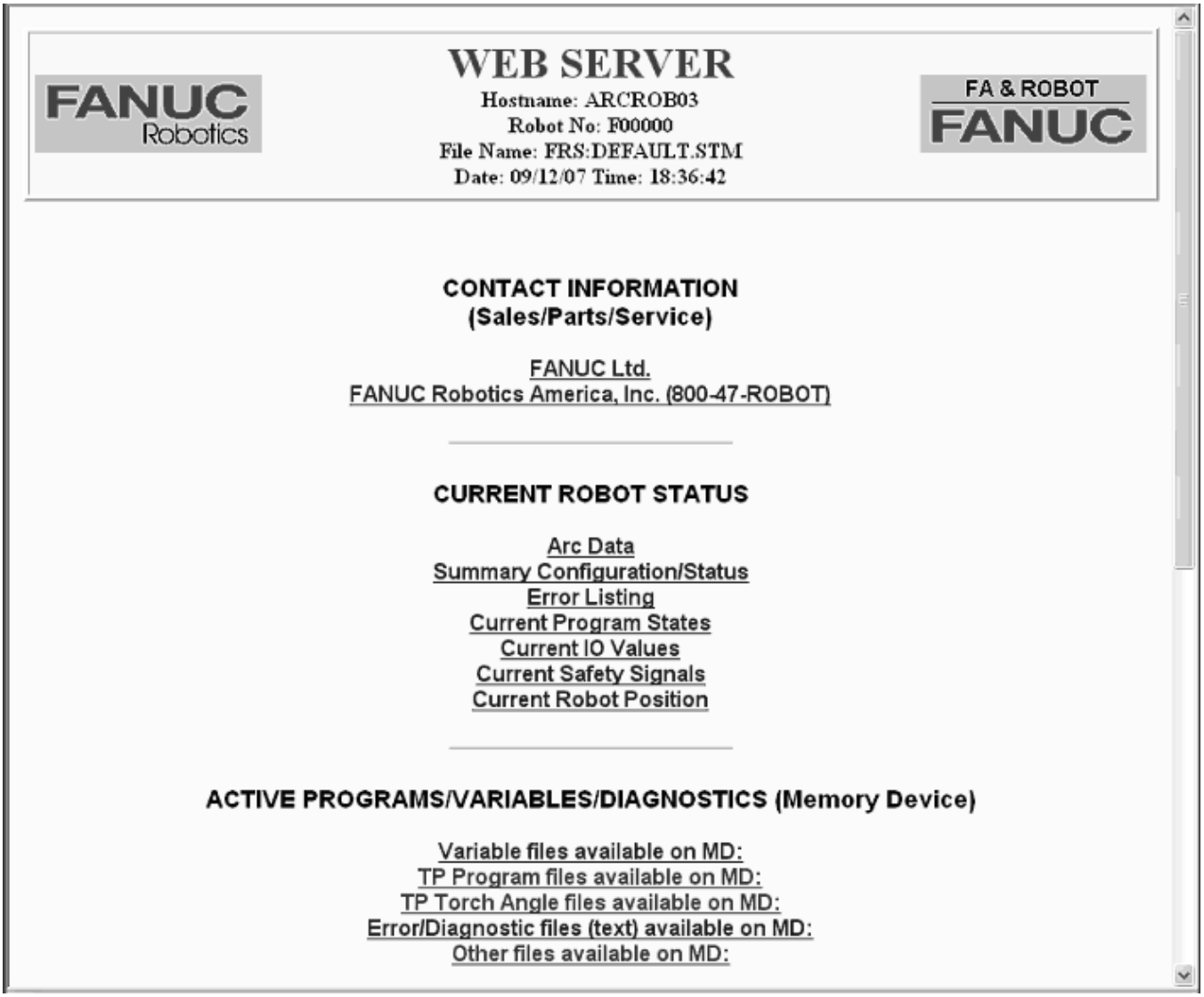


Figure 6–52. Web Page Torch Angle Files



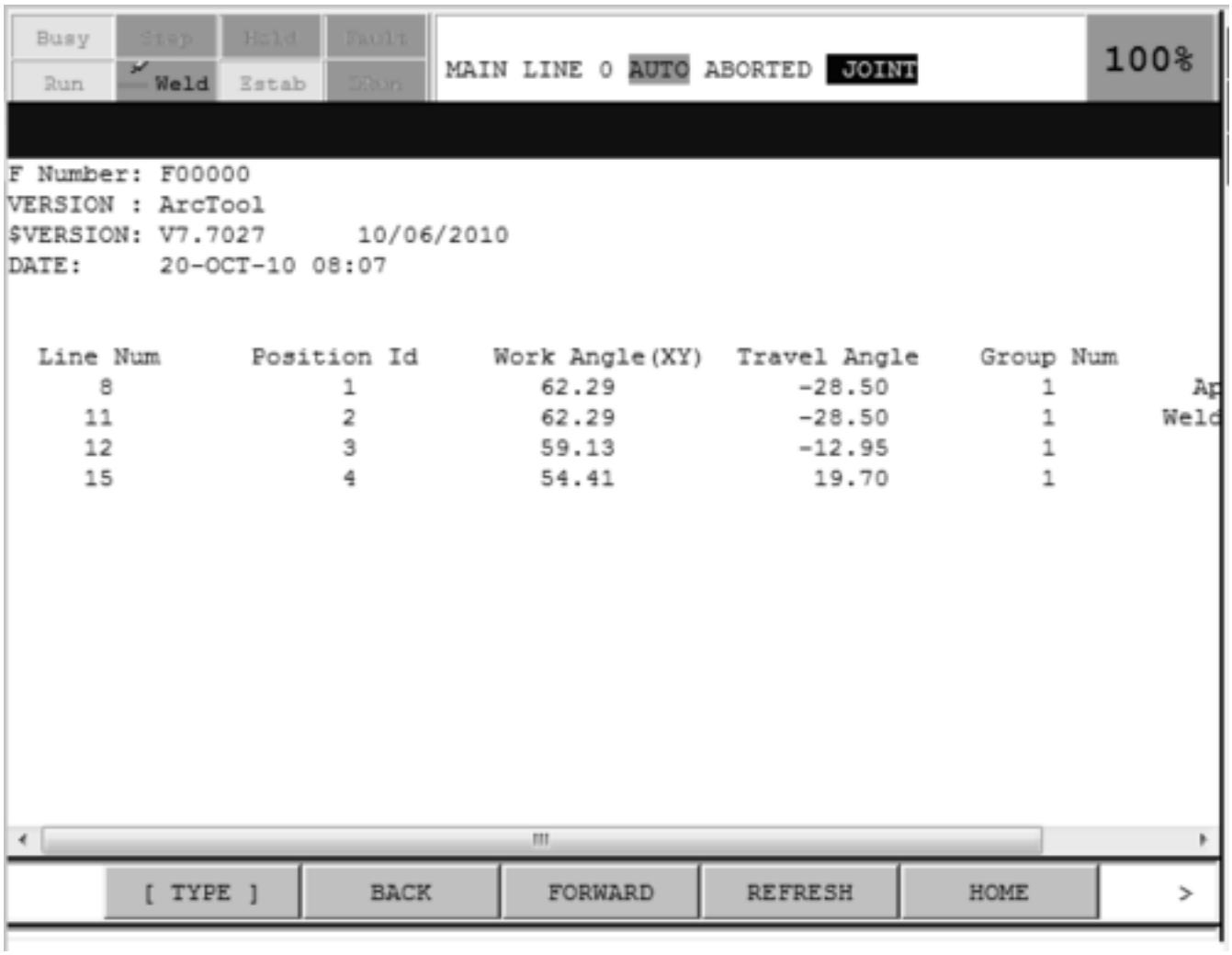
TP program name	Torch Angle ASCII file	Comment
ABORTIT.TP	ABORTIT.TA	*ABORT PRODUCTION*
DSP_WEBP.TP	DSP_WEBP.TA	*Display web page*
GETDATA.TP	GETDATA.TA	*Get PC Data*
LISTMENU.TP	LISTMENU.TA	*LIST MENU MACRO*
MENUTEST.TP	MENUTEST.TA	*Example menus*
TANG1.TP	TANG1.TA	
TANG124.TP	TANG124.TA	
TANG124T.TP	TANG124T.TA	
TANG124TW.TP	TANG124TW.TA	
TANG124_C.TP	TANG124_C.TA	
TANG124_CT.TP	TANG124_CT.TA	
TANG124_CTW.TP	TANG124_CTW.TA	
TANG1A.TP	TANG1A.TA	
TASTMAST.TP	TASTMAST.TA	*TAST Mastering*
TEST_TIG_1.TP	TEST_TIG_1.TA	
TIG2.TP	TIG2.TA	
TIG_1.TP	TIG_1.TA	

A Torch Angle ASCII report can be generated for each Teach Pendant program on the controller. A report will be displayed on the web page if user clicks on one of the file names listed in the second column of the table.

- The ASCII file is tab delimited, and can be saved and exported to Excel.
- Torch angle report file consists of the following information:
  - — Line Num
  - — Position Id
  - — Work Angle
  - — Travel Angle
  - — Group Num
  - — Position Comment

The right-most column contains the Position Comment. This column can be viewed by scrolling the display to the right.

Figure 6–53. Web Page Torch Angle Report File



The Torch Angle Web Page displays static work angles and travel angles of taught positions. The values can be different than the run time display due to motion blending

The robot web pages can be accessed from a remote computer using a standard browser if the Web Server is installed on the robot. The Web Server is part of the Internet Connectivity/Customization Option, R558.

When viewing a torch angle report remotely you can use the browser to “select all” and “copy” the report contents. The data can then be saved from your clip board into a file. The file can be imported into a spreadsheet application as shown in Figure 6–54

Figure 6–54. Torch Angle Report in a Spreadsheet Application

	A	B	C	D	E	F
1	F Number: F00000					
2	VERSION : ArcTool					
3	$VERSION: V7.7027 10/06/2010					
4	DATE: 20-OCT-10 08:04					
5						
6						
7	Line Num	Position Id	Work Angle(XY)	Travel Angle	Group Num	Pos Comment
8	8	1	62.29	-28.5	1	Approach Pos
9	11	2	62.29	-28.5	1	Weld Segment 1
10	12	3	59.13	-12.95	1	Depart
11	15	4	54.41	19.7	1	

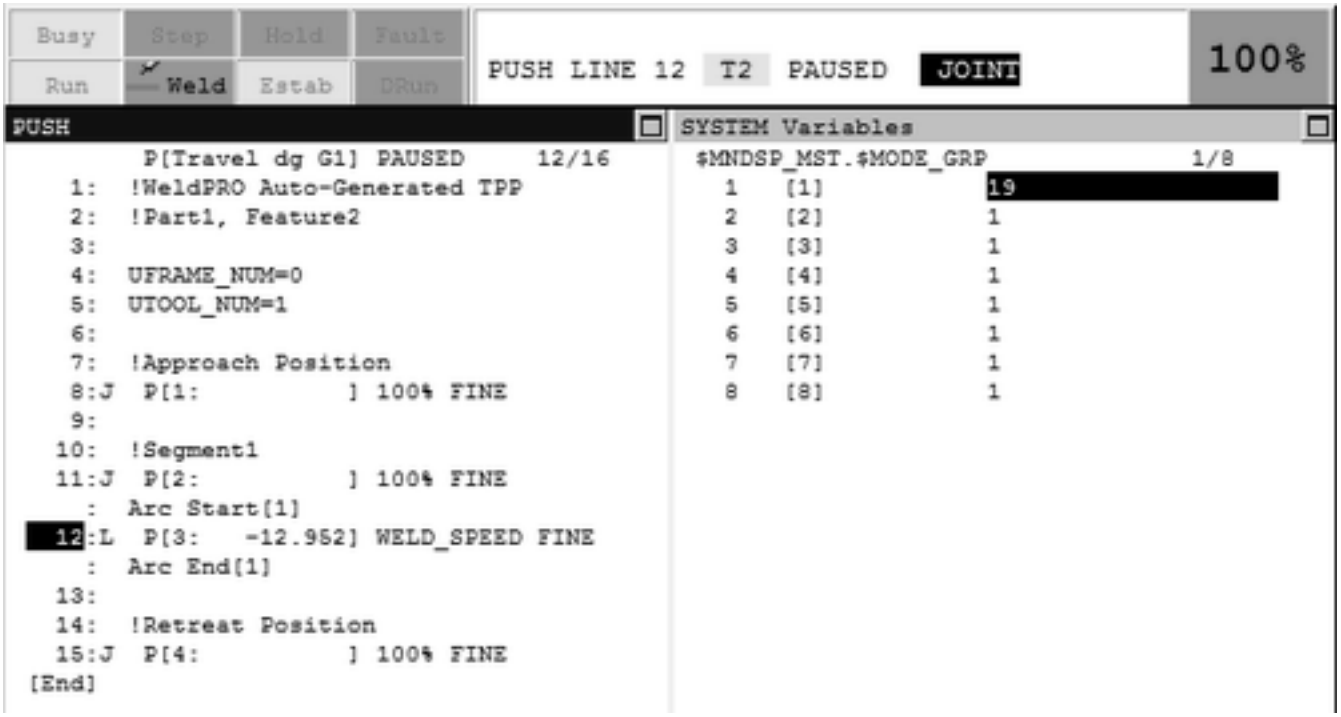
Motion Status Display in the TP Editor

The Motion Status display feature dynamically displays the current value of a variety of motion related items directly within the motion line. Refer to Section for more information on this feature.

The Motion Status display supports display mode 19 for the travel angle and mode 20 for the work angle. These two modes only display data on the currently active motion line.

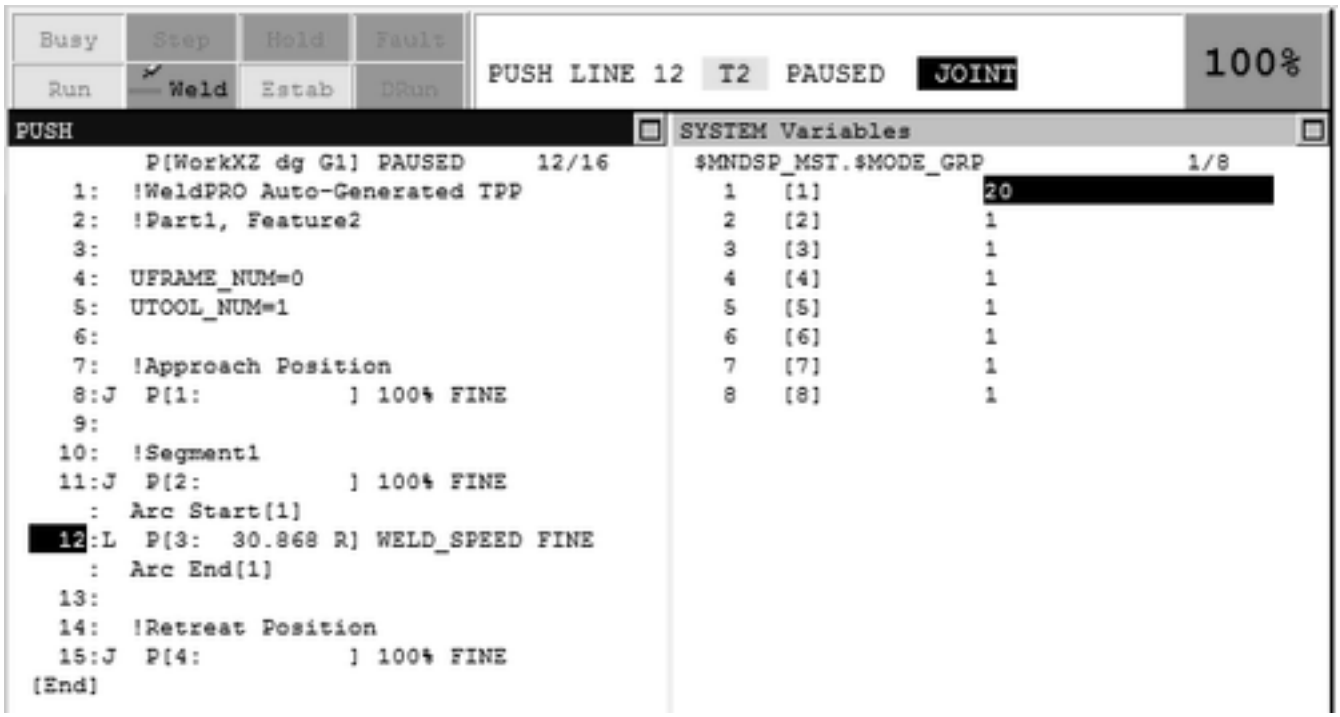
- Set $mndsp_mst.$disp_enable = TRUE to enable the motion status display.
- Set $mndsp_mst.$mode_grp[1] = 19 to display group 1 Travel Angle.
- Set $mndsp_mst.$mode_grp[1] = 20 to display group 1 Work Angle.

Figure 6–55. Travel Angle Progress Display



The header above the program is "P[Travel dg G1]". This indicates the motion status display is set to mode 19 for the Travel Angle, the angle is in degrees, and the angle is for group 1. Group 1 is the selected jog group. The Travel Angle of 12.952 is shown on line 12 of the program. The negative sign indicates this is a push angle.

Figure 6–56. Work Angle Progress Display



The header above the program is “P[WorkXZ dg G1]”. This indicates the motion status display is set to mode 20 for the Work Angle. The “XZ” indicates the selected reference plane is the XZ plane. The “dg” indicates the angle is in degrees.

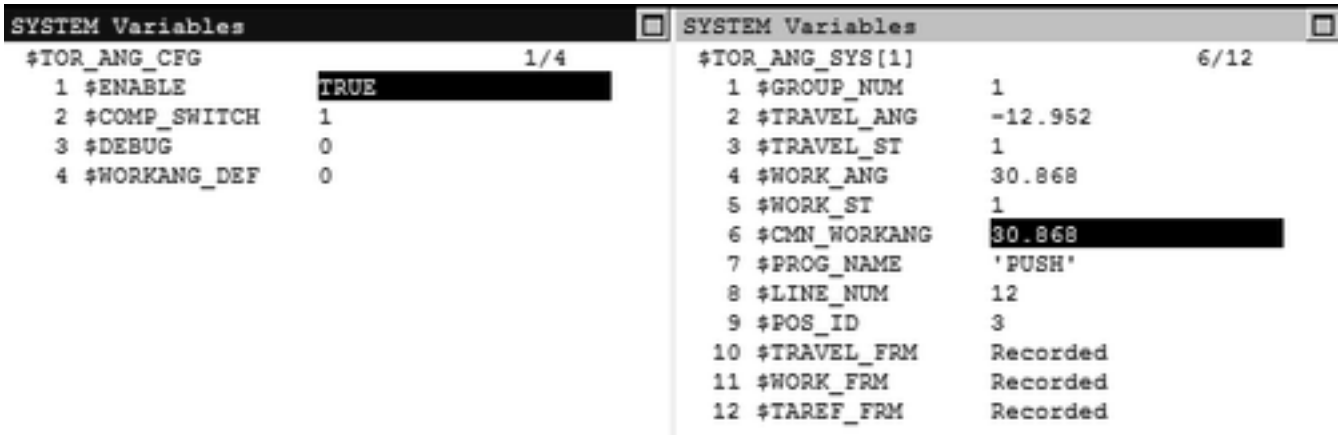
The program is paused on line 12. The work angle at this position is 30.868 degrees. The letter R appended after the angle indicates the torch is to the “Right” of vertical (when viewed from behind the torch in the direction of the weld.) A letter L would be appended after the angle if the torch were to the “Left” of vertical.

Torch Angle System Variables

When the Torch Angle option is loaded and enabled and a Teach Pendant program is executing the Torch Angles are calculated and stored in system variables ($tor_ang_sys[].$travel_ang and $tor_ang_sys[].$work_ang). These variables are used as inputs to update a graphical display on the Torch Angle Status screen.

Note the value of $work_ang can vary from -180 to +180 degrees. The value of $cmn_workang will only vary from 0 to 90 degrees when the WorkXY reference plane is selected.

Figure 6–57. Torch Angle System Variables



Torch Angle Status Menu

The Torch Angle Status menu displays the current torch angles for the selected jog group if that group is a robot. The menu contains a graphical and a numeric indication of the work and travel angles. It indicates if the Work Angle is to the Left or Right of vertical axis and if the Travel Angle is Push or Drag.

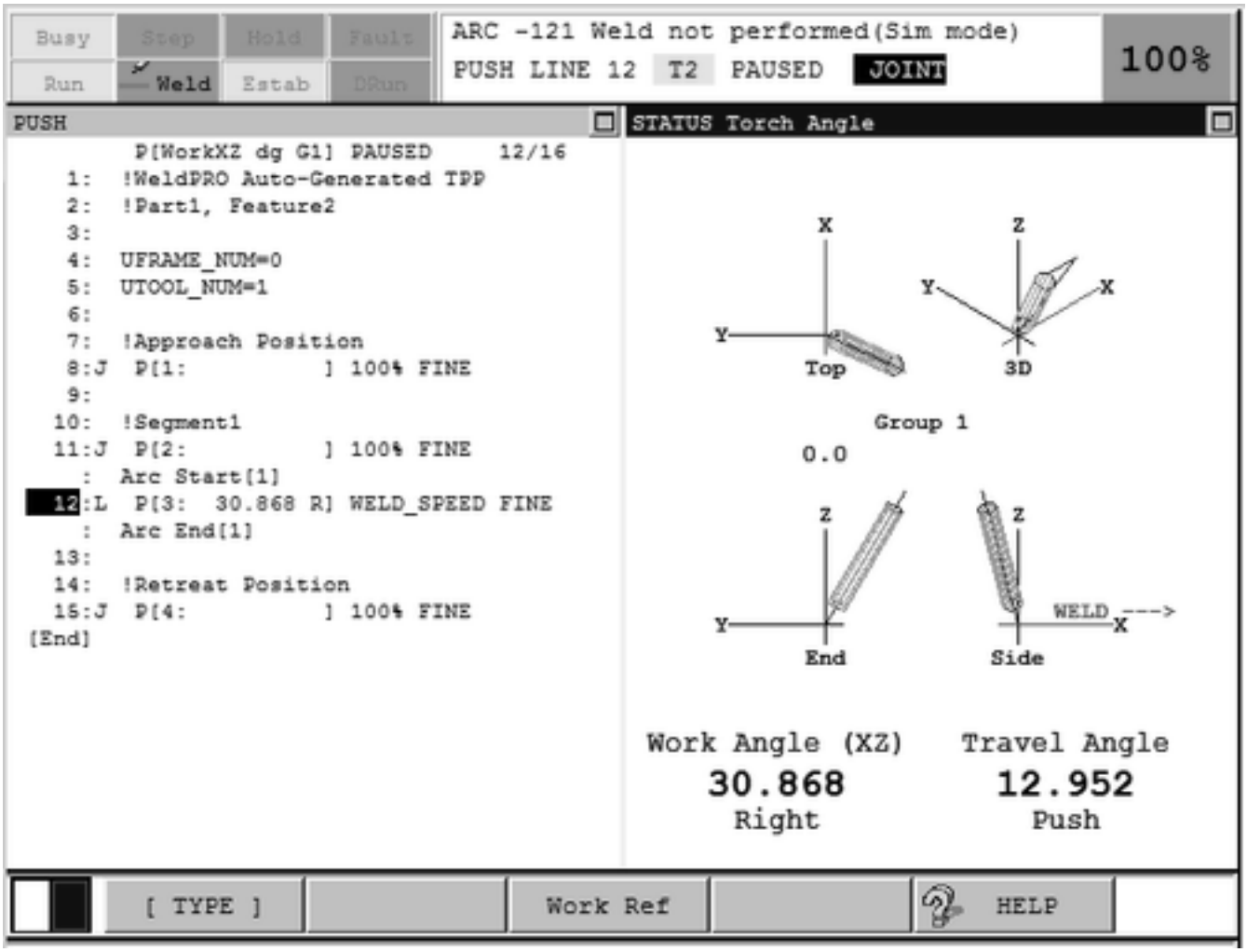
When the program is aborted the torch angles are not available for display. This is indicated by the torch being aligned with the Z axis and “(inactive)” displayed below the two angles.

To display the Torch Angle Status menu:

1. Press MENUS.
2. Select STATUS.
3. Press F1, [TYPE].
4. Select Torch Angle.

For Torch Angle graphical display, select “Torch Angle” from STATUS item on main Menu.

Figure 6–58. Torch Angle Graphical Display



Use the F3 function key “Work Ref” to change between the XZ and XY reference planes.

6.4.11 Correspondence Between Teach Pendant Program Motion and KAREL Program Motion

The motion control functions that are supported both in the $GROUP system variables and in the teach pendant motion instruction use the value that is specified in the teach pendant motion instruction. Refer to Table 6–11 for the correspondence between the $GROUP system variables used for KAREL program motion and the teach pendant motion instruction.

Table 6–11. Correspondence between $GROUP System Variables and the Teach Pendant Motion Instruction

System Variable	Teach Pendant Motion Instruction Component
$GROUP.$motype	Motion type
$GROUP.$speed	Speed - mm/sec, cm/min, inch/min
$GROUP.$rotspeed	Speed - deg/sec
$GROUP.$seg_time	Speed - sec
$GROUP.$termtype	Termination type
$GROUP.$orient_type	Wrist joint motion option
$GROUP.$accel_ovrd	Acceleration override (ACC) motion option
$GROUP.$ext_indep	Simultaneous/Independent EV motion option
$GROUP.$ext_speed	Simultaneous/Independent EV motion option
$GROUP.$cnt_shortmo	PTH motion option

The single value of the speed field in the teach pendant motion instruction can take on the function of three system variables:

- If **translational speed (mm/sec, cm/min,inch/min)** is specified, then the rotational speed ($GROUP.$rotspeed) is set to $PARAM_GROUP.$rotspeedlim. The resulting motion is limited first by the command translational speed and second by the rotational speed limit.
- If **rotational speed (deg/sec)** is specified, then the translational speed ($GROUP.$speed) is set to $PARAM_GROUP.$speedlim. The resulting motion is limited first by the command rotational speed and second by the translational speed limit.
- If **time-based motion (sec)** is specified, then the translational speed limit uses $PARAM_GROUP.$speedlim (or $PARAM_GROUP.$jntvellim for joint motion) and $PARAM_GROUP.$rotspeedlim as speed limits. This is similar to how KAREL programs handle time-based motion.

Refer to the *Software Reference Manual* and the *KAREL Reference Manual* for more detailed information on system variables and KAREL program motion.

6.5 REGISTER INSTRUCTIONS

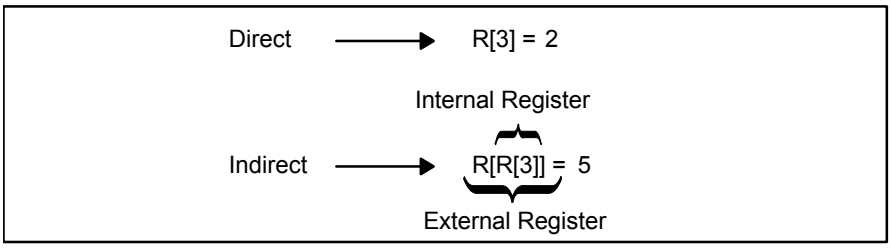
A*register* stores one number. Up to 999 registers are available for all the programs in the controller combined. The default number of registers is 32. Registers are identified by numbers. You can increase the number of registers during a controlled start. Refer to the " System Operations" Appendix in the Setup and Operations Manual for information on performing a Controlled start. Register instructions manipulate register data arithmetically.

Register Addressing

Many instructions employ*direct* or *indirect addressing* techniques. When direct addressing is used, the actual value is entered into the instruction. For example, if the register instruction R[2]= 5 is used, the current contents of register 2 is replaced with the value 5.

When indirect addressing is used, the instruction contains a register within a register. This indicates that the actual value of the internal register becomes the register number of the external register. See Figure 6–59 .

Figure 6–59. Direct and Indirect Addressing Example



In Figure 6–59 , the first instruction illustrates direct addressing. This instruction causes the current contents of register 3 to be replaced with the value 2.

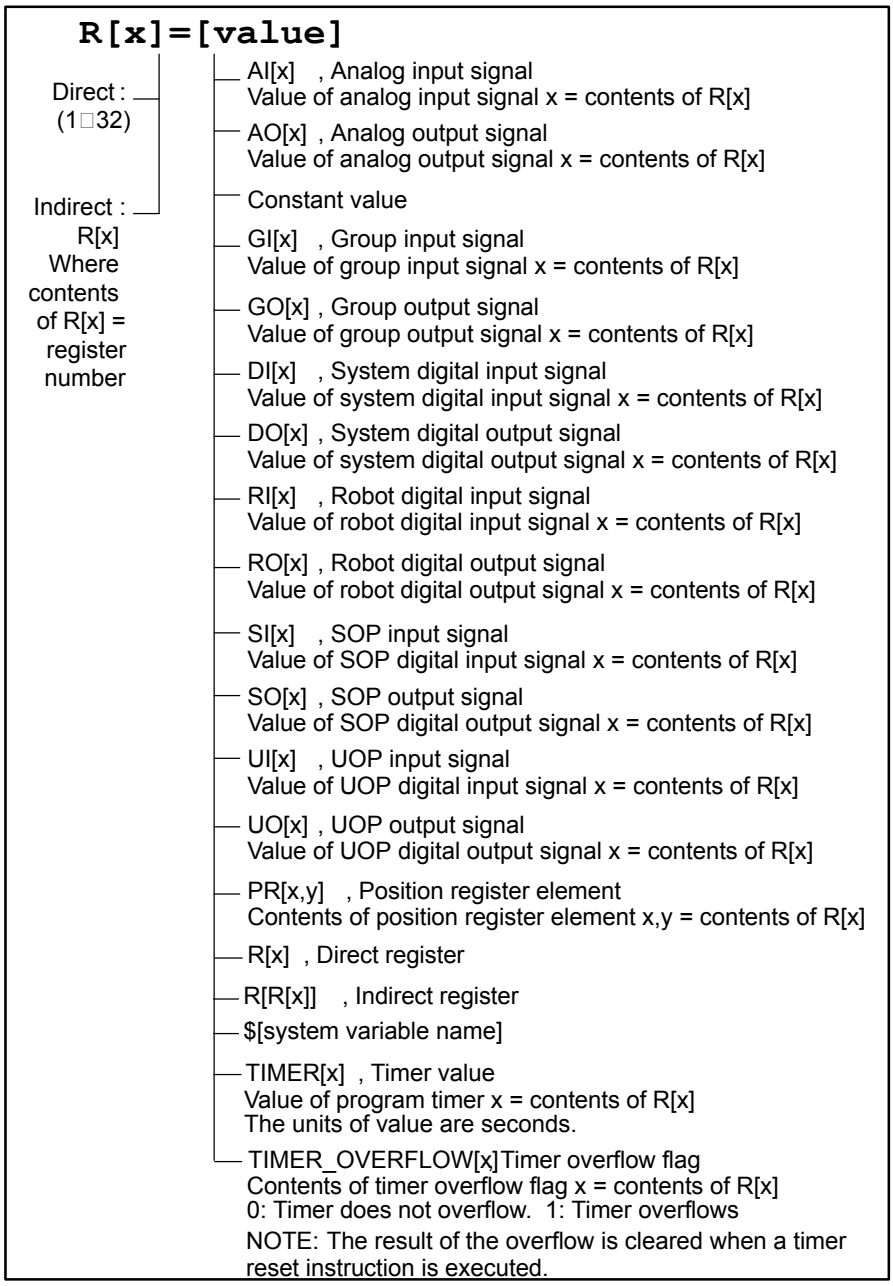
The second instruction in Figure 6–59 illustrates indirect addressing. In this instruction, R[3] is the internal register and R[R[3]] is the external register. Since in the previous instruction the value of the internal register R[3] is 2, the external register number becomes R[R[3]=2] or R[2]. Therefore, the result of the second instruction is that the contents of the external register, R[2], is to be replaced with the value 5.

R[x] = [value]

The R[x] = [value] instruction stores a value in a register. See Figure 6–60 .

Figure 6–60. R[x] = [value]

```
R[...] = ...
```



R[x]=[value][operator][value]

The R[x] = [value] [operator] [value] instructions store the result of an arithmetic operation in a register. The arithmetic operations are

- Addition
- Subtraction
- Multiplication

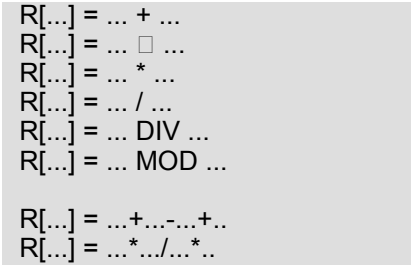
- Division
- Whole number division (DIV)
- Remainder division (MOD)

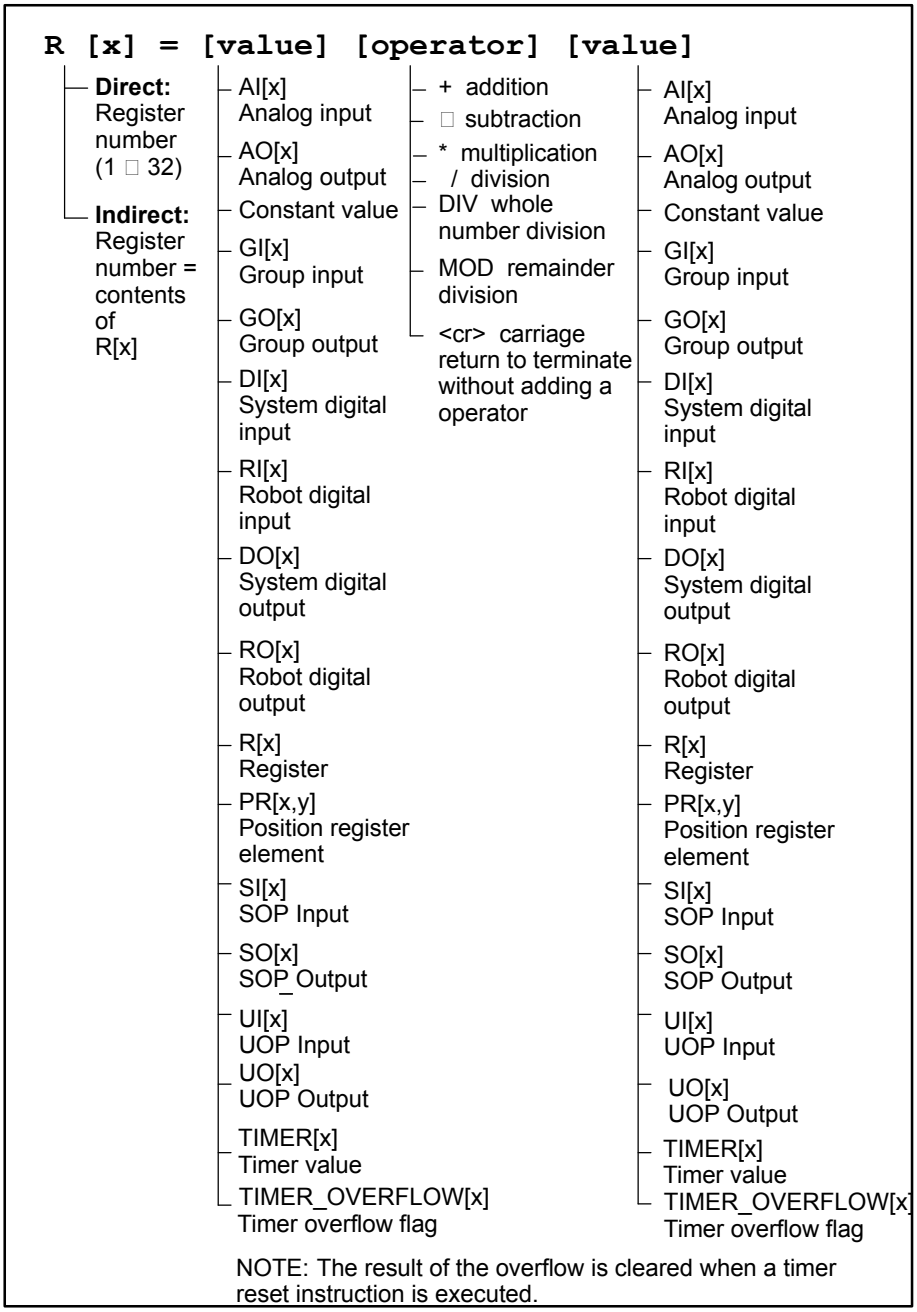
See Figure 6–61 .

You can use multiple arithmetic operators in a single instruction. However, there are the following limitations:

- You can mix + and - in the same instruction. Arithmetic operations within an instruction that mixes + and - will be performed from left to right. You cannot mix * or / in an instruction that already contains + or -.
- You can mix * and / in the same instruction. Arithmetic operations within an instruction that mixes + and - will be performed from left to right. You cannot mix + or - in an instruction that already contains * or /.
- The maximum number of arithmetic operators you can have in the same instruction is 5.

Figure 6–61. R[x] = [value] [operator] [value]




R [x] = [value] [operator] [value]
Direct: Register number (1 □ 32)
Indirect: Register number = contents of R[x]
AI[x] Analog input
AO[x] Analog output
Constant value
GI[x] Group input
GO[x] Group output
DI[x] System digital input
RI[x] Robot digital input
DO[x] System digital output
RO[x] Robot digital output
R[x] Register
PR[x,y] Position register element
SI[x] SOP Input
SO[x] SOP Output
UI[x] UOP Input
UO[x] UOP Output
TIMER[x] Timer value
TIMER_OVERFLOW[x] Timer overflow flag
+ addition
□ subtraction
* multiplication
/ division
DIV whole number division
MOD remainder division
<cr> carriage return to terminate without adding a operator
AI[x] Analog input
AO[x] Analog output
Constant value
GI[x] Group input
GO[x] Group output
DI[x] System digital input
RI[x] Robot digital input
DO[x] System digital output
RO[x] Robot digital output
R[x] Register
PR[x,y] Position register element
SI[x] SOP Input
SO[x] SOP Output
UI[x] UOP Input
UO[x] UOP Output
TIMER[x] Timer value
TIMER_OVERFLOW[x] Timer overflow flag
NOTE: The result of the overflow is cleared when a timer reset instruction is executed.

6.6 POSITION REGISTER INSTRUCTIONS

6.6.1 Overview

A *position register* stores positional information (x,y,z,w,p,r, configuration). Up to 200 position registers are available for all programs in the controller combined. The default number of position registers is 10. Position registers are identified by numbers.

Note You cannot use PR[1] as a position register in your programs, as SpotTool uses this internally as a home position register.

You can increase the number of position registers at controlled start. Refer to the "System Operations" appendix of the Setup and Operations Manual for information on performing a controlled start.

There are two kinds of position register instructions:

- Instructions that manipulate position registers, PR[x]
- Instructions that manipulate position register elements, PR[i,j]

If your system is configured to have more than one group, you can set the*group mask* when you create any position register instruction. The group mask allows you to use function keys to specify:

- Whether the group mask will be used. If the group mask is not used, the position register instruction affects the default group only.
- The group or groups that the position register instruction will affect.

6.6.2 PR[x] Position Register Instructions

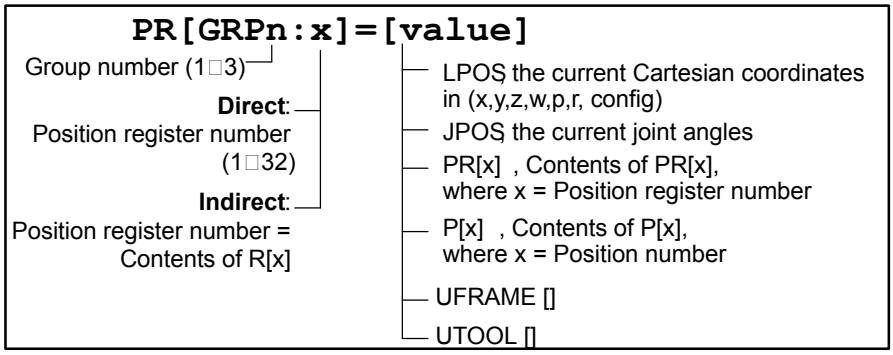
PR[GRPn:x] position register instructions manipulate the position register. They include assignment, addition, and subtraction instructions.

PR[GRPn:x] = [value]

The PR[GRPn:x] = [value] instruction stores positional information in a position register. See Figure 6–62 .

Figure 6–62. PR[GRPn:x] = [value]





PR[GRPn:x]=[value][operator] [value]

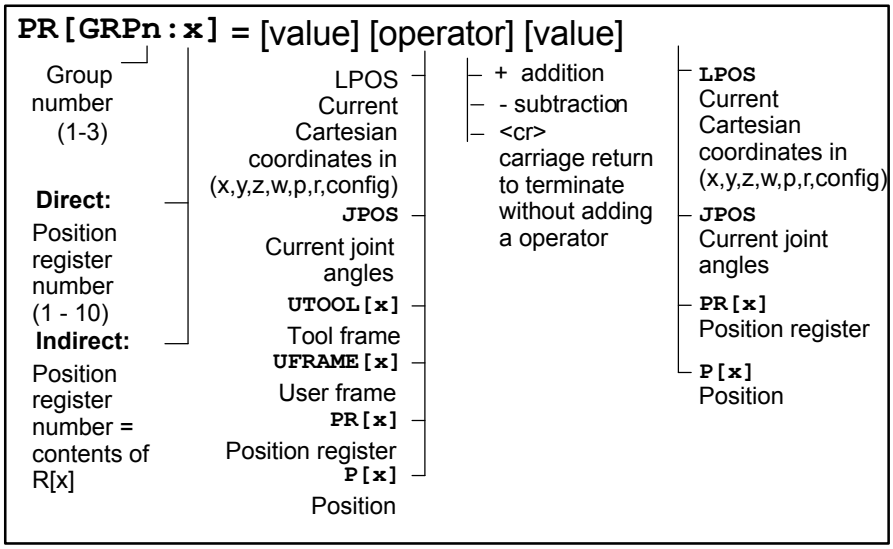
The PR[GRPn:x] = [value] [operator] [value] instructions store the result of an arithmetic operation in a register. The arithmetic operations are addition and subtraction. See Figure 6–63 .

You can use multiple arithmetic operators in a single instruction. However, there are the following limitations:

- You cannot mix +, -, or */ in the same instruction.
- The maximum number of arithmetic operators you can have in the same instruction is 5.

Figure 6–63. PR[GRPn:x] = [value] [operator] [value]

```
PR[...] = ... + ...
PR[...] = ... □ ...
```

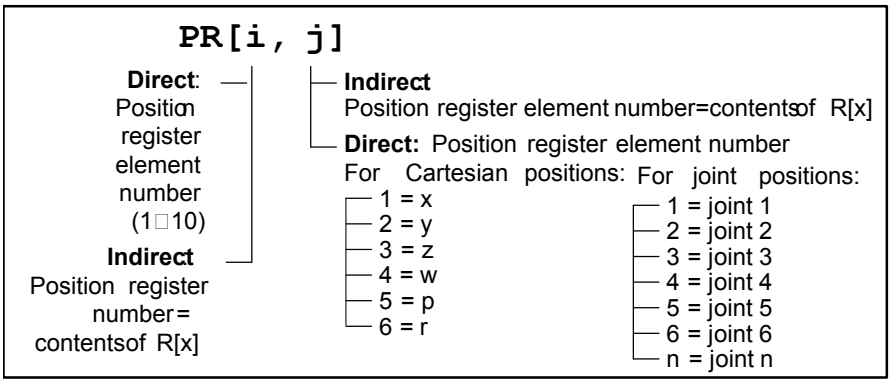


6.6.3 PR[i,j] Position Register Element Instructions

PR[i,j] position register element instructions manipulate a specific position register element. A position register element is one element of a specified position register. In the designation PR[i,j], the **i** represents the position register number and the **j** represents the position register element.

Position register element instructions include assignment, addition, and subtraction instructions. See Figure 6–64 .

Figure 6–64. Position Register Element PR[i,j]

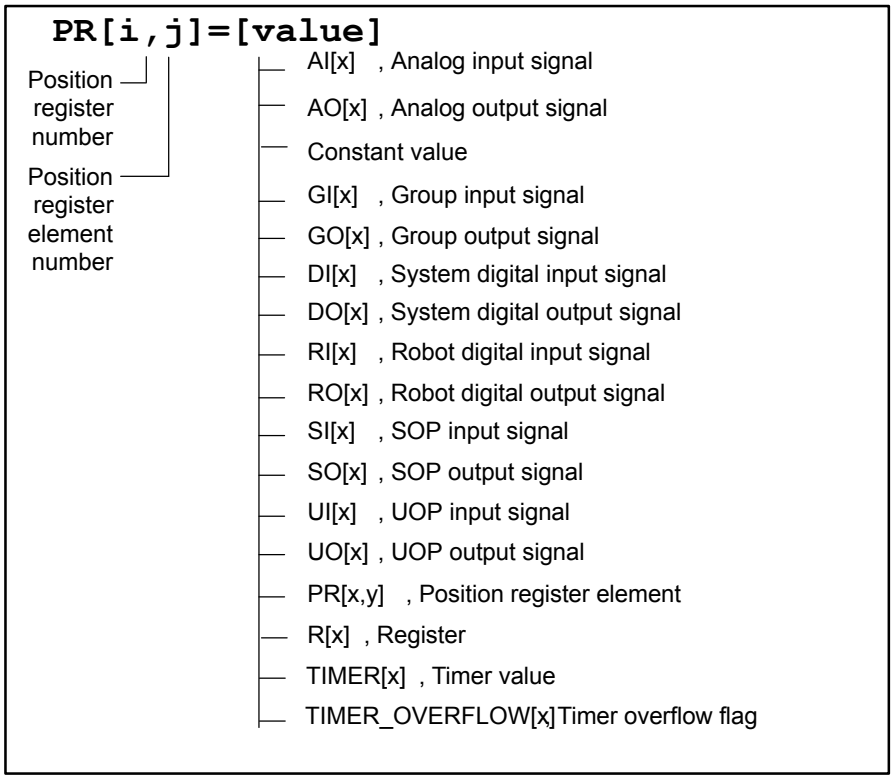


PR[i,j] =[value]

The PR[i,j] = [value] instruction stores positional information in a position register element. See Figure 6–65 .

Figure 6–65. PR[i,j] = [value]

```
PR[...,...] = ...
```



PR[i,j]=[value][operator][value]

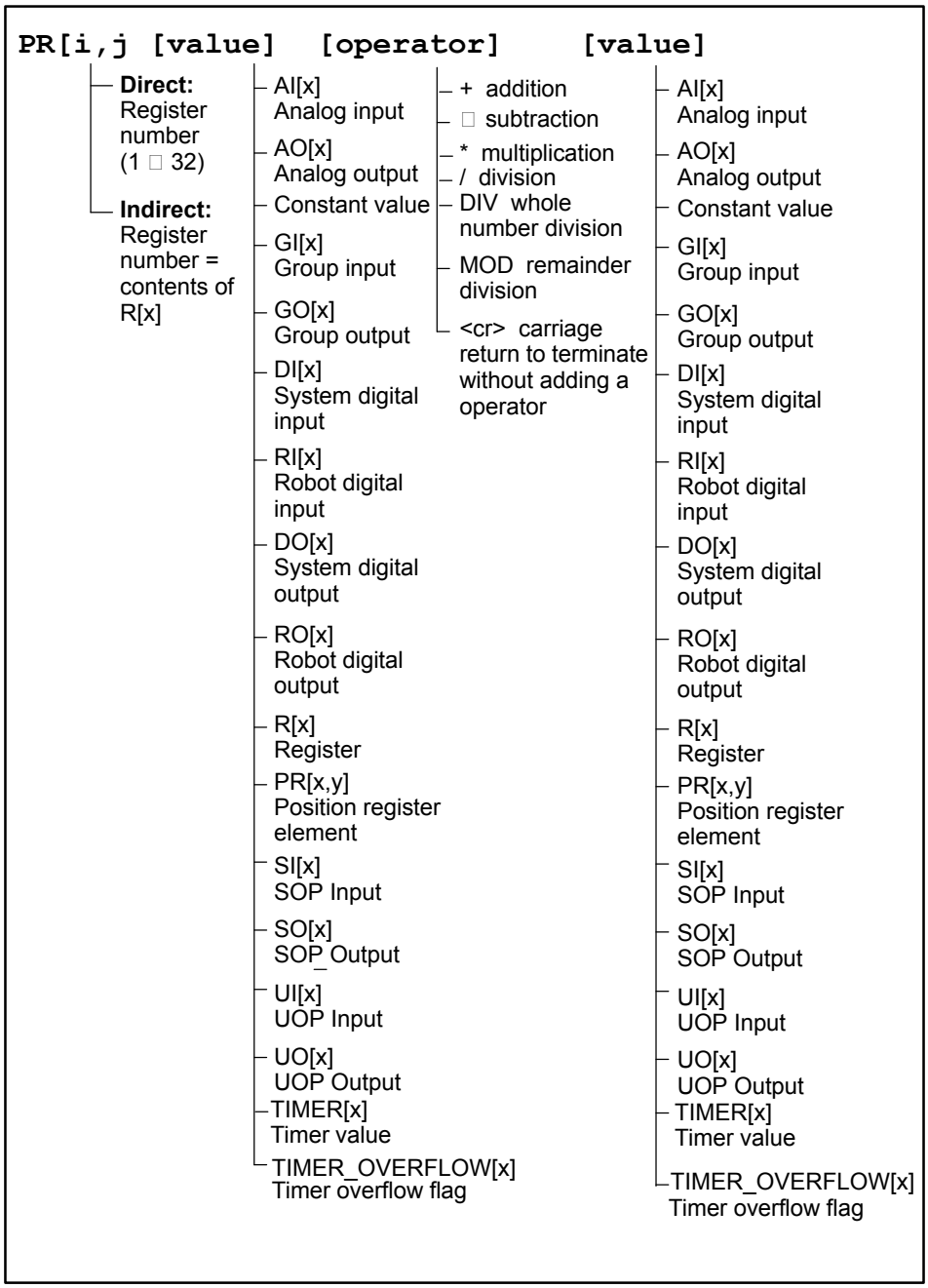
The PR[i,j] = [value] [operator] [value] instructions store the result of an arithmetic operation in a position register element. The arithmetic operations are addition, subtraction, multiplication, division, whole number division (DIV), and remainder division (MOD). See Figure 6–66 .

You can use multiple arithmetic operators in a single instruction. However, there are the following limitations:

- You can mix + and - in the same instruction. Arithmetic operations within an instruction that mixes + and - will be performed from left to right. You cannot mix * or / in an instruction that already contains + or -.
- You can mix * and / in the same instruction. Arithmetic operations within an instruction that mixes + and - will be performed from left to right. You cannot mix + or - in an instruction that already contains * or /.
- The maximum number of arithmetic operators you can have in the same instruction is 5.

Figure 6–66. PR[i,j] = [value] [operator] [value]

```
PR[...,...]=...+...
PR[...,...]=...□...
PR[...,...]=...*...
PR[...,...]=.../...
PR[...,...]=...DIV...
PR[...,...]=...MOD...
```



6.7 STRING REGISTER INSTRUCTIONS

6.7.1 Overview

A string register stores alphanumeric character strings. Each string register can hold a maximum of 254 characters. The default number of string registers is 25.

You can increase the number of string register at controlled start. Refer to the “System Operations” appendix of the Setup and Operations Manual for information on performing a controlled start.

There are several kinds of string register instructions:

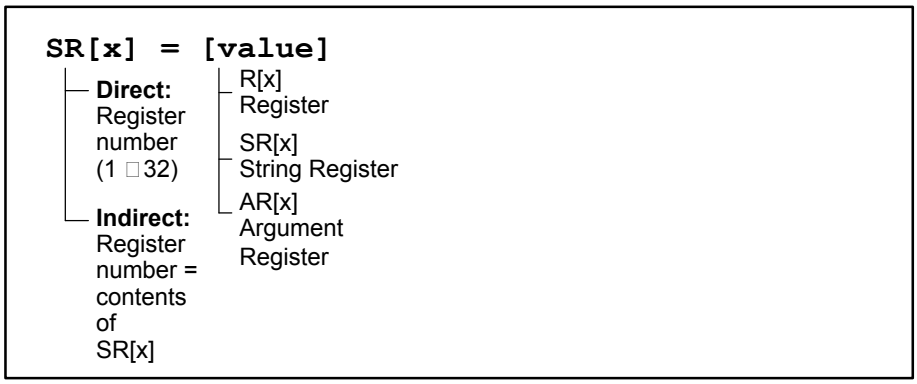
- String register assignment, SR[x]=
- String register concatenation, SR[x]+
- String register comparison, SR[x]<, >, =, <>, <=, >=
- String register length, R[y]=STRLEN SR[x]
- String register substring search, R[y]=FINDSTR SR[x], SR[z]
- String register substring cut, SR[x]=SUBSTR SR[n],R[y],R[z]

String registers can be addressed using direct and indirect techniques. Please see Figure 6.46 in the Register Instructions section for more information.

6.7.2 String Register Assignment and Concatenation

SR[x] string register instructions manipulate the string register. The string register supports assignment and concatenation.

Figure 6–67. String Register Assignment



Examples of string register assignment instructions:

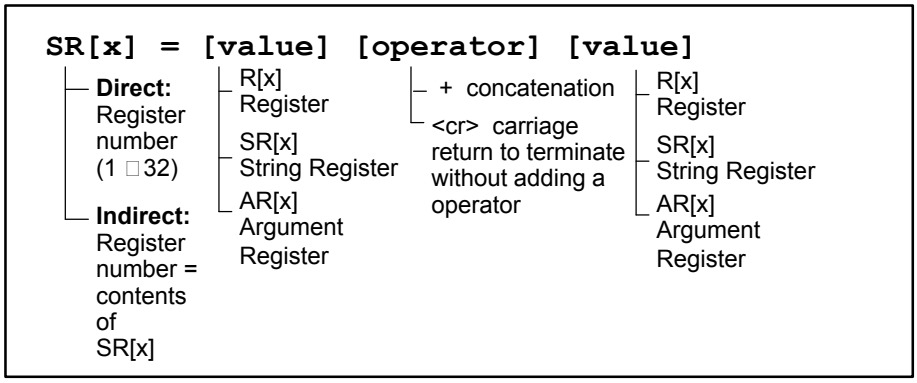
SR[x]=SR[y]

SR[x]=AR[y]

SR[x]=R[y]

R[x]=SR[y]

Figure 6–68. String Register Concatenation



Examples of string register concatenation instructions:

SR[x]=SR[y]+SR[z]

SR[x]=SR[y]+R[z]

SR[x]=AR[y]+SR[z]+R[n]

String register instructions automatically convert numeric data types. This automatic conversion is explained in section Section 6.7.3 .

6.7.3 String Conversion and Precedence

String and numeric data are automatically converted when using any register instruction.

Conversion from either integer or floating point numeric data is allowed. Floating point data will be rounded to 6 decimal places.

Examples of conversions from numeric to string data:

Table 6–12. SR[x]=R[y]

R[y] Value	SR[x] Result
R[y]=1234	SR[x]='1234'
R[y]=12.34	SR[x]='12.34'
R[y]=5.123456789	SR[x]='5.123457'

Conversion from strings to numeric data is allowed. An overflow alarm can occur if the maximum value for a floating point number is exceeded in the string representation. Conversion from string to numeric data will stop when the first alpha character is found.

Examples of conversions from string to numeric data:

Table 6–13. R[x]=SR[y]

SR[y] Value	R[x] Result
SR[y]='1234'	R[x] = 1234
SR[y]='12.34'	R[x]=12.34
SR[y]='765abc'	R[x]=765
SR[y]='fgh'	R[x]=0

There is data type conversion precedence when multiple conversions occur within an instruction. The data is converted to the data type of the left instruction for each operation.

Examples of conversion precedence:

Table 6–14. SR[x]=R[y]+SR[z]

R[y]=123.456 + SR[z]='345.678'	SR[x]='456.134'
R[y]=456 + SR[z]='1abc2'	SR[x]='457'

Table 6–15. SR[x]=SR[y]+R[z]

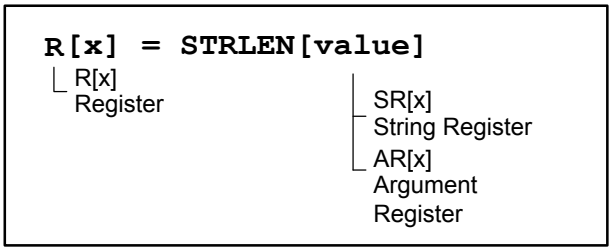
SR[y]='123.' + R[z]=345	SR[x]='123.345'
SR[y]='xyz' + R[z]=81573	SR[x]='xyz81573'

6.7.4 STRLEN Instruction

The STRLEN instruction will return the length of string stored in the specified register.

R[x]=STRLEN SR[y]

Figure 6–69. STRLEN Instruction



Example STRLEN usage:

Table 6–16. R[x]=STRLEN SR[y]:

SR[y] Value	R[x] Result
SR[y]='abcdefghij'	R[x]=10
SR[y]='zyx1,2,3,4,5,6,lm'	R[x]=17
SR[y]=''	R[x]=0

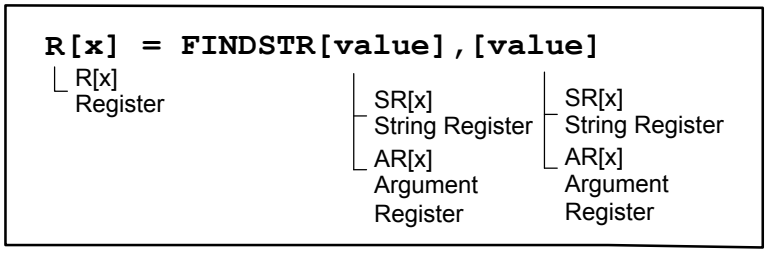
If STRLEN is used with an argument register, AR[x], whose data type is not a string a type mismatch alarm will occur.

6.7.5 FINDSTR Instruction

The FINDSTR instruction will search the target string for the search string and returns the index within the target string. All string comparisons are case insensitive. If the search string is not found within the target string an index of zero, 0, is returned.

R[x]=FINDSTR SR[y],SR[z]

Figure 6–70. FINDSTR Instruction



Example of FINDSTR usage:

Table 6–17. R[x]=FINDSTR SR[y],SR[z]

SR[z] Value with SR[y]='We are searching for a string in this string'	R[x] Result
SR[z]='we'	R[x]=1
SR[z]='string'	R[x]=24
SR[z]='other'	R[x]=0
SR[z]=''	R[x]=0

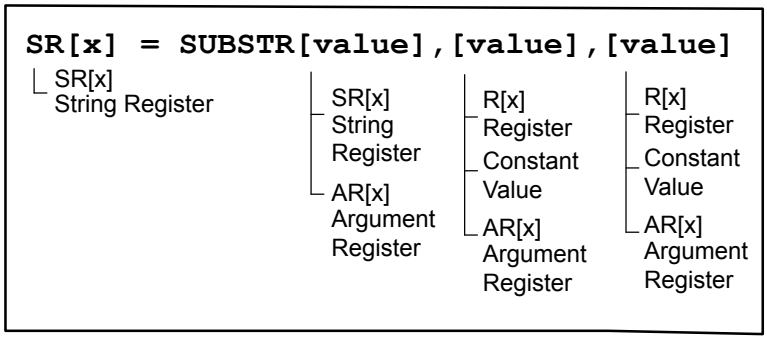
If FINDSTR is used with an argument register, AR[x], whose data type is not a string a type mismatch alarm will occur.

6.7.6 SUBSTR Instruction

The SUBSTR instruction will return a substring from the target string based on the start point and length specified.

SR[n]=SUBSTR SR[x],R[y],R[z]

Figure 6–71. SUBSTR Instruction



Example of SUBSTR usage:

Table 6–18. SR[n]=SUBSTR SR[x],R[y],R[z]

R[y] and R[z] Values with SR[x]='This string will be broken apart.'	SR[x] Result
R[y]=1,R[z]=4	SR[x]='This'
R[y]=28,R[z]=5	SR[x]='apart'
R[y]=8,R[z]=0	SR[x]=''

An overflow alarm will occur in any of the following conditions:

- Starting point is less than or equal to zero
- Length is less than zero
- Starting point is greater than the target string length
- Starting point plus length is greater than the target string length

If an argument register, AR[x], is used as the target string and whose data type is not a string, a type mismatch alarm will occur. If an argument register, AR[x], is used as the starting point or length and whose data type is not a numeric type mismatch alarm will occur.

6.8 INPUT/OUTPUT INSTRUCTIONS

6.8.1 Overview

Input/output, or I/O, instructions allow the program to turn on and off output signals and receive input signals. There are several kinds of I/O instructions:

- Digital input and output instructions
- Robot digital input and output instructions
- Analog input and output instructions
- Group input and output instructions
- PLC I/O
- Welding Input and output instructions

Note I/O instructions are initiated immediately after completion of the previous instruction.

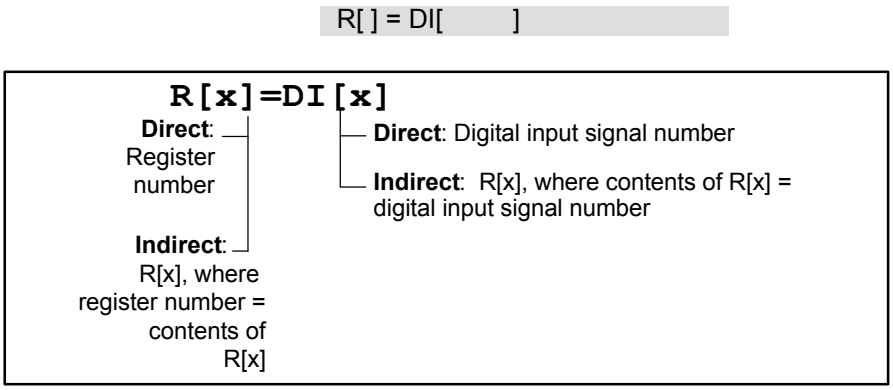
6.8.2 Digital Input and Output Instructions

Digital input (DI) and digital output (DO) signals are user-controlled input and output signals. Use digital input and output instructions to control digital input and output signals in a program. Refer to the "Input/Output (I/O) Setup" chapter in the Setup and Operations Manual.

R[x] = DI[x]

The R[x] = DI[x] instruction stores the condition of a digital input signal line (ON=1, OFF=0) in a register. See Figure 6–72 .

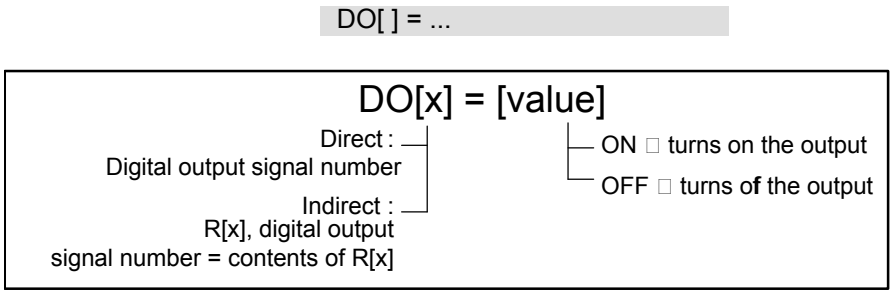
Figure 6–72. R[x] = DI[x]



DO[x] = ON/OFF

The DO[x] = ON/OFF instruction turns on or off the specified digital output signal. See Figure 6–73 .

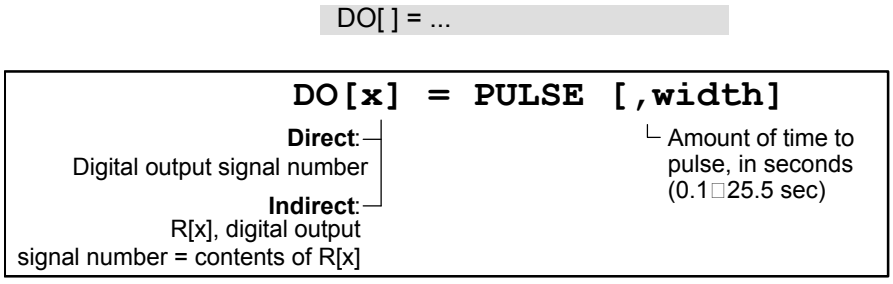
Figure 6–73. DO[x] = ON/OFF



DO[x] = PULSE [,width]

The DO[x]=PULSE [,width] instruction turns on the digital output signal for the time specified. See Figure 6–74 .

Figure 6–74. DO[x] = PULSE [,width]

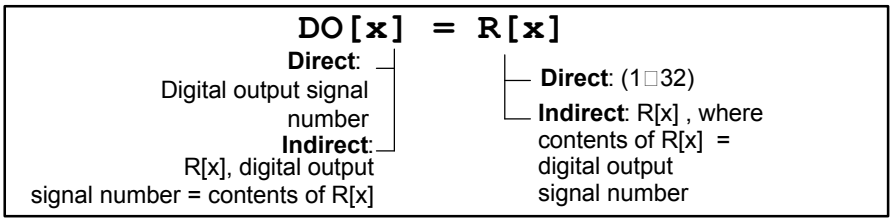


DO[x] = R[x]

The DO[x] = R[x] instruction turns on or off the specified digital output signal based on the value of the register. A value of 0 turns the specified digital output OFF. All values except zero turn the specified digital output ON. See Figure 6–75 .

Figure 6–75. DO[x] = R[x]





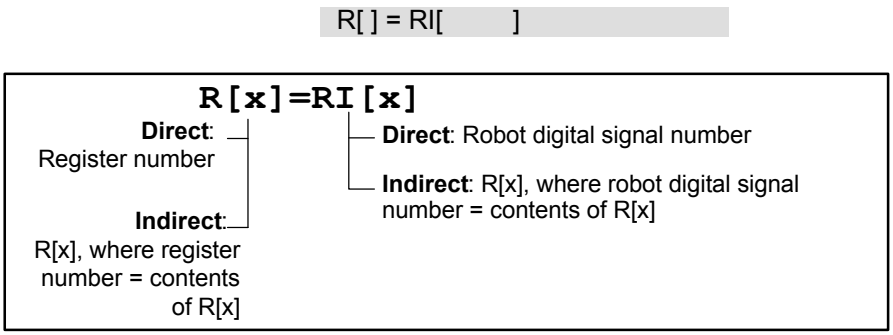
6.8.3 Robot Digital Input and Output Instructions

Robot digital input (RI) and robot digital output (RO) signals are used to communicate between the controller and the robot. Refer to the "Input/Output (I/O) Setup" chapter in the Setup and Operations Manual.

R[x] = RI[x]

The R[x] = RI[x] instruction stores the condition of specified robot digital input signal (ON=1, OFF=0) in a register. See Figure 6–76 .

Figure 6–76. R[x] = RI[x]

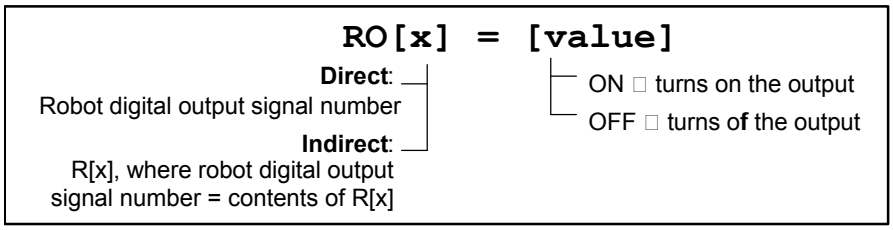


RO[x] = ON/OFF

The RO[x] = ON/OFF instruction turns on or off the specified robot digital output signal. See Figure 6–77 .

Figure 6–77. RO[x] = ON/OFF

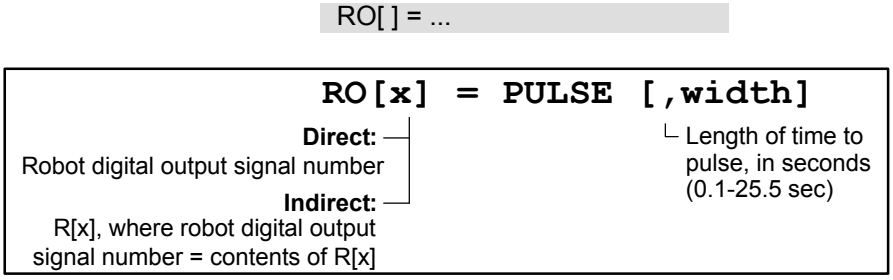




RO[x] = PULSE [,width]

The RO[x]=PULSE [,width] instruction turns on the specified robot digital output signal for the time specified. See Figure 6–78 .

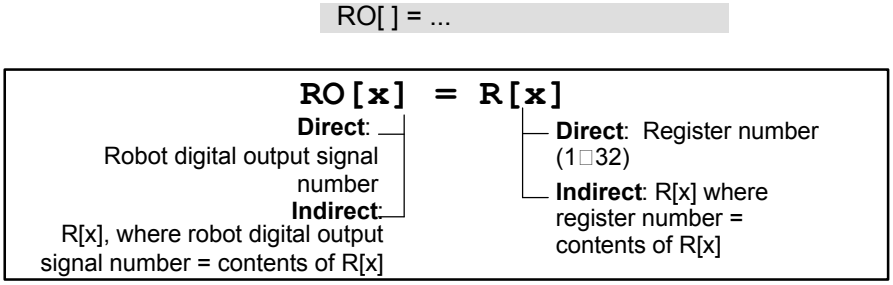
Figure 6–78. RO[x] = PULSE [,width]



RO[x] = R[x]

The RO[x] = R[x] instruction turns on or off the specified robot digital output signal based on the value of the register (1=ON, 0=OFF). See Figure 6–79 .

Figure 6–79. RO[x] = R[x]



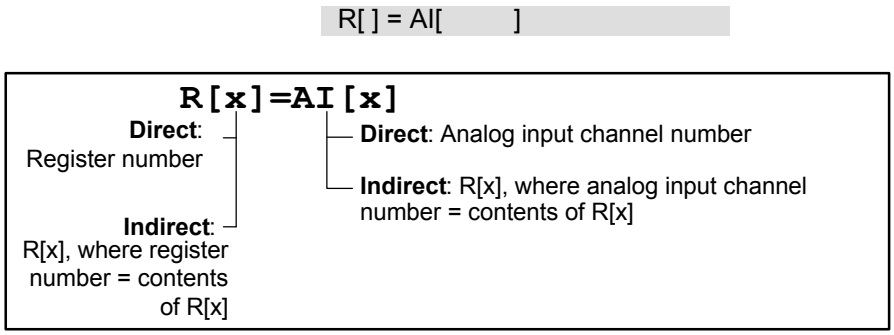
6.8.4 Analog Input and Output Instructions

Analog input (AI) and analog output (AO) signals are continuous input and output signals whose magnitudes indicate data values, such as temperatures and voltages. Refer to the “Input/Output (I/O) Setup” chapter in the Setup and Operations Manual.

R[x] = AI[x]

The R[x] = AI[x] instruction stores the value on an analog input channel in a register. See Figure 6–80 .

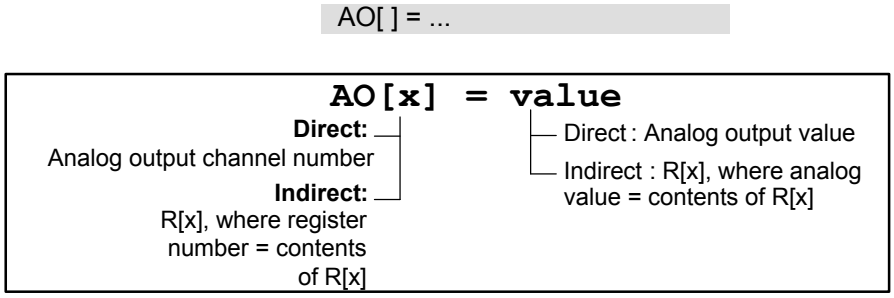
Figure 6–80. R[x] = AI[x]



AO[x] = value

The AO[x]=value instruction sends a value on an analog output channel. See Figure 6–81 .

Figure 6–81. AO[x] = value



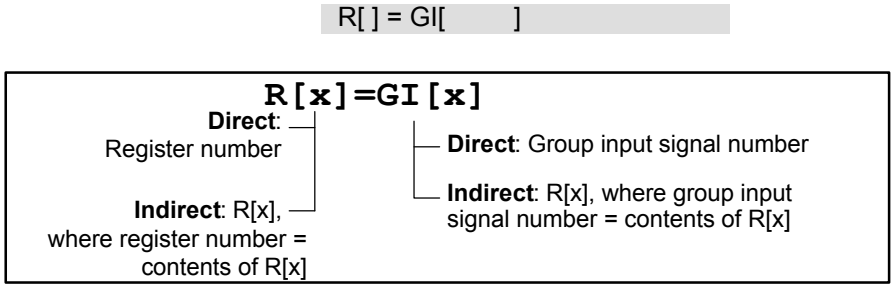
6.8.5 Group Input and Output Instructions

Group input (GI) and group output (GO) signals are several digital input and output signals that have been assigned to a group, can be read as a binary number, and can be controlled by one instruction. Refer to the “Input/Output (I/O) Setup” chapter in the Setup and Operations Manual.

R[x] = GI[x]

The R[x] = GI[x] instruction places the decimal value of the binary number on the specified group input into the specified register. See Figure 6–82 .

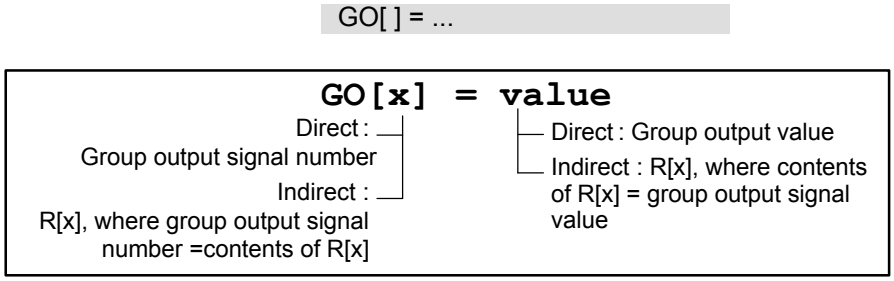
Figure 6–82. R[x] = GI[x]



GO[x] = value

The GO[x]=value instruction sends the binary equivalent of a value on the specified group output lines. See Figure 6–83 .

Figure 6–83. GO[x] = value



6.9 BRANCHING INSTRUCTIONS

6.9.1 Overview

Branching instructions cause the program to branch, or jump, from one place in a program to another. There are three kinds of branching instructions:

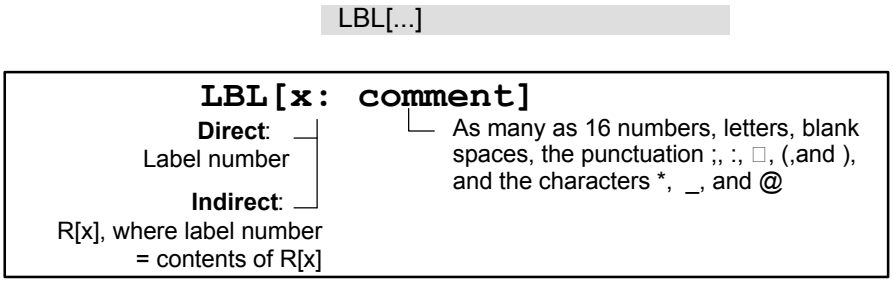
- Label definition instruction
- Unconditional branching instructions
- Conditional branching instructions

6.9.2 Label Definition Instruction LBL[x]

A *label* marks the location in a program that is the destination of a program branch. A label is defined using a *label definition instruction*.

A comment can be added to describe the label. After a label has been defined, it can be used with conditional and unconditional branching instructions. See Figure 6–84 .

Figure 6–84. LBL[x]



6.9.3 Unconditional Branching Instructions

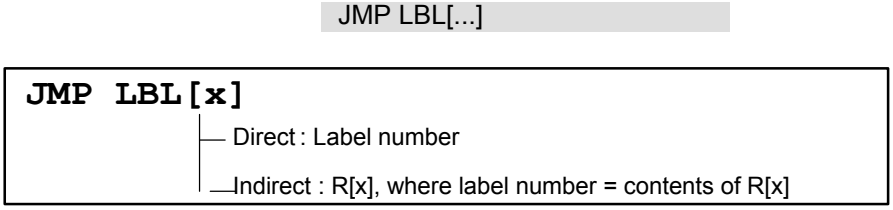
Unconditional branching instructions branch from one place in a program to another any time they are executed. There are two kinds of unconditional branching instructions:

- **Jump instructions** - Cause the program to branch to a named label.
- **Sub program call instructions** - Cause the program to branch to another program.

JMP LBL[x]

The JMP LBL[x] instruction causes the program to branch to the specified label. See Figure 6–85 .

Figure 6–85. JMP LBL[x]

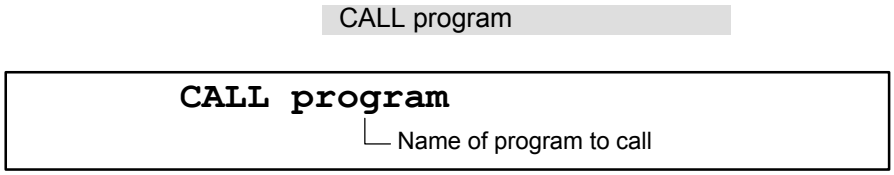


CALL program

The CALL program instruction causes the program to branch to another program and execute it. When the called program finishes executing, it returns to the main program at the first instruction after the call program instruction. See Figure 6–86 . Refer to Section 6.18 for information on program call parameters.

In DispenseTool, this instruction can be used only in a job program.

Figure 6–86. CALL program



Note When you enable an application process for a program, you are setting the application mask. The following rules dictate the kinds of programs you can CALL when application masks are used:

- A program with an application mask set to NONE can CALL a program (using the program CALL instruction) that has an application mask set to a specific application (SpotTool+, HandlingTool, or DispenseTool).
- A program with an application mask set to a specific application can CALL a program that has an application mask set to NONE or to the same application.
- A program with an application mask set to a specific application cannot CALL a program that has a different application mask.

Note If you have the Parameters for CALL Program option, you can include parameters in the CALL program instruction. Refer to Section 6.18 for more information.

END

The program end instruction signals the end of a program. See Figure 6–87 .

Figure 6–87. Program End Instruction



6.9.4 Conditional Branching Instructions

Conditional branching instructions branch from one place to another in a program, depending on whether certain conditions are true. There are two kinds of conditional branching instructions:

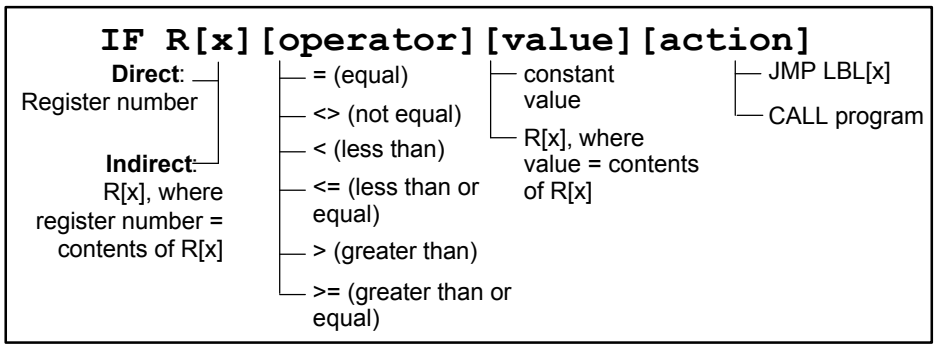
- **IF instructions** - Branch to a specified label or program if certain conditions are true. There are register IF instructions and input/output IF instructions.
- **SELECT instructions** - Branch to one of several jump or call instructions, depending on the value of a register.

IF R[x] [operator] [value] [action]

Register IF instructions compare the value contained in a register with another value and take an action if the comparison is true. See Figure 6–88 .

Figure 6–88. Register IF Instruction

```
IF R[...] = ... ...
IF R[...] <> ... ...
IF R[...] < ... ...
IF R[...] <= ... ...
IF R[...] > ... ...
IF R[...] >= ... ...
```



IF [I/O] [operator] [value] [action]

Input/output IF instructions compare an input or output value with another value and take an action if the comparison is true.

See Figure 6–89 to Figure 6–91 .

Figure 6–89. I/O IF Instruction for DI/DO, RI/RO, SI/SO and UI/UO

```
Example
IF DI[...] = ... ...
IF DO[...] <> ... ...
```

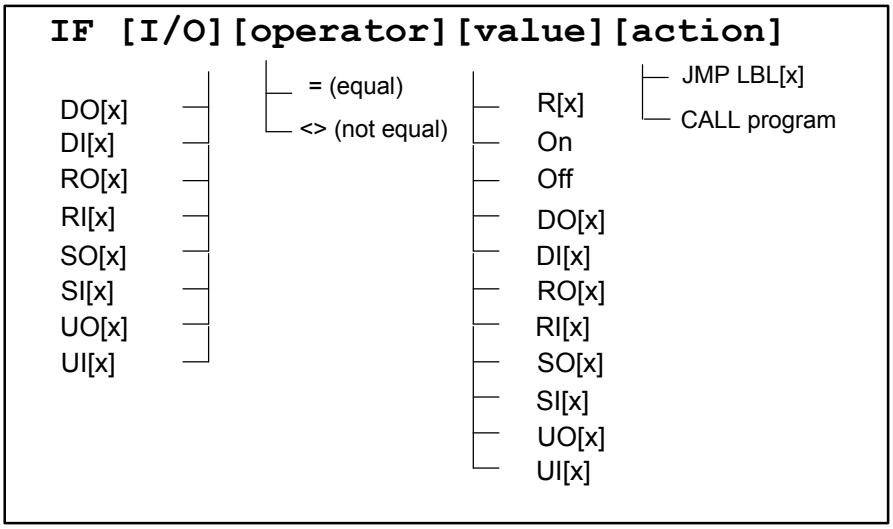


Figure 6–90. I/O IF Instruction for PL

```
Example
IF PL[...] = ... ...
IF PL[...] <> ... ...
```

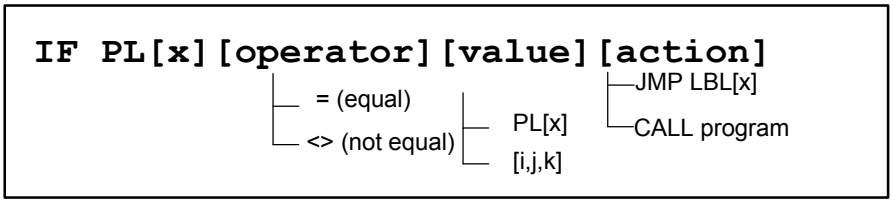
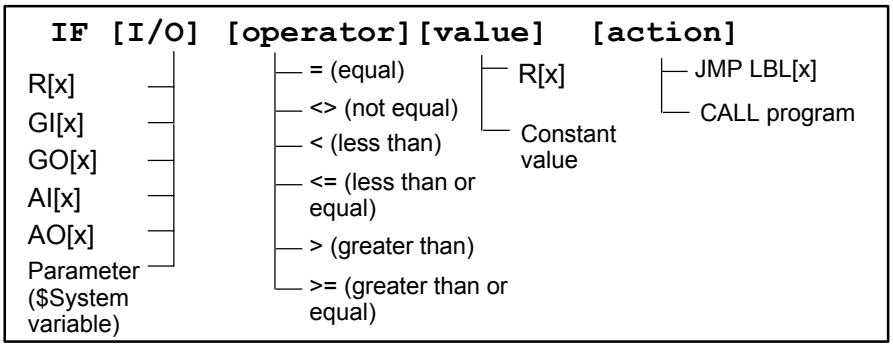


Figure 6–91. I/O IF Instruction for R, AI/AO, GI/GO and System Variable

```
Example
IF R[...]=... ...
IF R[...]<>... ...
IF R[...]<... ...
IF R[...]<=... ...
IF R[...]>... ...
IF R[...]>=... ...
```



For an IF instruction, conditions can be connected using AND or OR, as follows:

- AND operator

  IF [cond1] AND [cond2] AND ..., [action]

  For example,**1: IF R[1]=1 AND R[2]=2 AND DI[2]=ON,JMP LBL[2]**

- OR instruction

  IF[cond1] OR [cond2] OR ..., [action]

  For example,**1: IF DI[10]=ON OR R[7]=R[8], JMPLBL[2]**

  Note You cannot mix the AND and OR operators in the same operation.

When you replace the operator between AND and OR, any operators taught in the same line are also replaced automatically and the following message is displayed.

TPIF-062 AND operator was replaced to OR

TPIF-063 OR operator was replaced to AND

The maximum number of logical conditions that can be taught in the same operation is 5.

IF [cond1] OR [cond2] OR [cond 3] OR [cond4]OR [cond5], [action] (Maximum of five logical conditions)

Mixed Logic IF Statement

The following is an example of a mixed logic IF statement:

```
IF (R[1] = (GI[1] + R[1]) * AI[1]), JMP LBL[1]
IF (DI[1] AND (!DI[2] OR DI[3])), JMP LBL[1]
```

- You can specify mixed logic expressions in the condition part of an IF statement.
- The result of the IF statement must be boolean.
- When the result of the expression is on, the action part of the statement, for example, JMP LBL, is executed.
- The following statements can be used in the action part of an IF statement when mixed logic is used in the condition part:

```
JMP LBL[ ]
CALL
Mixed Logic assignment statement
Pulse statement
```

- Mixed Logic assignment statements and Pulse statements can be specified in the action part of the IF statement only when the condition part is a Mixed Logic expression. See the following example:

```
IF (DI[1]), DO[1]=(On)
IF (DI[2]), DO[1]=Pulse
```

- The maximum number of items (data or operators) in an IF statement is approximately 20. The exact maximum number of items varies according to data type.

Refer to Section 6.30 for more information on mixed logic instructions.

SELECT R[x] = [value1] [action]= [value2] [action]= [valueN] [action] ELSE [action]

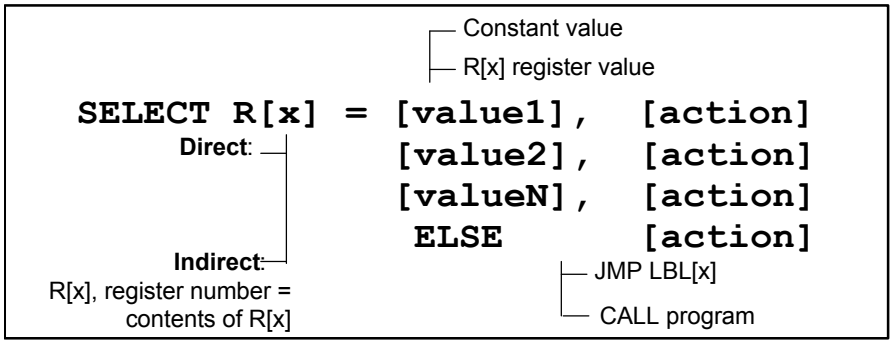
A select instruction compares the value of a register with one of several values and takes an action if the comparison is true:

- If the value of the register equals one of the values, the jump or call instruction associated with that value is executed.
- If the value of the register does not equal one of the values, the jump or call instruction associated with the word ELSE is executed.

See Figure 6–92 .

Figure 6–92. Select Instruction

```
SELECT R[  ] = ...
ELSE ...
```



6.10 WAIT INSTRUCTIONS

Wait instructions delay program execution for a specified time or until a specified condition is true. When a wait instruction is executed, the robot does not execute any motion instructions. There are two kinds of wait instructions:

- **WAIT time** - delays program execution for a specified time.
- **WAIT condition** - delay program execution until specified conditions are true.

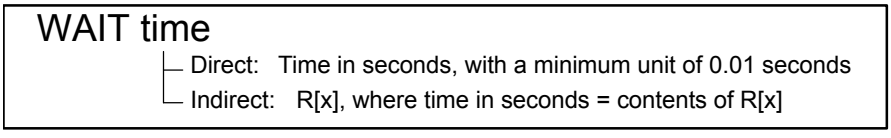
Note Using WAIT instructions in line- or rail-tracking paths is **NOT** recommended. If an E-STOP occurs during the execution of a WAIT instruction, the robot might not continue to track the part until the next position comes into bounds.

WAIT time

The WAIT time instruction delays program execution for a specified time. The time in a WAIT time instruction is specified in seconds, with a minimum unit of 0.01 seconds. See Figure 6–93 .

Figure 6–93. Wait Time

```
WAIT ... (sec)
```



WAIT [item] [operator] [value] [time]

WAIT condition instructions delay program execution until specified conditions are true or until an amount of time elapses (a *timeout* occurs). The timeout can be specified as one of the following:

- **Forever** - the program will wait until the condition is true.
- **Time out, LBL[i]** - the program will wait for the time specified in Timeout. If the condition is still not true, the program will branch to the specified label.

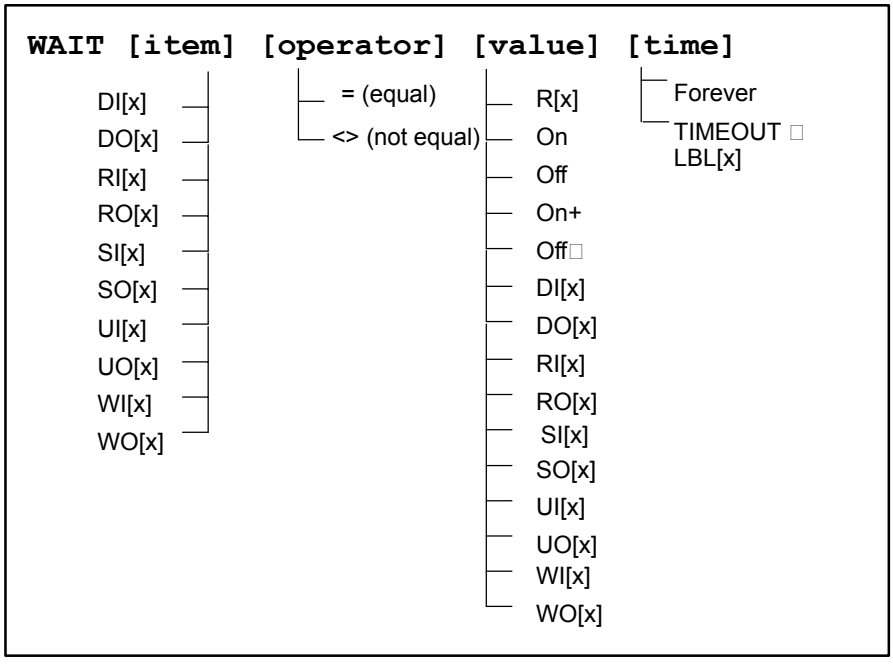
Specify the **timeout** by setting the system variable $WAITTMOUT to a time, in 100ths of a second. The default timeout value is 3000 hundredths of a second. You can set $WAITTMOUT using the parameter name instruction. Refer to Section 6.11.10 for information on the parameter name instruction.

See Figure 6–94 to Figure 6–97 for examples.

Figure 6–94. WAIT Condition for DI/DO, RI/RO, SI/SO, and UI/UO

```
WAIT ... = ...
WAIT ... <> ...
```

Figure 6–95. WAIT Condition for DI/DO, RI/RO, SI/SO, UI/UO, and WI/WO



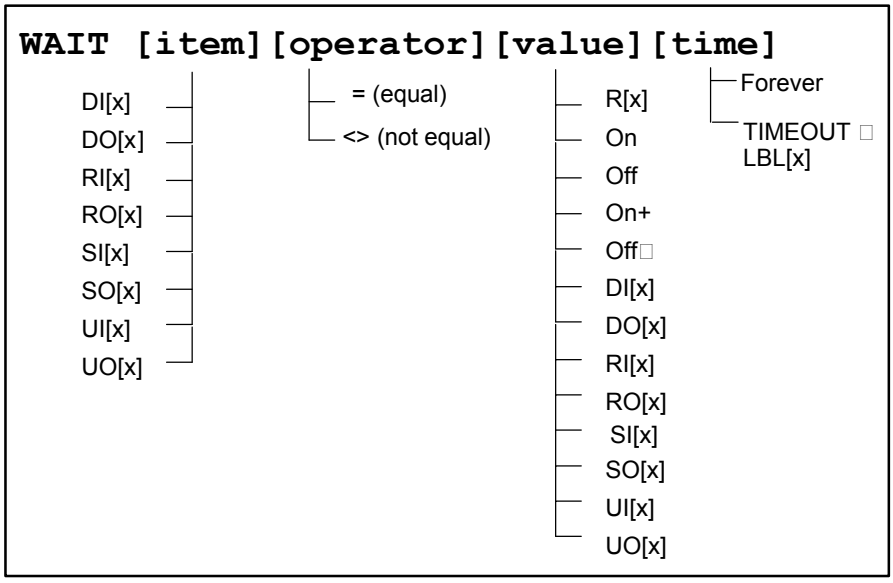


Figure 6–96. WAIT Condition for R, GI/GO, AI/AO, and Parameters

```
WAIT ... = ...
WAIT ... <> ...
WAIT ... < ...
WAIT ... <= ...
WAIT ... > ...
WAIT ... >= ...
```

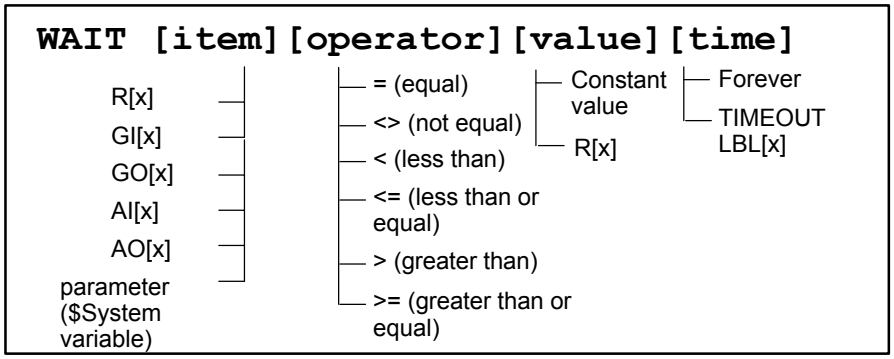
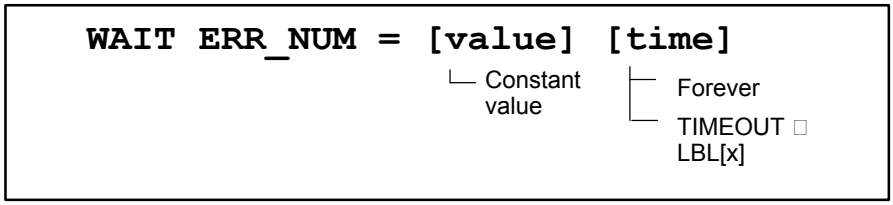


Figure 6–97. WAIT Condition

```
WAIT ERR_NUM = ...
```



Error Number

Where:

ERR_NUM =aaabbb

aaa : Error facility code (decimal); bbb : Error number (decimal)

Refer to the “Error Codes and Recovery” appendix in the Setup and Operations Manual.

If 0 is specified as the error number "aaabbb" when an error occurs, the condition is satisfied.

For example, the instruction

```
WAIT ERR_NUM=11006, CALL PROG_A
```

Will cause the program PROG_A to be called when a "SRVO-0 06 HAND BROKEN" error occurs. (SRVO errors are facility code 11.)

Operators

For WAIT instructions, logical instruction editing can contain multiple logical statements connected by AND or OR operators.

- AND operator

  WAIT [cond1] AND [cond2] AND ...

  For example,**1: WAIT DI[1]=ON AND DI[2]=ON, TIMEOUT,LBL[1]**
- OR instruction

  WAIT [cond1] OR [cond2] OR ...

  For example,**1: IF DI[10]=ON OR R[7]=R[8], JMPLBL[2]**

Note You cannot mix the AND and OR operators in the same operation.

If an instruction contains multiple ORs or ANDs, and you change one of them, the others will also change. In this case, the following message is displayed:

TPIF-062 AND operator was replaced to OR

TPIF-063 OR operator was replaced to AND

The maximum number of logical conditions that you can teach in the same operation is 5. For example

WAIT [cond1] OR [cond2] OR [cond3] OR [cond4]OR [cond5] (Maximum of five logical conditions)

WAIT Mixed Logic Instruction

Figure 6–98. Mixed Logic WAIT Example

```
WAIT (DI[1] AND (!DI[2] OR DI[3]))
```

- Mixed Logic expressions can be specified in the condition of a WAIT statement
- The result of the expression must be boolean.
- The WAIT statement waits until the result of the expression becomes ON.
- "On+","Off-" and "ERR_NUM" cannot be specified in mixed logic instructions. You must use normal logic instructions to specify them.
- The maximum number of items (data or operators) in a WAIT statement is approximately 20. The exact maximum number of items varies according to the data type.

Refer to Section 6.30 for more information on mixed logic instructions.

Output when WAITing on Input

The Output when WAITING on Input feature allows you to turn on a digital output that indicates that a WAIT instruction in a teach pendant program has been waiting on a specified digital input for longer than a specified time. This item is set on the SYSTEM Configuration menu. Refer to "SYSTEM Configuration Setup," in your application-specific *Setup and Operations Manual* for more information.

6.11 MISCELLANEOUS INSTRUCTIONS

6.11.1 Overview

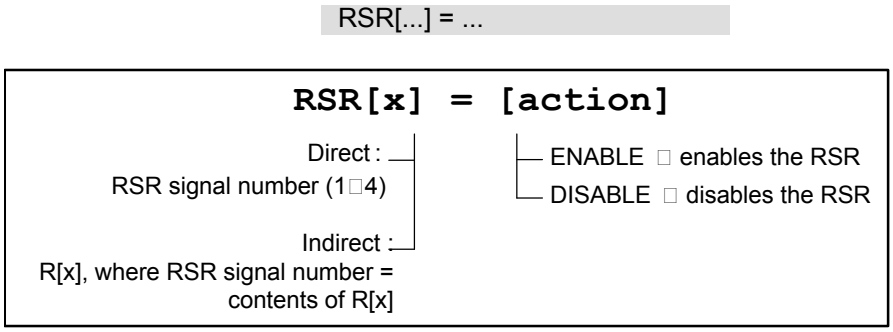
There are miscellaneous instructions for production control, user alarms, timer setting, speed override, program remarks, message handling, and parameter setting.

6.11.2 RSR Enable/Disable Instruction

RSR[x] = [action]

The RSR enable/disable instruction enables and disables the queueing process of the specified RSR. When an RSR signal is set to disable, the RSR signal will be ignored. See Figure 6–99 .

Figure 6–99. RSR Enable/Disable



6.11.3 User Alarm Instruction

The user alarm instruction puts the program in an alarm condition, pauses the program, and causes a message to be displayed on the error message line as follows:

UALM[x]

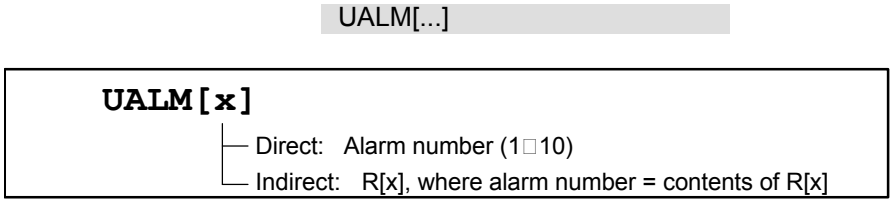
```
INTP 213 UALM[x] Message (prog_name, line_num)
```

For example:

```
INTP 213 UALM[1] Check feeder (RSR001, 47)
```

If the program is resumed, program execution will continue from the next program line. The user alarm instruction specifies the alarm message to be displayed. Refer to the "General Setup" chapter of the Setup and Operations Manual for information on how to define the contents of a user alarm. See Figure 6–100 .

Figure 6–100. User Alarm



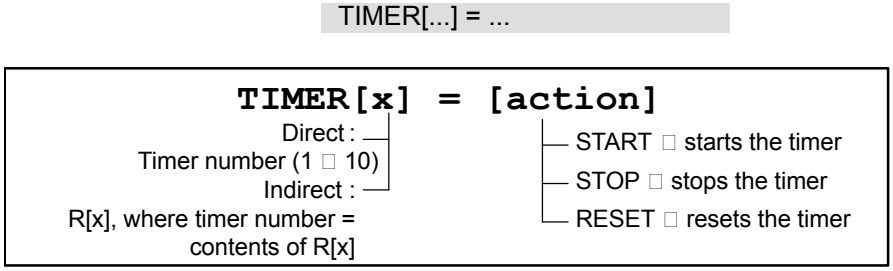
6.11.4 Timer Instruction

Timer instructions allow you to start, stop, and reset up to ten different timers in a program. Timers allow you to determine how long a routine takes to execute, or how long your entire production program takes to execute. Timers can be started in one program and then stopped in another. The status of each timer is displayed in the $TIMER[n] system variable, where n is the number of the timer. See Figure 6–101 .

TIMER[x] = [action]

You can display the status of program timers on the STATUS Prg Timer screen.

Figure 6–101. Timer

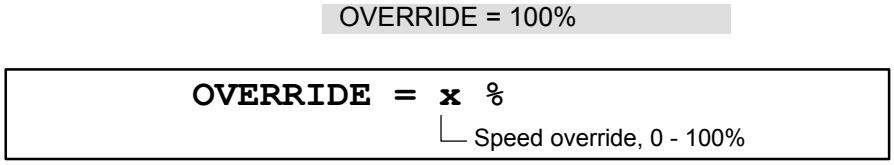


6.11.5 OVERRIDE Instruction

The OVERRIDE instruction sets the speed override to a percentage value of the programmed speed. See Figure 6–102 .

OVERRIDE = x %

Figure 6–102. OVERRIDE



6.11.6 Remark Instruction

The remark instruction allows you to annotate the program. Remark information does not affect the execution of the program. When you add a remark instruction, you enter the message to display within the program. The remark instruction can be from 1 to 32 alphabetic, numeric, punctuation, and blank space characters. The first character of a remark instruction is an exclamation point (!).

```
!remark text
```

6.11.7 Multi-lng Remark Instruction

The multi-lng remark instruction allows you to annotate the program. There are 2 reasons you may wish to use this instead of the original remark instruction:

- You support multiple languages and you wish the remarks to be localized for each language.
- Or you wish to add remarks that are longer than 32 characters.

The multi-lng remark information does not affect the execution of the program. When you add a multi-lng remark instruction, you enter the message to display within the program. The multi-lng remark instruction can be alphabetic, numeric, punctuation, and blank space characters. The first two characters are two hyphens (—). The multi-lng remark has the following characteristics:

- Each remark can be 242 characters although the total for all languages cannot exceed the maximum line length which is also 242 characters.
- The remark will automatically be split among multiple lines in the editor based on the window width. It uses space to split the line so if the remark has no spaces the display will be truncated instead.
- When a new remark is entered it is based on the current language. To enter or display a remark for a different language, you must change the language in the General Setup menu.
- When the language is changed, the remark will only display the two hyphens until you enter a new remark for that language.

- When a program is loaded, it may contain a remark in a language that does not exist on the controller. The language is not created. The remarks will be created but hidden from the editor.
- The .LS file will display all remarks for all languages, even if the language does not exist on the controller.

  In the editor, the remark will look as follows:

```
1:  --This program opens the raw part
 :  gripper
```

  In the .LS file, the remark will look as follows:

```
1:  --eg:This program opens the raw part gripper
 :  --kn:This is KANJI version of remark
 :  --sp:This is Spanish version of remark
 :  --jp:Japanese
 :  --fr:French
 :  --gr:German
 :  --ch:Chinese
 :  --tw:Taiwanese
 :  --ot:Other ;
```

- It is useful to add remarks for multiple languages in the .LS file and use "Ascii Upload" option to create the program. The suffix must be in lower case. The last suffix must contain the ; to end the remark.

6.11.8 Line Remark (Comment out) Instruction

The Line Remark instruction allows you to disable the execution and editing of an line (comment out). Any instruction can be remarked. The Line Remark has the following characteristics:

- The Line Remark instruction will prefix the line with // and then display the line as normal.
- When the language is changed, the Line Remark instruction is localized.
- You cannot create a line with //. A line is created in the normal way. You press [EDCMD] Remark to convert a line to a Line Remark.
- [EDCMD] Remark will allow multiple lines to be selected and F4 REMARK or F5 UNREMARK will remark or unremark all the selected lines.

- You cannot use // to comment out just a portion of a line.
- You cannot edit a line with //. Right arrow in the TP Editor will not move the cursor off the line number. TOUCHUP on a motion line is ignored.
- You can only remove the // using [EDCMD] Remark and selecting unremark.
- The line is still valid even though it is commented out. In other words, a position still exists for the motion lines and a label still exists for the label lines. In the case of a label, the label will be commented out so it is no longer valid to jump to. When a label is unremarked, then the label will be uncommented out unless another label has been added with the same number.
- You can perform other modifications on the line such as delete, copy and paste, find and replace, and renumber.
- Line Remark instructions must be valid instructions in order to successfully load a TP program or to perform Ascii Upload. You can use the Multi-lng Remark Instruction if you need to comment out an invalid instruction for Ascii Upload.
- The toolbars in the icon editor have a Line Remark Toggle button. When pressed, the current line is toggled as a remark/unremark.

6.11.9 Message Instruction

The MESSAGE instruction displays the specified message on the USER screen. The message can be from 1 to 23 alphabetic, numeric, punctuation, and blank space characters. If you want a blank line between messages, leave the message content empty. See Figure 6–103 .

When the MESSAGE [message content] instruction is executed, the user screen is displayed automatically.

MESSAGE [message content]

Figure 6–103. Message Instruction



6.11.10 Parameter Name Instruction

You can display and change the value of a system variable through the parameter name instruction, by using teach pendant read and write operations. Refer to the “Status Displays and Indicators” chapter in the Setup and Operations Manual for more information on system variables.

Note Some system variables only allow you to display their value. Therefore, you might not be able to change the value of some system variables using the parameter name instruction.

There are two kinds of data types for a system variable:

- Numeric data type, which can be stored in a register.
- Position data type, which can be stored in a position register. The following data types are possible:
  - — Cartesian (xyzwpr)
  - — Joint (axes J1 through J6)

When a position data type system variable is stored in a position register, the position register takes on the data type of the system variable.

Note If the system variable you are setting requires a BOOLEAN value (true or false), use 1 for TRUE and 0 for FALSE.

Caution

Do not try to store a numeric system variable in a position register or a position system variable in a numeric register; otherwise, you will cause an error message to be displayed.

Warning

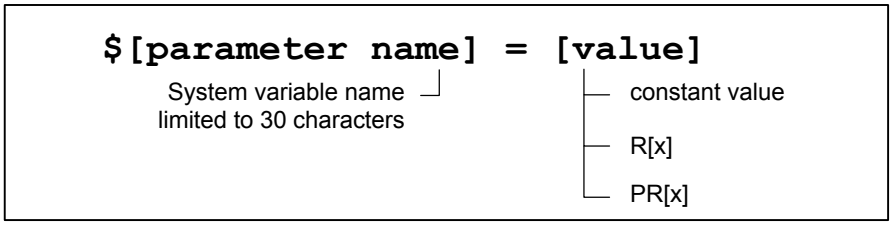
System variables control how the robot and controller operate. Do not set system variables unless you are certain of their effect; otherwise, you could injure personnel, damage equipment, or disrupt the normal operation of the robot and controller.

$[parameter name] = [value]

The $[parameter name]=[value] instruction allows you to change (write) the value of a system variable. See Figure 6–104 .

Figure 6–104. Parameter Name Write Instruction

```
$...= ...
```

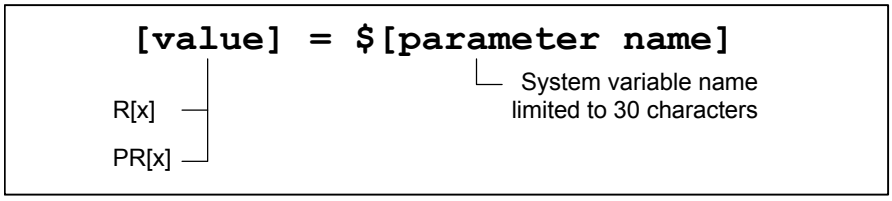


R[ ] / PR[ ] = $[parameter name]

The [value] = $[parameter name] instruction allows you to display (read) the value of a system variable. See Figure 6–105 .

Figure 6–105. Parameter Name Read Instruction

... = $...



6.11.11 Maximum Speed Instruction

The maximum speed instructions set the maximum speed of joint motion and linear or circular motion in the program. If the motion speed exceeds the value designated by this instruction, the motion speed is limited by the designated value.

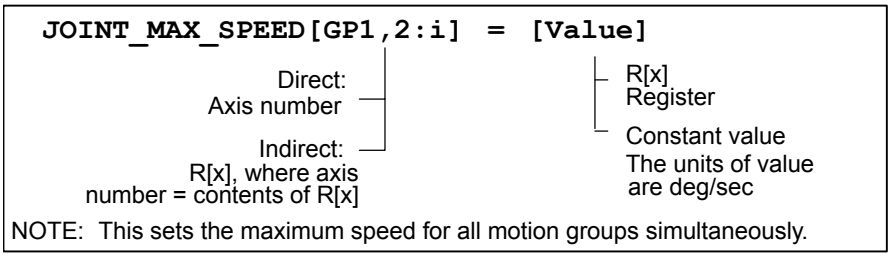
If you use a maximum speed instruction and

- If a macro program is called, the maximum speed value is set back to the default value.
- If a called macro program sets the maximum speed, the maximum speed value is set back to the default value when returning to the calling program.

Figure 6–106 and Figure 6–107 show the maximum speed instructions used in a multiple motion group system.

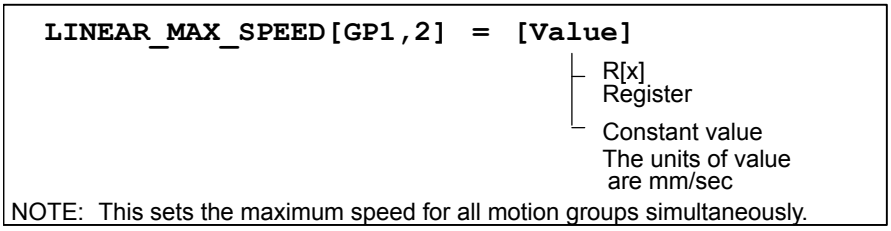
JOINT_MAX_SPEED[...] = ...

Figure 6–106. JOINT_MAX_SPEED Instruction - Multiple Motion Group Syntax



LINEAR_MAX_SPEED[...] = ...

Figure 6–107. LINEAR_MAX_SPEED Instruction - Multiple Motion Group Syntax



The maximum speed instructions have been shown as they would be used in a multiple motion group system. The syntax for the commands is shown in Figure 6–108 and Figure 6–109 when they are used in a single motion group system.

Figure 6–108. JOINT_MAX_SPEED Instruction - Single Motion Group Syntax

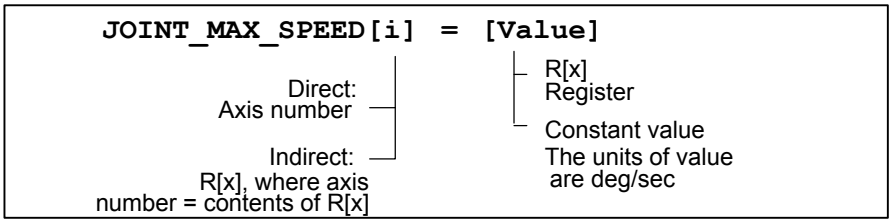
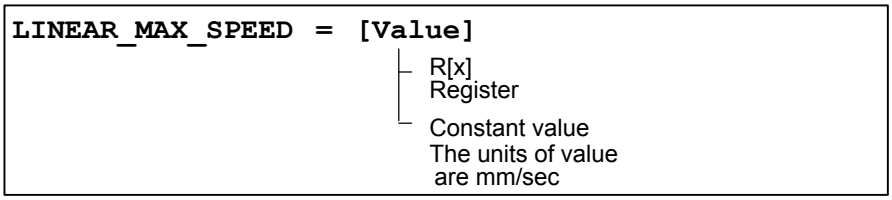


Figure 6–109. LINEAR_MAX_SPEED Instruction - Single Motion Group Syntax



6.12 SKIP INSTRUCTION

```
SKIP CONDITION [I/O] = [VALUE]
```

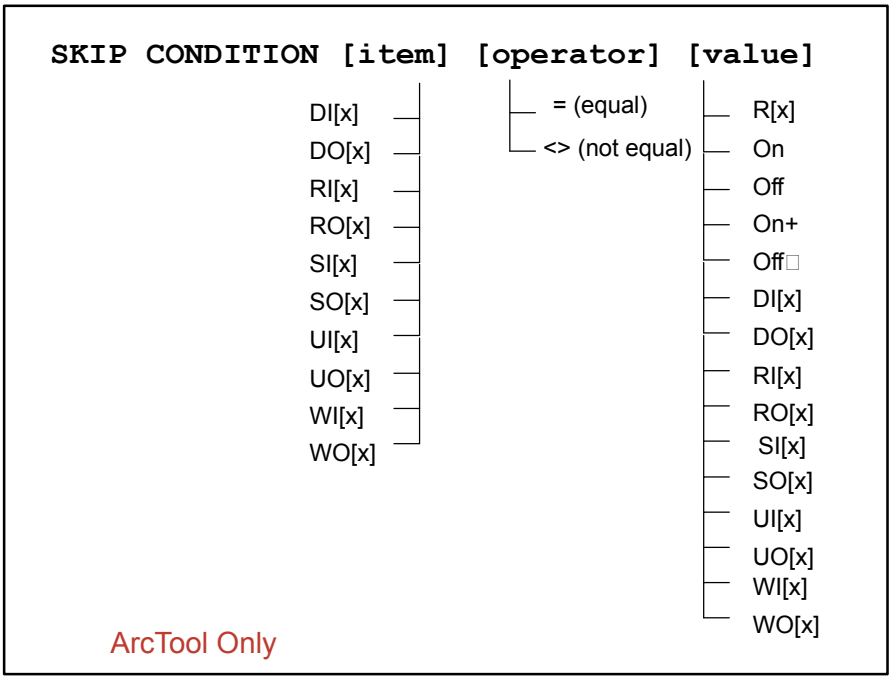
The *skip instruction* sets the conditions to execute robot motion when you use the skip motion option in a motion instruction. These conditions are true until they are reset by another skip instruction. Refer to Section 6.4.10 for more information.

See Figure 6–110 to Figure 6–113 .

Figure 6–110. Skip Condition for DO/DI, RO/RI, SO/SI, and UO/UI

SKIP CONDITION ...=...
SKIP CONDITION ...<>...

Figure 6–111. Skip Condition for DI/DO, RI/RO, SI/SO, UI/UO, and WI/WO



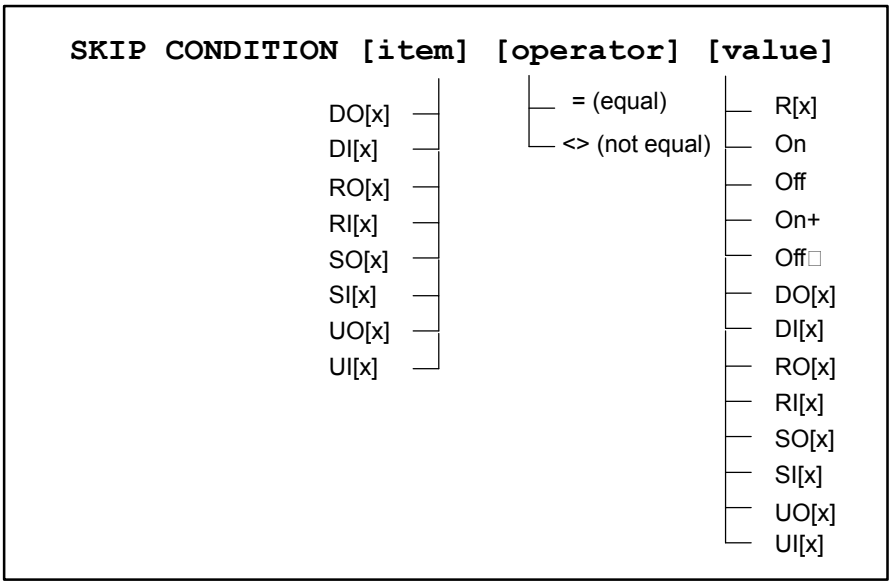


Figure 6–112. Skip Condition for R, GI/GO, AI/AO, and Parameters

```
SKIP CONDITION ...=...
SKIP CONDITION ...<>...
SKIP CONDITION ...<...
SKIP CONDITION ...<=...
SKIP CONDITION ...>...
SKIP CONDITION ...>=...
```

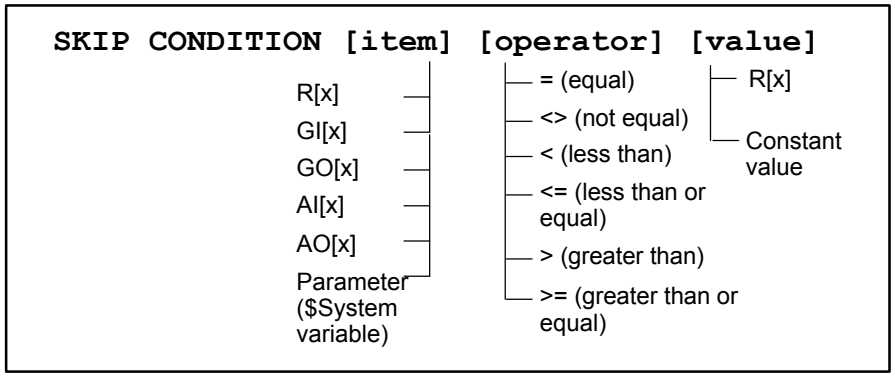
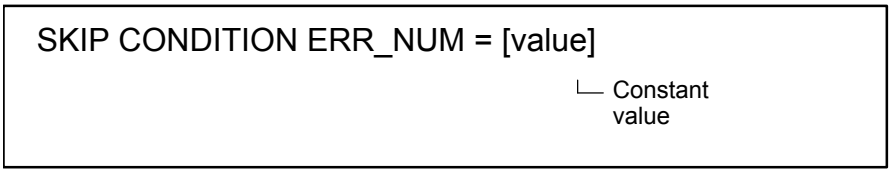


Figure 6–113. Skip Condition

```
SKIP CONDITION ERR_NUM=...
```



Error Number

ERR_NUM =aaabbb

aaa : Error ID (decimal); bbb : Error number (decimal)

Refer to the “Error Codes and Recovery”appendix in the Setup and Operations Manual.

If 0 is specified as error number "aaabbb," when any kind of error occurs, the condition is satisfied.

For example,

```
SKIP CONDITION ERR_NUM=11006
```

This specifies the "SRVO-006 Hand broken" error because SRVO ID number is 11.

Operators

For the SKIP instruction, you can connect conditions using AND or OR operators, as follows:

- AND operator

  SKIP CONDITION [cond1] AND [cond2] AND ...

  For example,

  ```
  1:  SKIP CONDITION R[1]=1 AND R[2]=2
  ```

- OR instruction

  SKIP CONDITION [cond1] OR [cond2] OR ...

  For example,

  ```
  1:  IF DI[10]=ON OR R[7]=R[8], JMP LBL[2]
  1:  SKIP CONDITION R[1]=1 OR R[2]=2
  ```

  Note You cannot mix AND and OR in the same operation.

If you replace the operator between AND and OR, any other operators taught in the same line are also replaced automatically and the following message is displayed:

```
TPIF-062     AND operator was replaced to OR
PIF-063     OR operator was replaced to AND
```

The maximum number of logical conditions that can be taught in the same operation is 5.

SKIP CONDITION [cond1] OR [cond2] OR [cond3]OR[cond4] OR [cond5] (Maximum of five logical conditions)

6.13 QUICK SKIP (HIGH-SPEED SKIP)

6.13.1 Overview

Quick Skip allows you to store the position of the robot when the skip conditions are met in programmed position registers.

Digital servo control stops the robot quickly by developing the maximum torque of the motor when the robot detects that the skip conditions are met.

6.13.2 Execution

The quick skip function can be used in program teaching. There is no need to set system variables.

- The skip conditions for the quick skip function are taught in the same way as the ordinary skip function.
- Teach a quick skip instruction (an additional operation instruction) in the same way as the ordinary line skip instruction. Select the high–speed skip instruction from the additional operation instruction menu.

```
Motion modify JOINT 30%
 1 Skip, LBL, PR     <-Quick skip instruction
 3 Skip, LBL,     <-Ordinary skip instruction
```

- Specify the label, position register, and position storage format.

```
Skip, LBL[10], PR[5]=LPOS
                     or JPOS
```

Sample Program

```
              :
8: SKIP CONDITION SDI[3]=ON
              :
10: LP[2]500mm/sec FINE
```

```
11: LP[3]100mm/sec
  : SKIP, LBL[10], PR[5]=LPOS
                :
                :
30: LBL[10]
```

- When SDI[3] is turned on during execution of the 11th line, the current position is stored in a form of Cartesian coordinates.
- When SDI[3] is not turned on during execution of the 11th line, a branch to LBL[10] is made after the execution of the 11th line ends. In this case, no position data is stored in PR[5].

6.13.3 Limitations

<1> Position read error

As the programmed operation speed is slower, the position read accuracy under skip conditions becomes higher. (As a guideline, an error of about 1.5 mm is generated for 100 mm/sec. The error is proportional to the speed.)

6.14 OFFSET/FRAME INSTRUCTIONS

Offset/frame instructions specify positional offset information or the frames used for positional information. There are five offset instructions:

- **Positional offset condition** - contains information on the distance or degrees to offset positional information

> **Caution**
>
> In ArcTool, recorded positions are not affected by UFRAME, and UFRAME has no effect during playback. However,**position registers are recorded with respect to UFRAME.** If you change UFRAME, any recorded position registers will also change.

- **User frame**
  - — Sets the number of the user frame to use
  - — Defines a user frame
- **Tool frame**
  - — Sets the number of the tool frame to use

— Defines a tool frame

If your system is configured to have more than one group, you can set the *group mask* when you create any offset instruction that contains a position register. The group mask allows you to use function keys to specify:

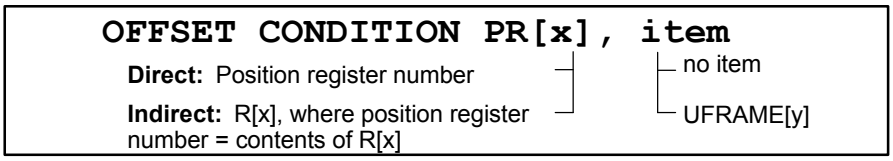
- Whether the group mask will be used. If the group mask is not used, the position register will affect the default group only.
- The group or groups that the position register will affect.

OFFSET CONDITION PR[x] item

The OFFSET CONDITION PR[x] item instruction specifies a position register that contains the offset information used when the OFFSET command is executed. When a user frame is specified in UFRAME[y], that user frame is used when the offset command uses the offset specified in PR[x]. The OFFSET command is entered in the motion instruction. Refer to Section 6.4.10 for more information. See Figure 6–114 .

Figure 6–114. Offset Condition

OFFSET CONDITION ...



UFRAME_NUM = [value]

The UFRAME_NUM=[value] instruction sets the number of the user frame to use. A value of zero indicates that no user frame is used. This means that world frame is used. See Figure 6–115 . Refer to the "General Setup" chapter in the Setup and Operations Manual for information on setting up the user frame.

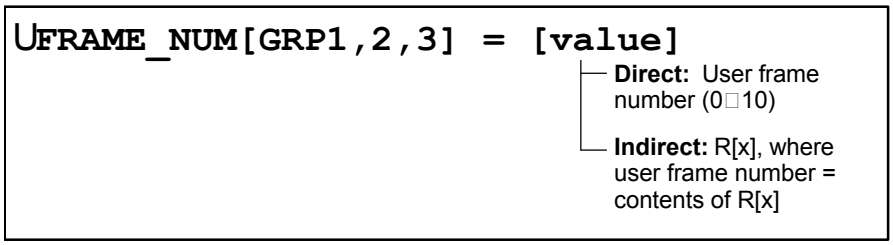
Note To verify that this feature is enabled, check the value of $USEUFRAME and be sure it is set to TRUE. Also, make sure the UFRAME number is not zero before you teach data.

Note You must execute the UFRAME_NUM = value instruction after you insert it into a teach pendant program in order for the subsequent positions to be recorded correctly with respect to the proper user frame.

Note This instruction can be used only if your system has the *Userframe input* option installed.

Figure 6–115. UFRAME_NUM=[value]

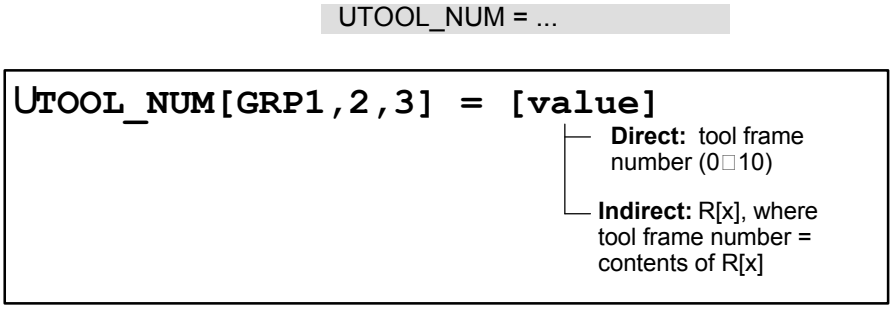
UFRAME_NUM = ...



UTOOL_NUM = [value]

The UTOOL_NUM=[value] instruction sets the number of the tool frame to use. A value of zero indicates that no tool frame is used. This means that the frame defined by the faceplate coordinates is used. See Figure 6–116 . Refer to the "General Setup" chapter in the Setup and Operations Manual for information on setting up the tool frame.

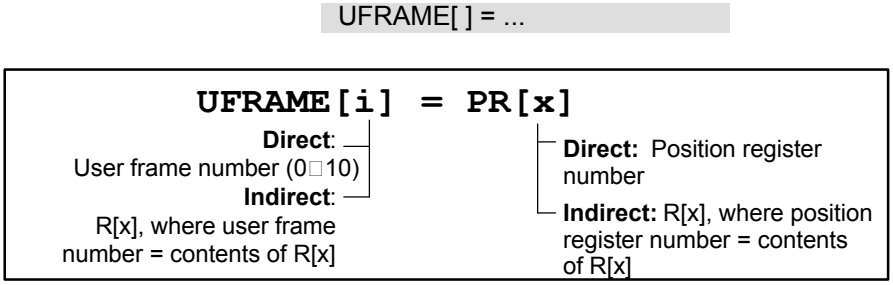
Figure 6–116. UTOOL_NUM=[value]



UFRAME[i] = PR[x]

The UFRAME[i] = PR[x] instruction defines the specified user frame using the information contained in a position register. See Figure 6–117 .

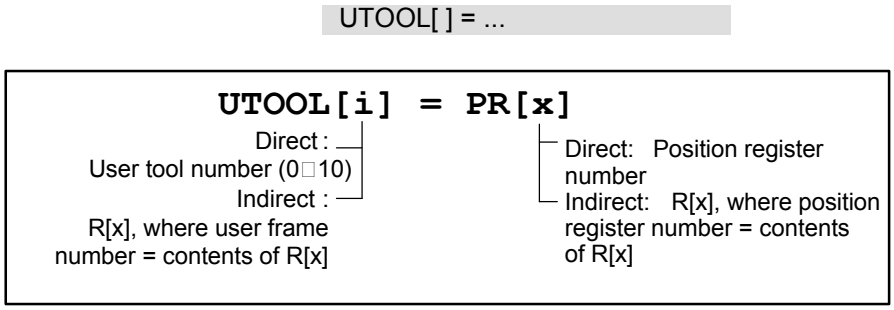
Figure 6–117. UFRAME[i] = PR[x]



UTOOL[i] = PR[x]

The UTOOL[i] = PR[x] instruction defines the specified tool frame using the information contained in a position register. See Figure 6–118 .

Figure 6–118. UTOOL[i] = PR[x]



6.15 TOOL OFFSET CONDITION INSTRUCTION

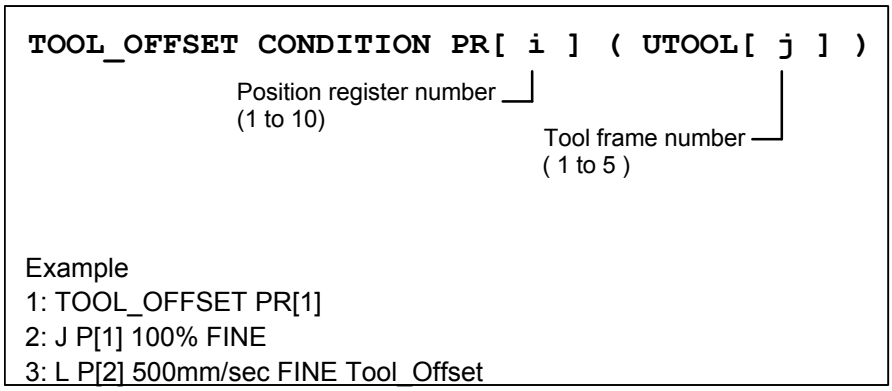
A tool offset condition instruction specifies the offset condition used in a tool offset instruction. Execute a tool offset condition instruction before executing the corresponding tool offset instruction. After the tool offset conditions have been specified, they remain in effect until the program terminates or the next tool offset condition instruction is executed. See the following screen for an example.

```
Instruction
1  Miscellaneous   5  Tool_Offset
2  Program control 6  MACRO
3  Skip            7
4  Offset          8  ---next page---
```

- The position register specifies the direction in which the target position shifts, as well as the amount of shift.
- The tool coordinate system is used for specifying offset conditions.
- When the number of a tool coordinate system is not specified, the currently selected tool coordinate system is used.
- When the position data is given as joint coordinates, an alarm is issued and the program pauses.

A tool offset instruction moves the robot from the target position (that was originally recorded in the position data) to another position by the amount specified using the tool offset condition instruction. The tool offset condition instruction specifies the condition when the offset is applied. See Figure 6–119 .

Figure 6–119. Tool Offset Condition Instruction



6.16 MULTIPLE CONTROL INSTRUCTIONS

Multiple control instructions are used for *multi-tasking*. Multi-tasking allows you to execute more than one task at a time.

In DispenseTool, these instructions can be used only in a job program.

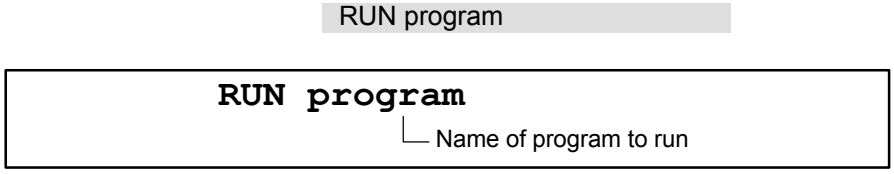
Note In DispenseTool, the system variable $SCR.MAXNUMTASK must be set (at a controlled start) equal to the number of tasks you are using, plus 2. $SCR.MAXNUMTASK initially defaults to 2 because two interpreters are required for internal use by the system. For example, if you are using multiple control instructions, and you have two tasks, $SCR.MAXNUMTASK should be set to 4.

Note In PaintTool, multiple control instructions are available only when the sub type is set to none or macro.

RUN program

The RUN program instruction causes the selected program to begin to execute immediately. The parent program continues to execute. See Figure 6–120 .

Figure 6–120. RUN program

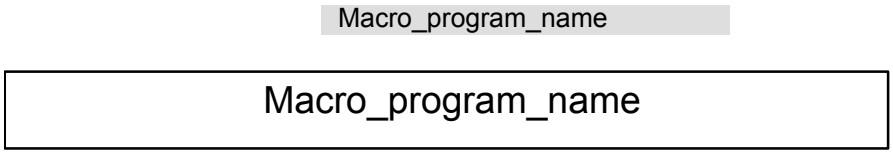


6.17 MACRO COMMAND INSTRUCTION

The *macro command instruction* specifies the macro command to be executed when the program is run. A macro command is a separate program that contains a series of instructions to perform a task.

You can define as many as 20 to 99 macro commands, depending on how your system was set up. In SpotTool+, you can define as many as 40 macro commands. See Figure 6–121 .

Figure 6–121. Macro Command Instruction



Refer to the "General Setup" chapter of this manual for information on how to set up and execute macro commands. Refer to Section 6.18 for information on macro instruction parameters.

Note When using Parameters for Macros, you can include parameters in the macro command instruction. Refer to Section 6.18 for more information.

The DispenseTool software provides macro commands in the following areas.

- General, and macro command setup
- Tool control
  - — Four position tool
  - — Dispense material handling tool
- Interference zone control
- Dispensing equipment-specific control
  - — Common dispensing equipment
  - — ISD Gear Meter
  - — ISD Double Acting Shot Meter
  - — ISD Single Acting Shot Meter

The SpotTool+ software provides predefined macros to assist in the spot welding process. These are:

- CLR OF TRANSFER
- ENTER I-ZONE
- EXIT I-ZONE
- SAFE ZONE

- MOVE TO HOME
- MOVE TO REPAIR
- AT POUNCE
- OPEN CLAMP EARLY
- REPOSITION CLAMP

Note The predefined macros listed above might differ for customized software packages. For SpotTool+ applications, Macro is on page two of the [INST] subwindow.

6.18 PARAMETERS FOR PROGRAM CALL AND MACRO INSTRUCTIONS

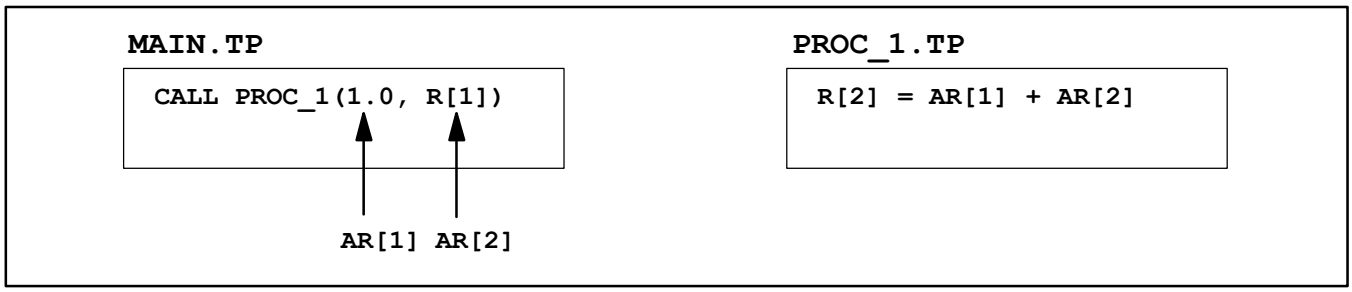
6.18.1 Overview

A *parameter* is an argument you pass with a program call or MACRO instruction to a corresponding sub-program or macro program. This is similar to passing a parameter with a KAREL routine.

In HandlingTool, you can pass parameters to teach pendant programs and macros.

In the example shown in Figure 6–122 , the main program, MAIN.TP, calls the sub program, PROC_1.TP, and uses two parameters. These parameters can be used in the sub-program as two *argument registers, AR[1]* and *AR[2]* .

Figure 6–122. Parameter Example



To use parameters, you do the following:

1. Define the parameter(s) in the CALL program instruction or macro program instruction.
2. Use the parameters within the sub-program or macro program by including one of the permitted instructions.

A parameter can be one of the following:

- Constant
- String
- Argument register (AR[ ])
- Register (R[ ])

This section contains information on the following topics:

- Parameter instruction syntax
- String value selections
- Argument registers
- Parameter guidelines
- Including parameters in program call and MACRO instructions
- Including argument registers in sub-programs

6.18.2 Parameter Instruction Syntax

You can use parameters in the following kinds of instructions:

- Program call instructions
- MACRO instructions

Refer to Table 6–19 for example parameter instructions.

Table 6–19. Parameter Instructions

Instruction Name	Example
Program Call instruction	CALL SUBPRG_1(1, R[3], AR[1])
MACRO instruction	Vacuum_Hand_Release(2.5) Set_UTOOL(1)

Refer to Table 6–20 for example parameter data types.

Table 6–20. Parameter Data Types

Parameter Type	Example
Constant	1, 3.5
String*	'Perch'
Register	R[6]
Argument Register	AR[3]

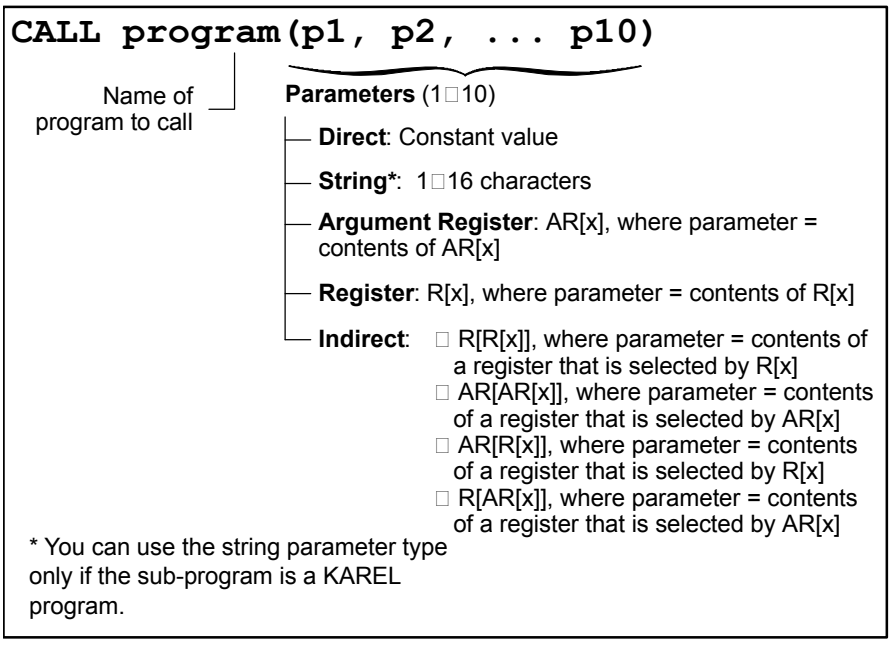
* You can use the string parameter type only if the sub-program is a KAREL program.

Program Call Instructions

The syntax of the program call instruction is shown in Figure 6–123 .

Figure 6–123. CALL Program with Parameters

```
CALL program(p1,p2,...)
```



You cannot use parameters for a program call in a conditional branching instruction such as an IF or SELECT instruction. In this case, you can change the program as shown in Figure 6–124 .

Figure 6–124. Parameters for Conditional Branching Example

Parameters cannot be used

```
IF R[1] = 3, CALL PROC_5
```

Parameters can be used

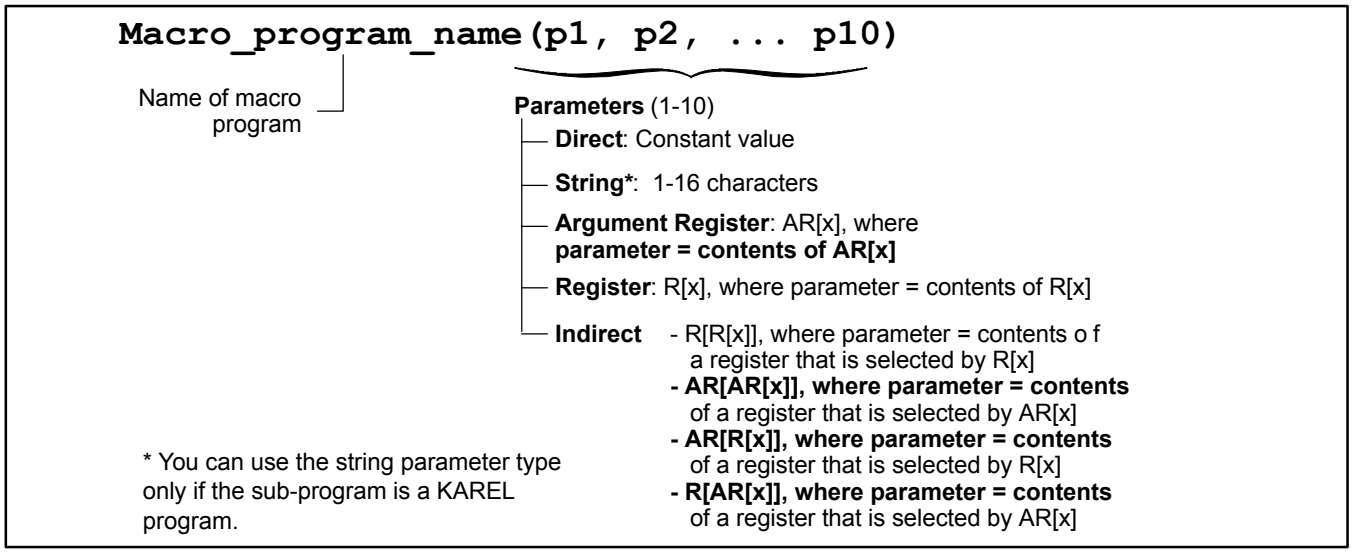
```
IF R[1] <> 3, JMP LBL[1]
CALL PROC_5(1, R[2])
LBL[1]
```

MACRO Instructions

The syntax of the MACRO instruction is shown in Figure 6–125 .

Figure 6–125. MACRO Program with Parameters

```
Macro_program_name(p1,p
2,...)
```



6.18.3 String Value Selections

When you program instructions for call parameters and you select string, the subwindow lists choices. You can control the choices you see in the subwindow by initializing the system variables summarized in Table 6–21 . There are sets of choices: a top-level category and a lower-level sub-category.

Note String parameters can only be passed to a KAREL sub-program.

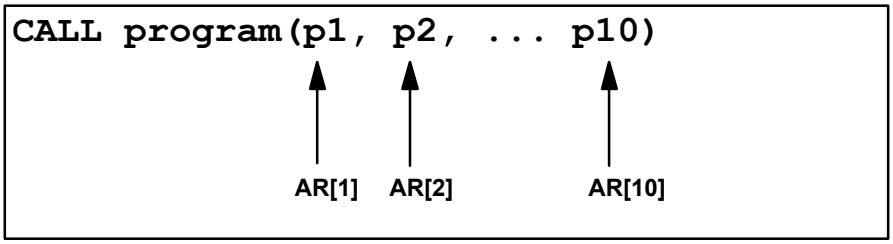
Table 6–21. String Parameter System Variables

Item	System Variables	Notes	HandlingTool Default Value
String category	**Top-level categories:** $ARG_STRING[i].$title i = 1-10	At least one character, up to 16 characters	$ARG_STRING[1].$title = MENUS the menu utility option uses this first entry. Do not make changes here. The menu utility program will overwrite your changes. $ARG_STRING[2].$title = 'PARTS' $ARG_STRING[3].$title = 'TOOL' $ARG_STRING[4].$title = 'WORK' $ARG_STRING[5].$title = 'POS' $ARG_STRING[6].$title = 'DEV' $ARG_STRING[7].$title = 'PALT' $ARG_STRING[8].$title = 'GRIP' $ARG_STRING[9].$title = 'USER' $ARG_STRING[10].$title = 'PREG'
String	**Lower level sub-categories:** $ARG_STRING[*i*].$item *j* i = 1-10 j = 1-20	Up to 16 characters	$ARG_STRING[2].$item1= 'PARTS_ITEM1' $ARG_STRING[3].$item1 = 'TOOL_ITEM1' $ARG_STRING[4].$item1 = 'WORK_ITEM1' $ARG_STRING[5].$item1 = 'POS_ITEM1' $ARG_STRING[6].$item1 = 'DEV_ITEM1' $ARG_STRING[7].$item1 = 'PALT_ITEM1' $ARG_STRING[8].$item1 = 'GRIP_ITEM1' $ARG_STRING[9].$item1 = 'USER_ITEM1' $ARG_STRING[10].$item1 = 'PREG_ITEM1'
Function key label "Words" displayed for F1-F5 keys when you select parameter string, F5, STRINGS.	$ARG_WORD[i] i = 1-5	Up to 7 characters	$ARG_WORD[1] = '$' $ARG_WORD[2] = '[' $ARG_WORD[3] = ']'

6.18.4 Argument Registers

When you refer to a parameter within a sub-program or macro program, you refer to it as an *argument register*. From left to right in the instruction, the first parameter is AR[1], the second parameter is AR[2], and so forth. See Figure 6–126 .

Figure 6–126. Argument Registers



You **can use** the argument registers in specific instructions in the sub-program or macro program. Table 6–22 lists the kinds of instructions that can use argument registers.

Table 6–22. Instructions That Can Use AR[]

You Can Use an Argument Register...	Example
On the right side of an assignment instruction	R[1] =**AR[1]** + R[2] + **AR[4]** IF R[1] =**AR[1]** AND R[2] = **AR[4]** , JMP LBL[1]
	GO[1] =**AR[2]**
	IF R[7] =**AR[1]** , JMP LBL[1] WAIT AI[1] <>**AR[2]** , JMP LBL[1]
	UFRAME_NUM =**AR[3]** UTOOL_NUM = **AR[4]**
As an indirect index in an instruction	R[**AR[1]**] = R[**AR[2]**] DO[**AR[1]**] = ON
As a parameter for program call instruction	CALL SUBPRG_1(**AR[5]**)
As a parameter for MACRO instruction	Hand 3 Release (**AR[1]**]

Note You **cannot use** an argument register as the index for an indirect register, as follows:

```
Not allowed:

R[ R[ AR[1] ] ]
Allowed:

AR[ [ AR[1] ]
```

6.18.5 Guidelines for Using Parameters

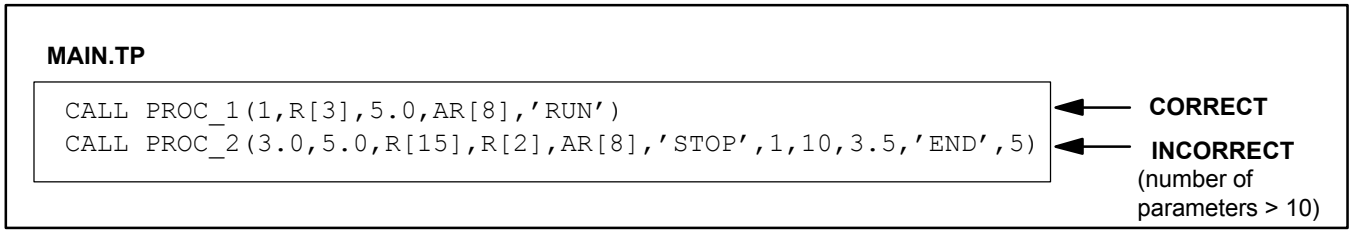
Follow the guidelines in this section to use parameters correctly and avoid errors.

Use No More than Ten Parameters in an Instruction

You can use up to ten parameters in a program call or MACRO program instruction.

See Figure 6–127 for an example.

Figure 6–127. Use No More than Ten Parameters in an Instruction



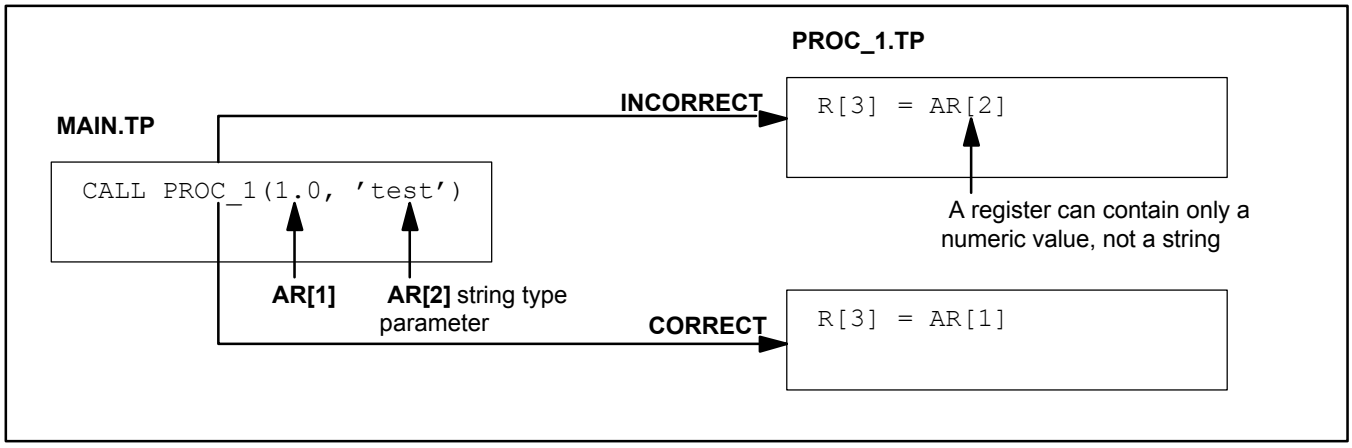
Make Sure Data Types Match

The parameter type in AR[ ] must match the data type in the sub-program or macro program instruction.

The compatibility of data types between parameters used in the main program and sub-program is not checked until the main program is executed. If the data type specified in the main program does not match how the argument register is used in the sub-program, an error will occur.

In the example shown in Figure 6–128 , if string data is stored in AR[3] as defined in the main program, when the instruction R[3] = AR[2] is executed in the sub-program, an alarm occurs.

Figure 6–128. Make Sure Data Types Match

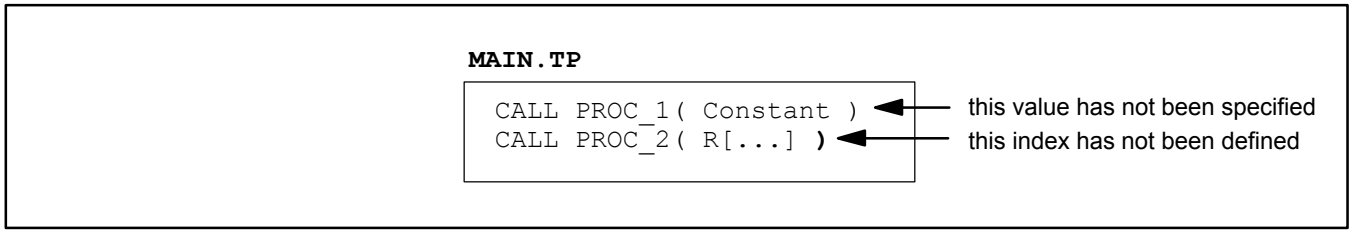


Define All Required Elements of the Parameters You Add

You must define all required elements of a parameter you add to an instruction in the main program.

Registers and argument registers require index numbers. Constants and strings require values. In the examples shown in Figure 6–129 , the constant parameter has not been specified and the register index has not been defined. When the sub-program is executed, an error will occur.

Figure 6–129. Define All Parameter Elements

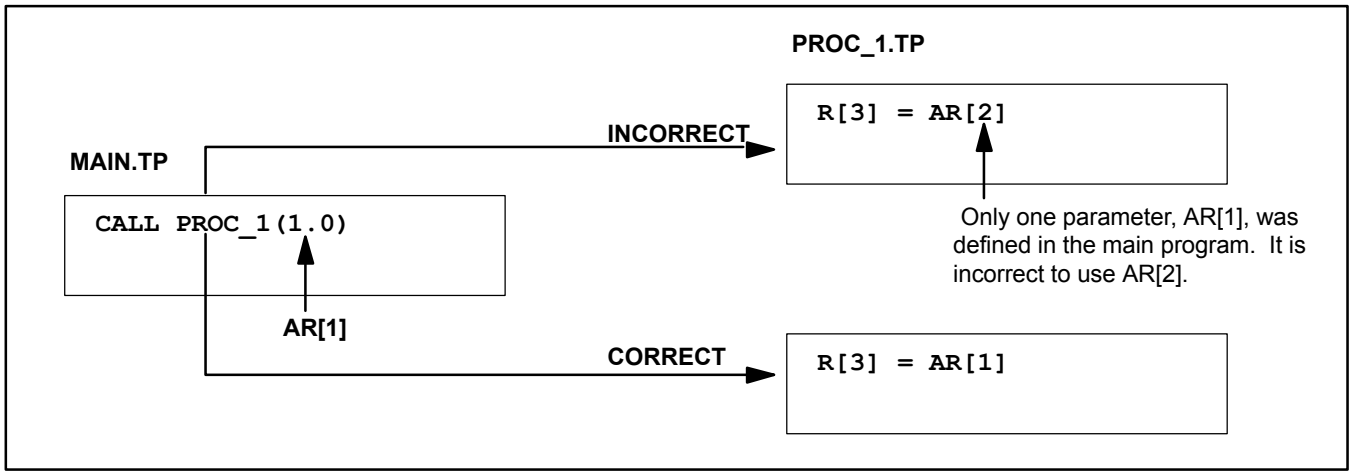


In a Sub-Program, Use Parameters Defined in the Main Program

Argument registers used in a sub-program must be defined in the corresponding main program.

In the example shown in Figure 6–130 , the program MAIN sets only one parameter, but the sub-program PROC_1 uses a second parameter (AR[2]). The sub-program cannot use a parameter that has not been defined. When that instruction is executed, an error will occur.

Figure 6–130. Use Parameters Defined in the Main Program



In a Main Program, You Can Define Parameters that are Not Used in the Sub-Program

Parameters can be defined in the main program that are not used in the sub-program.

You can use this feature to pass optional parameters. The sub-program can provide branches that process a parameter only if it exists.

Execute Sub-Programs or Macro Programs only from Main Programs

Sub-programs or MACRO programs that use argument registers can be executed from the corresponding main programs only.

You cannot execute a sub-program that uses AR[ ] values unless the sub-program is called from a main program. The main program supplies the values of the parameters used in the sub-program. If you execute the sub-program independently of the main program, the parameters will not have any values, and the following error is displayed: "INTP-288 Parameter does not exist." Therefore, the sub-program that uses the parameters cannot be executed.

6.18.6 Including Parameters in Program Call and Macro Instructions

Use Procedure 6-5 to include parameters in program call and MACRO instructions.

Procedure 6-5 Including Parameters in Program Call and MACRO Instructions

Conditions

- You have created a teach pendant program.
- This program is not a process program.

Steps

1. Select the program you want to edit.
2. Press ENTER.
3. Insert a call program or MACRO instruction.
4. **To add a parameter to an instruction that has no parameters**,
   a. Move the cursor to the program call or MACRO program instruction to which you want to add parameters.
   b. Press the right arrow key to move the cursor to the end of the instruction as shown in the following screen.

```
5: CALL PROC_1
```

   c. Press F4, [CHOICE].

```
1: R[ ]          5: <None>
2: Constant      6: <Insert>
3: String        7:
4: AR[ ]         8:
```

d. Select the kind of parameter you want to insert:

- To insert a **constant** parameter, go to Step 9 .
- To insert a **string** parameter, go to Step 10 .
- To insert an **argument register (AR[])** parameter, go to Step 11 .
- To insert a**register (R[ ])** parameter, go to Step 7 .

5. To insert a parameter in an instruction that has other parameters,

a. Move the cursor to the program call or MACRO program instruction in which you want to insert a parameter.

b. Move the cursor**to the right of** where you want to insert the parameter. See Figure 6–131 .

Figure 6–131. Cursor Position to Insert Parameters

Original Instruction	Cursor Position for Insert	Resulting Instruction
... 5: CALL PROC_1(1) ...	... 5: CALL PROC_1(1) ...	... 5: CALL PROC_1(1, 2) ...
... 5: CALL PROC_1(1, 2) ...	... 5: CALL PROC_1(1, 2) ...	... 5: CALL PROC_1(1, R[3] 2) ...

c. Press F4, [CHOICE].

```
1: R[ ]          5: <None>
2: Constant      6: <Insert>
3: String        7:
4: AR[ ]         8:
```

d. **If you are inserting a parameter between existing parameters**, select < Insert>. Otherwise, continue to Step 5 Step 5e .

e. Select the kind of parameter you want to insert:

- To insert a**constant** parameter, go to Step 9 .
- To insert a**string** parameter, go to Step 10 .
- To insert an**argument register (AR[])** parameter, go to Step 11 .
- To insert a**register (R[ ])** parameter, go to Step 7 .

6. To delete a parameter,

a. Move the cursor to the program call or MACRO program instruction in which you want to delete a parameter.

b. Move the cursor to the parameter you want to delete.

c. Press F4, [CHOICE].

```
1: R[ ]          5: <None>
2: Constant      6: <Insert>
3: String        7:
4: AR[ ]         8:
```

d. Select <None>.

Note If no parameter is set for the instruction or the cursor is on ")" on the end of the line, no parameter will be deleted and the sub-menu will be closed.

7. To insert a register (R[]) parameter,

a. Select R[ ].

b. Type the index of the register and press ENTER.

8. To insert an indirect register (R[AR[ ]]orR[R[]]) parameter,

a. Select R[ ].

b. **To change the index between an R[ ] and an AR[** , move the cursor to the register and press F3, INDIRECT, repeatedly. The index will be changed as follows:

```
R[ R[...] ] -> R[ AR[...] ] -> R[ R[...] ] -> ...
```

9. **To insert a constant parameter**,
   a. Select Constant.
   b. Type the value of the constant and press ENTER.
10. **To insert a string parameter**,
   a. Select String. See the following screen for an example.

```
MENUTEST
                                              4/22
1:  !MENUTES
2:  Clear User Page
3:
4:  Prompt Box Msg('NotAtPerch')
5:
6:  Op. Entry Menu('Chute')Select item
```

   b. Select the kind of string you want.

```
String select
1 MENUS         5 POS
2 PARTS         6 DEV
3 TOOL          7 PALT
4 WORK          8 --next page--
```

```
String select
1 GRIP          5
2 USER          6
3 PREG          7
4               8 --next page--
```

- To select a string from a list of already defined strings, move the cursor to the string you want and press F5, CHANGE. Move the cursor to the group of strings from which you want to select and press ENTER. Move the cursor to the string you want and press ENTER.

```
String select
1 NotAtPerch    5 ErrorRecov
2 IsItSafe      6 Status1
3 CycleInterup  7 Status2
4 Part Select   8 Chute
```

- To enter a string directly, press F5, String, press the appropriate function keys to type the string, and press ENTER.

c. **To change a string after you have entered one**, move the cursor to the string, press F5, CHANGE, and repeat Step 10b .

Note If you want to change the string choices that are displayed, you must set system variables. Refer to Section 6.18.3 for more information.

11. To insert an argument register AR[] parameter,

a. Select AR[ ].

b. Type the index of the argument register and press ENTER.

12. To insert an indirect argument register (AR[R[]]orAR[AR[ ]]) parameter,

a. Select AR[ ].

b. **To change between an AR[ ] and an R[ ]**, move the cursor to the AR[ ] and press F3, INDIRECT, repeatedly. The index will be changed as follows:

```
AR[ R[...] ] -> AR[ AR[...] ] -> AR[ R[...] ] -> ...
```

Note To include an AR[ ] as an indirect index, move the cursor to the index, and press F3, INDIRECT, two times.

6.18.7 Including Argument Registers in Sub-Programs

Use Procedure 6-6 to include argument registers (AR[ ]) in a sub-program.

Procedure 6-6 Including Argument Registers in a Sub-Program

Conditions

- You have created a teach pendant program that includes a program call or macro program instruction.

Steps

1. Select the sub-program or macro program you want to edit.
2. Press ENTER.
3. Insert one of the kinds of instructions that can contain an AR[ ]. Refer to Table 6–23 .

Table 6–23. Instructions that can Use AR[]

You Can Use an Argument Register...	Example
On the right side of an assignment instruction	R[1] =**AR[1]** + R[2] + **AR[4]** IF R[1] =**AR[1]** AND R[2] = **AR[4]** , JMP LBL[1]
	GO[1] =**AR[2]**
	IF R[7] =**AR[1]** , JMP LBL[1] WAIT AI[1] <>**AR[2]** , JMP LBL[1]
	UFRAME_NUM =**AR[3]** UTOOL_NUM**AR[4]**
As an indirect index in an instruction	R[**AR[1]**] = R[**AR[2]**] DO[**AR[1]**] = ON
As a parameter for program call instruction	CALL SUBPRG_1(**AR[5]**)
As a parameter for MACRO instruction	Hand 3 Release (**AR[1]**]

4. In the instruction you just inserted, move the cursor to the element you want to change to an AR[ ].
5. Press F4, [CHOICE].
6. Select AR[ ].
7. Type the index and press ENTER.
8. **To include an AR[ ] as an indirect index**, move the cursor to the index, and press F3, INDIRECT, two times.

6.19 PROGRAM CONTROL INSTRUCTIONS

6.19.1 Overview

Program control instructions direct program execution. Use these when you want areas of your program to pause, abort, resume a program, and handle errors.

Note In PaintTool, program control instructions are available only when the sub type is set to none or macro.

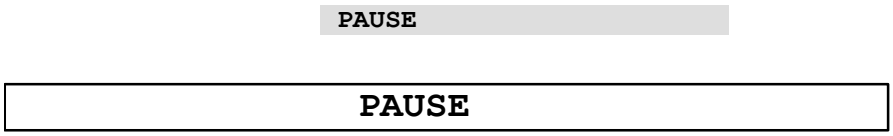
6.19.2 PAUSE Instruction

A PAUSE instruction suspends program execution in the following manner:

- Any motion already begun continues until completed.
- All connected timers continue being incremented.
- All PULSE instructions that are currently running continue to run until they are completed.
- Any instruction that is currently running, except program call instructions, is completed. Program call instructions are performed when the program is resumed.

See Figure 6–132 .

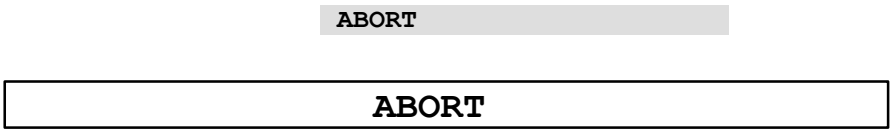
Figure 6–132. PAUSE



6.19.3 ABORT Instruction

An ABORT instruction ends the program and cancels any motion in progress or pending. After an ABORT instruction is executed, the program cannot continue; it must be restarted. See Figure 6–133 .

Figure 6–133. ABORT



6.19.4 Error Program Instruction

The error program instruction defines the program name that will be stored in the system variable $ERROR_PROG. The use of the $ERROR_PROG system variable varies depending on how your system is set up. See Figure 6–134 .

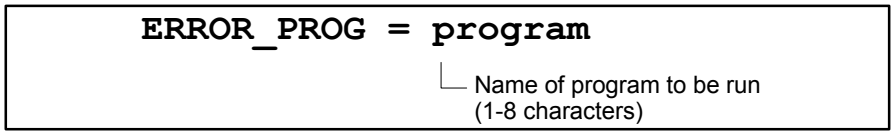
ERROR_PROG = program

You must assign a proper error recovery program name from a teach pendant program. This enables

the shell task to recover from the error more efficiently. The responsibility of the shell task is to execute the*error* program which is set in $ERROR_PROG.

Figure 6–134. Error Program

```
ERROR_PROG = ...
```



6.19.5 Resume Program Instruction

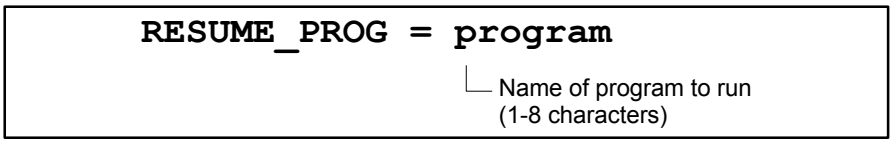
The resume program instruction defines the program name that will be stored in the system variable $RESUME_PROG. The use of the $RESUME_PROG system variable varies depending on how your system is set up. See Figure 6–135 .

(ERROR RECOVERY OPTION)

Note This program instruction is not the same as the Fast Fault Recovery resume program function.

Figure 6–135. RESUME_PROG = program

```
RESUME_PROG = ...
```



6.19.6 Maintenance Program Instruction

The maintenance program instruction defines the program name that will be used as the maintenance program, if the error recovery option is used. See Figure 6–136 .

MAINT_PROG = program

Refer to the “Advanced Functions” chapter in the Setup and Operations Manual for more information.

Figure 6–136. MAINT_PROG = program

```
MAINT_PROG = ...
```

```
MAINT_PROG = program
                └─ Name of program to run
                   (1-8 characters)
```

6.19.7 Clear Resume Program Instruction

The clear resume program instruction clears the resume program, if the error recovery option is used. See Figure 6–137 .

CLEAR_RESUME_PROG

Refer to the "Advanced Functions" chapter in the Setup and Operations Manual for more information.

Figure 6–137. CLEAR_RESUME_PRO

```
CLEAR_RESUME_PROG
```

```
CLEAR_RESUME_PROG
```

6.19.8 Return Path Disable Instruction

The return path disable instruction disables the ability to use the return path, if the error recovery option is used. See Figure 6–138 .

RETURN_PATH_DSBL

Refer to the "Advanced Functions" chapter in the Setup and Operations Manual for more information.

Figure 6–138. RETURN_PATH_DSBL

```
RETURN_PATH_DSBL
```

```
RETURN_PATH_DSBL
```

6.20 MOTION GROUP INSTRUCTIONS (OPTION)

6.20.1 Overview

Motion group instructions allow you to program multiple motion groups independently of each other.

Motion group instruction can be used to specify the:

- Motion type for individual groups (except for circular motions)
- Travel speed for individual groups
- Termination type for individual groups

There are two types of motion group instructions:

- *Independent* motion group instruction Independent GP, Section 6.20.2
- *Simultaneous* motion group instruction Simultaneous GP, Section 6.20.3

To define an Independent or Simultaneous motion group instruction in a teach pendant program, refer to Section 6.20.4 .

6.20.2 Independent Motion Group Instructions

With an Independent motion group instruction, each motion group operates with a separately taught motion type, speed, and termination type.

The following example shows independent motion for motion groups 1 and 3, with motion mask [1,*,1,*,*,*,*,*].

```
Independent GP
GP1 L P[1] 90mm/s CNT 100
GP3 J P[1] 100% CNT 50
```

6.20.3 Simultaneous Motion Group Instructions

With a simultaneous motion group instruction, each motion group operates with a separately

taught motion type, but with the same time (the longest travel time among all groups based on the programmed speeds). The termination type for the motion group with the smallest CNT value (near the FINE value) is also used for the other motion groups.

The following example program shows simultaneous motion for motion groups 1 and 3, with motion mask [1,*,1,*,*,*,*,*]. The travel time for the motion group with the longest travel time is also used for the other motion groups. The programmed travel speeds might not always be used. The termination type value for group 3 (GP3), with the smallest CNT value, is also applied to group 1 (GP1).

```
Simultaneous GP
GP1 L P[1] 90mm/s CNT 100
GP3 J P[1] 100% CNT 50
```

6.20.4 Defining Motion Group Instructions

Procedure 6-7 describes how to define an Independent or Simultaneous motion group instruction. The example program specifies motion groups 1 and 3,with motion mask [1,*,1,*,*,*,*,*].

Note You cannot define an Independent or Simultaneous motion group instruction for a circular motion statement or a control statement.

Procedure 6-7 Defining Motion Group Instructions

Conditions

- You cannot specify Independent or Simultaneous motion group instructions for circular motion instructions.

Steps

1. Move the cursor to the line number of any motion statement, except for a circular motion statement. See the following screen for an example.

```
PNS0001
  1: L P[1] 1000mm/sec Cnt100
 [END]
```

2. Press F1, [INST]. A list of control instructions appears. You will see a screen similar to the following.

```
1 .....            5 .....
2 .....            6 Independent GP
3 .....            7 Simultaneous GP
4 .....            8 -- next page --
PNS0001
  1: L P[1] 1000mm/sec Cnt100
 [END]
```

3. Select Independent GP or Simultaneous GP. The contents of line 1 in the teach pendant program is transferred to the other groups. The position data might be changed. See the following screen for an example.

```
PNS0001
  1: Independent GP
   : GP1 L P[1] 1000mm/sec Cnt100
   : GP3 L P[1] 1000mm/sec Cnt100
```

4. For a motion instruction already within a Independent or Simultaneous motion group instruction, change the motion type, speed, and termination type in the same way as for ordinary motion instructions. Refer to Section 6.4 for more information.

Note You cannot **add** motion options to an Independent or Simultaneous motion group instruction after it has been created. All motion options must be added to the Independent or Simultaneous motion group instruction before it is created. However, you can delete motion options from an Independent or Simultaneous motion group instruction.

Cautions and Restrictions

When you use the motion group instruction function, observe the following cautions and restrictions.

- If a motion statement is a circular motion, a motion group instruction cannot be specified.
- You cannot add a circular motion type to an Independent or Simultaneous motion group instruction.
- Within a motion group instruction:
  - The position data format for each group cannot be changed (position <-> position register).
  - The position number for each group cannot be changed.

— Motion options cannot be added; deletion is possible.

— A group cannot be deleted or created.

- No search can be made for elements in a motion group instruction.
- You cannot perform simultaneous teaching or deletion for the program element replacement function in a motion group instruction.
- Within a motion group instruction, SHIFT + TOUCHUP> cannot be used to correct the position.
- If an incremental instruction is deleted from a motion group instruction, the position data used in the motion group instruction will be uninitialized.

If no motion group instruction has been specified, an ordinary motion instruction is executed. It is assumed all groups are executed simultaneously with the same motion type, time, termination type, and motion options. In this case, other groups are synchronized with the group that has the longest travel time.

6.21 POSITION REGISTER LOOK-AHEAD INSTRUCTIONS

While the robot is executing a program, it reads the lines ahead of the line currently being executed (look-ahead execution). The position register look-ahead execution function enables look-ahead execution for position registers. In DispenseTool, this instruction can be used only in a job program.

The position register look-ahead execution function includes the following program instructions:

- LOCK PREG
- UNLOCK PREG

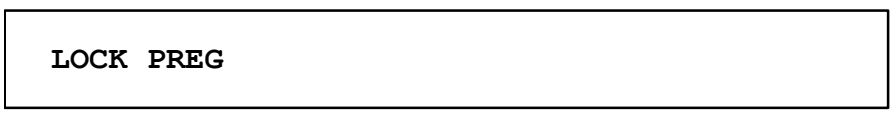
Refer to the “Advanced Functions” chapter in the Setup and Operations Manual for more detailed information on the position register look-ahead execution function.

LOCK PREG

This instruction locks all position registers. It prevents any change from being made to any position register. See Figure 6–139 .

Figure 6–139. LOCK PREG Instruction

```
LOCK PREG
```



UNLOCK PREG

This instruction unlocks the position registers. See Figure 6–140 .

Figure 6–140. UNLOCK PREG Instruction

```
UNLOCK PREG
```



6.22 CONDITION MONITOR INSTRUCTIONS

The condition monitor function monitors the condition of an I/O signal, register value, or alarm status, during teach pendant program execution. As soon as the condition is triggered, the specified teach pendant program is executed and interrupts the current program.

Condition monitor instructions are used to control the monitoring of conditions when a program is running. There are three condition monitor instructions used for program monitoring:

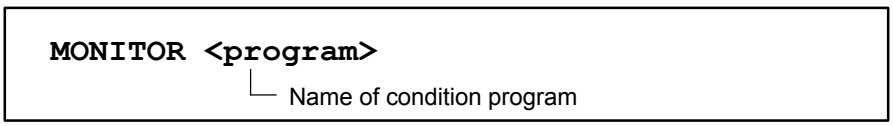
- MONITOR <program>
- MONITOR END <program>
- WHEN <condition> CALL <program>

MONITOR

This instruction starts monitoring the conditions taught in the specified condition program (ch sub type). See Figure 6–141

Figure 6–141. MONITOR Instruction

```
MONITOR <program>
```

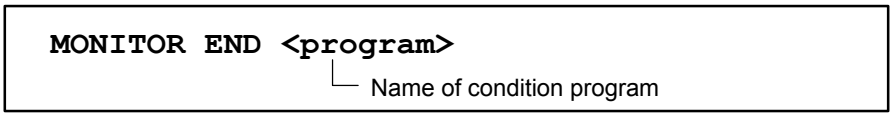


MONITOR END

This instruction stops monitoring the conditions taught in the specified condition program (ch sub type). See Figure 6–142 .

Figure 6–142. MONITOR END Instruction

```
MONITOR END <program>
```



WHEN <condition> CALL <program>

This instruction defines the conditions for which to monitor. You include WHEN instructions within your condition (ch sub-type) programs. WHEN instructions are the only instructions available when you create condition programs. See Figure 6–143 through Figure 6–146 .

In a condition handler program, you can teach multiple WHEN instructions as follows.

Multiple WHEN Instructions in a Condition Handler Program

```
1: WHEN  <cond1>  CALL  <program1>
2: WHEN  <cond2>  CALL  <program2>
3: WHEN  <cond3>  CALL  <program3>
```

You can connect the multiple conditions using AND/OR as follows.

Connecting Multiple Conditions using AND/OR

```
1: WHEN  <cond1>  AND  <cond2>  CALL  <program1>
2: WHEN  <cond1>  OR  <cond2>  OR  <cond3>  CALL <program2>
```

Note You cannot use both AND and OR in the same WHEN instruction.

Figure 6–143. Condition for Register, System Variable, and I/O Parameters

```
Example
WHEN ... = ... CALL ...
```

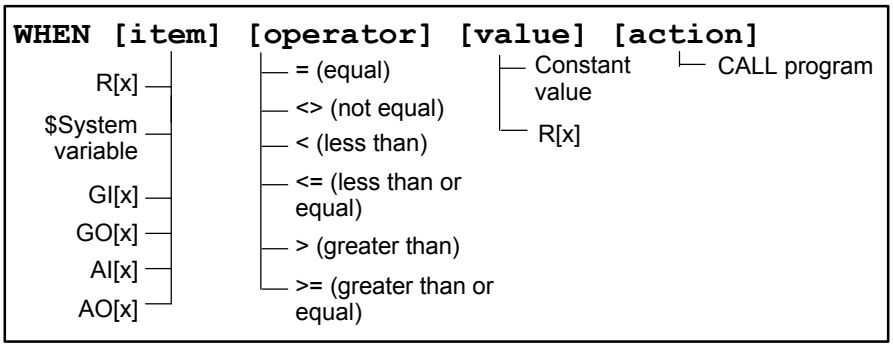


Figure 6–144. Condition2 for I/O

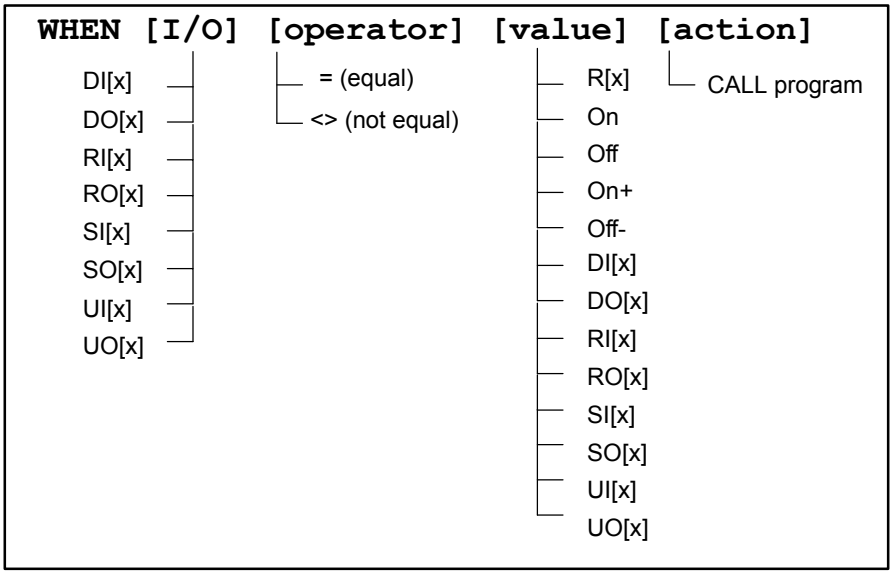


Figure 6–145. Condition2 for I/O

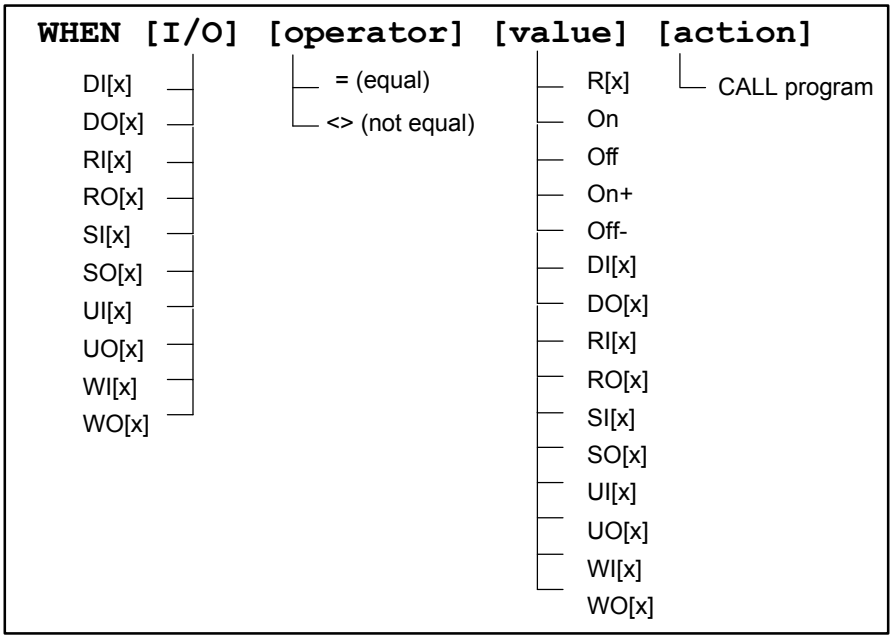
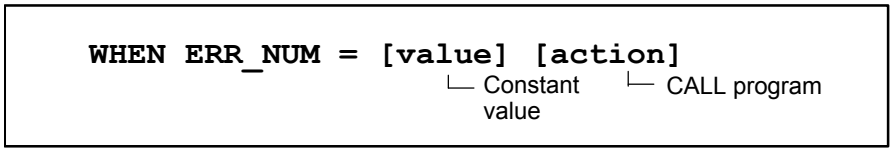


Figure 6–146. Condition for Error Status

```
Example
WHEN ... = ... CALL ...
```



ERROR NUMBER

ERR_NUM = aaabbb

aaa : Error facility code (decimal); Refer to the “Error Codes and Recovery” appendix in the Setup and Operations Manual.

.bbb : Error number (decimal)

Example: WHEN ERR_NUM=11006, CALL PROG_A

This means "SRVO-006 Hand broken" error because the SRVO facility code is 11.

If 0 is specified as error number "aaabbb," whenever any error occurs, the condition is satisfied.

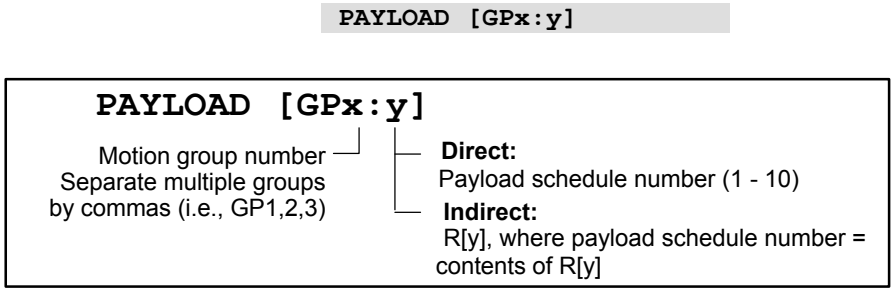
6.23 PAYLOAD INSTRUCTION

For some applications, you might need to adjust the payload several times within your teach pendant program. For example, if your application requires a change of end-of-arm tooling, you will need to adjust payload information to reflect this change.

See Figure 6–147 .

PAYLOAD [GPx:y]

Figure 6–147. Payload Instruction



The PAYLOAD[GPx:y] instruction allows you to specify the *payload schedule* to use. You can specify up to 10 different sets of payload information. Each set of payload information corresponds to a schedule number.

If the payload changes during your application, you must use the PAYLOAD[GPx:y] instruction to select the appropriate payload schedule.

Before you use a PAYLOAD[GPx:y] instruction, you must make sure you have set up the payload schedule that corresponds to the one you specify. Refer to the “General Setup” chapter in the Setup and Operations Manual for information on setting up payloads.

See Example of Using PAYLOAD[GPx:y] Instructions in a Teach Pendant Program for an example of using the PAYLOAD[GPx:y] instruction in a teach pendant program.

Example of Using PAYLOAD[GPx:y] Instructions in a Teach Pendant Program

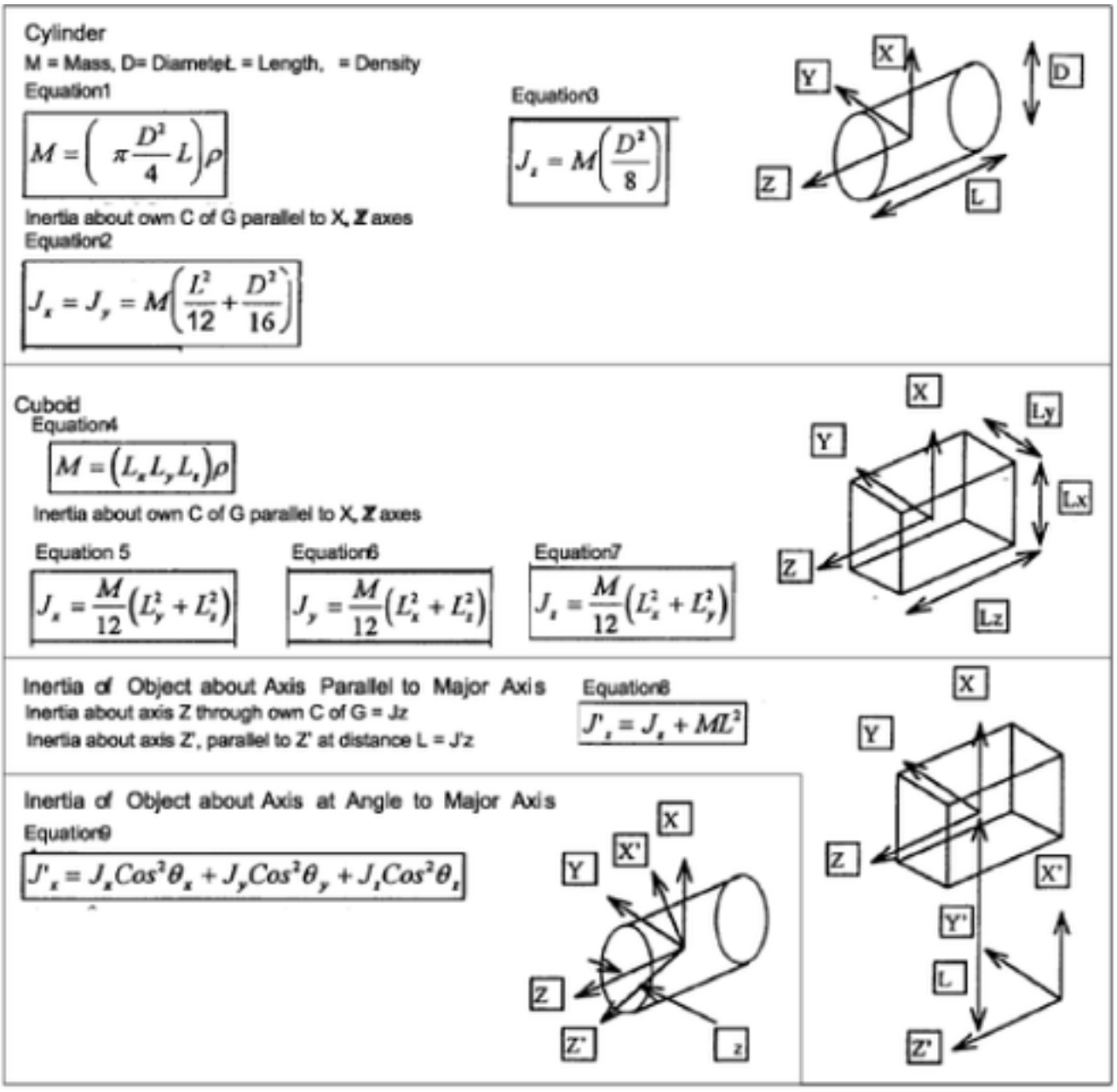
```
52:   PAYLOAD [GP1:1]
53: L P[1] 500mm/sec CNT100
54: L P[2] 2000mm/sec CNT100
55: L P[3] 500mm/sec FINE
56:   CALL toolchng
57:   PAYLOAD [GP1:2]
```

```
58: L P[2] 500mm/sec CNT100
59: L P[1] 2000mm/sec/ CNT100
60: L P[5] 500mm/sec FINE
61:   CALL toolchng
62:   PAYLOAD [GP1:1]
63: L P[1] 500mm/sec CNT100
```

Inertia Equations

Refer to Figure 6–148 for inertia equations to use in calculating inertia.

Figure 6–148. Inertia Equations



6.24 COLLISION GUARD INSTRUCTIONS (OPTION)

You can use the Collision Guard instructions to control Collision Guard during programmed motion.

COL DETECT ON

COL DETECT OFF

By default, Collision Guard is enabled.

- To**disable** Collision Guard, include the COL DETECT OFF instruction in a teach pendant program.

```
COL DETECT OFF
```

- To**enable** Collision Guard that has been disabled previously, include the COL DETECT ON instruction in a teach pendant program. Since Collision Guard is always enabled by default, you need to use the COL DETECT ON instruction only if you have previously used the COL DETECT OFF instruction.

```
COL DETECT ON
```

See Example of Collision Guard Instructions for an example of how to use these instructions in a teach pendant program.

Example of Collision Guard Instructions

```
10: J P[1] 100% FINE
11:   COL DETECT OFF
12: L P[2] 2000mm/sec CNT100
13: L P[3] 2000mm/sec CNT100
14: L P[4] 2000mm/sec CNT100
15:   COL DETECT ON
16: J P[5] 50% FINE
```

6.25 BASIC PROCESS AXES INSTRUCTIONS (OPTION)

6.25.1 Overview

The Basic Process Axes software option (R689) provides two TP instructions that can be used to set the speed for some or all properly set up process axes:

- SET ISDT SPEED
- STOP ALL ISDT

6.25.2 SET ISDT SPEED Instruction

The SET ISDT SPEED instruction, when executed, sends the specified speed values as commands directly to the specified process axes motors. The units of the speed value are user units. The relationship between the "user units" used in the SET ISDT SPEED instruction and the RPM at the motor shaft is defined by the following two quantities:

- The gear ratio specified in the Process Axes Setup routine at CONTROLLED start
- The scale factor specified in the Process Axes Setup menu.

This relationship is defined in equation .

$$speed\ (user\ units) = \frac{motor\ RPM}{gear\ ratio} * scale\ factor$$

Note Please see the Software Installation Manual for more information on setting the gear ratio for Process Axes, specifically the section entitled "Displaying, Modifying and Installing Process Axes."

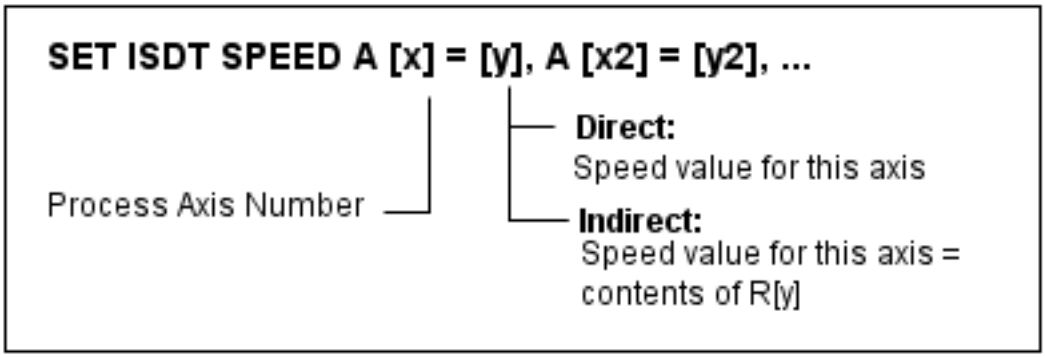
Note Please see the "Basic Process Axes" section in the "Advanced Functions" chapter of this manual for more details on how to set the scale factor in the Process Axes Setup menu.

Any properly setup process axes that are *not* specified in the SET ISDT SPEED instruction will not be affected by the instruction; they will continue operating at their current speed.

The syntax of the instruction is shown in figure Figure 6–149 .

Figure 6–149. SET ISDT SPEED A...=...





By default, when the SET ISDT SPEED instruction is first inserted, by default it has two fields, one for the axis number and one for the speed as shown in figure Figure 6–149 . Additional fields may be added by pressing F1 ADD AX. This creates an additional set of axis and speed fields as shown:

```
1: SET ISDT SPEED A...=...,A...=...
```

The cursor can then be moved to any of the fields and integer data can be entered. Additional fields can be added in this way, up to a total of eight sets of axis and speed data in a single instruction.

Note If an axis number is specified that does not correspond to a valid, properly setup process axes, then an ISDT-012 alarm will be posted at run time.

Note If the same axis number is specified in multiple fields on the same instruction line, then an ISDT-013 alarm will be posted at run time.

Note The speed value cannot be a negative number. If negative speed is desired, either indirect (i.e. numeric register) must be used for the speed value, or a negative scale factor should be set up.

6.25.3 STOP ALL ISDT Instruction

The STOP ALL ISDT instruction, when executed, will send a zero speed value to *all* process axes that are properly setup in the controller. This instruction does not have any arguments.

6.26 PALLETIZING INSTRUCTIONS

6.26.1 Overview

Palletizing instructions tell the robot when and how to palletize. These palletizing instructions allow you to either stack the pallet or unload the pallet. There are four kinds of palletizing instructions:

- PALLETIZING-B
- PALLETIZING-BX
- PALLETIZING-E
- PALLETIZING-EX

The palletizing instruction deals only with the actual indexing of the area of the program immediately close to the pallet. You must include any required I/O, other motions, and additional logic instructions.

To use palletizing instructions you:

1. Select one of the palletizing instructions.
2. Use the Pallet Editor to specify how the instruction will work in your program. The Pallet Editor appears only when entering or modifying a palletizing instruction.
3. Create the stack pattern by recording certain robot positions.
4. Create the route with which the robot moves to the pallet by recording certain robot positions.

Palletizing Program Example shows an example of a palletizing program.

Palletizing Program Example

```
1: R[1]=0
 2: PL[1]=[1,1,1]
 3: PL[2]=[1,1,1]
 4: !THE NEXT LINE OPENS THE GRIPPER
 5: RDO[1]=ON
 6: RDO[2]=OFF
 7: LBL[1]
 8: IF SDI[4]=ON, CALL BALLET
 9: J P[5] 100% CNT100
10: R[1]=R[1]+1
11: PR[3]=PR[4]
12: IF R[1]<6, JMP LBL[10]
13: IF R[1]=7, JMP LBL[10]
14: PR[3]=PR[2]
15: LBL[10]
16: PALLETIZING-B_1
```

```
17: J PAL_1[A_1] 100% FINE Offset,PR[1]
18: J PAL_1[BTM] 30%  FINE Offset,PR[1]
19: !CLOSE THE GRIPPER
20: RDO[1]=OFF
21: RDO[2]=ON
22: WAIT .50 (sec)
23: J PAL_1[R_1] 30% FINE Offset,PR[1]
24: PALLETIZING-END-1
25: J P[5] 100% CNT100
26: J P[1] 100% CNT100
27: WAIT SDI[10] =OFF
28: J P[2] 100% FINE
29: L P[4] 100 mm/sec FINE
30: !THE NEXT LINE OPENS THE GRIPPER
31: RDO[1]=ON
32: RDO[2]=OFF
33: WAIT .50 (sec)
34: L P[2] 100mm/sec FINE
35: WAIT SDI[11]=ON
36: J P[6] 100% CNT100
37: L P[9] 100mm/sec CNT100
38: L P[7] 100mm/sec FINE
39: !THE NEXT LINE CLOSES THE GRIPPER
40: RDO[1]=OFF
41: RDO[2]=ON
42: WAIT .50 (sec)
43: L P[7] 100mm/sec FINE
44: L P[9] 100mm/sec CNT100
45: J P[[6] 100% CNT100
46: J P[[1] 100% CNT100
47: J P[[5] 100% CNT100
48: J P[[12] 100% CNT100
49: L P[[10] 100mm/sec CNT100
50: J P[[11] 100% CNT100
51: L P[13]  100mm/sec CNT50
52: PALLETIZING-B_2
53: J PAL_2[A_1] 100% FINE Offset,PR[3]
54: J PAL_2[BTM] 10% FINE Offset,PR[3]
55: !THE NEXT LINE CLOSES THE GRIPPER
56: RDO[1]=OFF
57: RDO[2]=ON
58: WAIT .50(sec)
59: J PAL_2[R_1] 30% FINE Offset,PR[3]
61: IF R[1]<>9,JMP LBL[1]
62: R[1]=0
63: PL[1]=[1,1,1]
64: PL[2]=[1,1,1]
```

```
 65: LBL[2]
 66: IF SDI[4]=ON, CALL BALLET
 67: J P[5] 100% CNT100
 68: R[1]=R[1]+1
 69: PR[3]=PR[4]
 70: IF R[1]<6, JMP LBL[20]
 71: IF R[1]=7, JMP LBL[20]
 72: PR[3]=PR[2]
 73: LBL[20]
 74: PALLETIZING-B_2
 75: J PAL_2[A_1] 100% FINE Offset,PR[3]
 76: J PAL_2[BTM] 30%  FINE Offset,PR[3]
 77: CLOSE THE GRIPPER
 78: RDO[1]=OFF
 79: RDO[2]=ON
 80: WAIT .50 (sec)
 81: J PAL_2[R_1] 30% FINE Offset,PR[3]
 82: PALLETIZING-END_2
 83: J P[5] 100% CNT100
 84: J P[1] 100% CNT100
 85: WAIT SDI[10] =OFF
 86: J P[2] 100% FINE
 87: L P[4] 100 mm/sec FINE
 88: !THE NEXT LINE OPENS THE GRIPPER
 89: RDO[1]=ON
 90: RDO[2]=OFF
 91: WAIT .50 (sec)
 92: L P[2] 100mm/sec FINE
 93: WAIT SDI[11]=ON
 94: J P[6] 100% CNT100
 95: L P[9] 100mm/sec CNT100
 96: L P[7] 100mm/sec FINE
 97: !THE NEXT LINE CLOSES THE GRIPPER
 98: RDO[1]=OFF
 99: RDO[2]=ON
100: WAIT .50 (sec)
101: L P[8] 100mm/sec FINE
102: L P[9] 100mm/sec
103: J P[6] 100% CNT100
104: J P[1] 100% CNT100
105: J P[5] 100% CNT100
106: J P[12] 100% CNT100
107: L P[10] 100mm/sec CNT100
108: J P[11] 100% CNT100
109: L P[13] 100mm/sec CNT50
110: PALLETIZING-B_1
111: J PAL_1[A_1] 100% FINE Offset,PR[1]
```

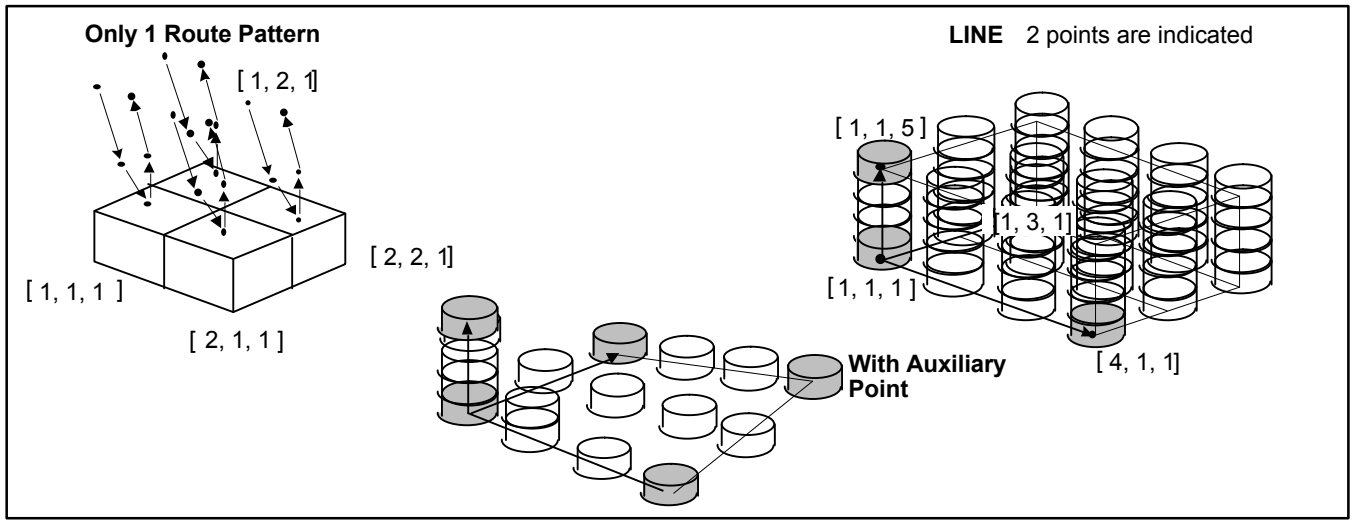
```
112: J_PAL_1[BTM] 30% FINE Offset,PR[1]
113: !THE NEXT LINE OPENS THE GRIPPER
114: RDO[1]=ON
115: RDO[2]=OFF
116: WAIT .50(sec)
117: J PAL_1[R_1] 30% FINE Offset,PR[1]
118: PALLETIZING-END_1
119: IF R[1]<>9,JMP LBL[2]
120: R[1]=0
121: JMP LBL[1]
```

6.26.2 PALLETIZING-B Instruction

The PALLETIZING-B instruction is the basic kind of palletizing instruction. You use this instruction when:

- The approach and retreat routes of your robot always have the same direction and orientation. The route is the number of robot positions you record to move your robot to the pallet and then away from the pallet. See Figure 6–150 .
- Your stacking pattern consists of a line pattern. A line pattern has columns, rows, and layers placed vertically, horizontally, and perpendicularly. For the line pattern, you record three end positions that indicate where the rows, columns, and layers are located. In Figure 6–150 , [1,1,1] is the last route position recorded, [1,1,5], [1,3,1] and [4,1,1] are the three recorded end positions.

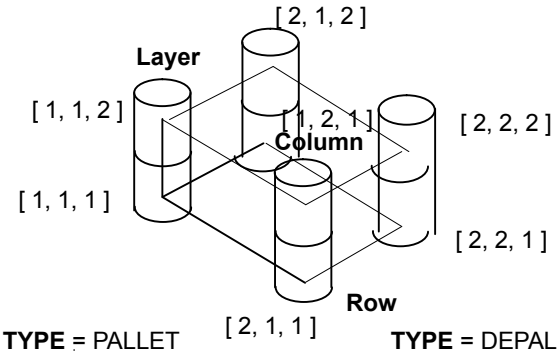
Figure 6–150. Example of PALLETIZING-B Instruction



Note Palletizing or depalletizing [1,1,1] is always on the bottom of the pallet on the first layer.

Table 6–24 lists and describes each PALLETIZING-B instruction item you can specify in the Pallet Editor.

Table 6–24. PALLETIZING-B Pallet Editor Items

<table>
<tr><th>ITEM</th><th>DESCRIPTION</th></tr>
<tr><td>Comment</td><td>This item allows you to enter a comment about this palletizing instruction.</td></tr>
<tr><td>Type</td><td>This item allows you to specify whether this palletizing instruction will palletize or depalletize. If set to PALLET, the robot stacks the pallet from bottom to top. If set to DEPALLET, the robot unloads the pallet from top to bottom.</td></tr>
<tr><td>Incr</td><td>This item allows you to specify how the pallet register will increment or decrement. This value, along with the specified order, controls which position will be processed by the robot next. The value of the pallet register indicates the next position to be processed. The order indicates which portion of the pallet register is updated first, second, and last. In the example, the specified order is row, column, layer. For palletizing, when Incr is set to 1, the instruction initializes the pallet register as the first row, first column, first layer. When Incr is set to —1 the instruction initializes the pallet register as the last row, last column, and first layer.
For depalletizing, when Incr is set to 1, the instruction initializes the pallet register as the last row, last column, last layer. When Incr is set to —1 the instruction initializes the pallet register as the first row, first column, and last layer.
 

TYPE = PALLET
<table>
<tr><td></td><td>INCR = 1</td><td>INCR = -1</td></tr>
<tr><td></td><td colspan="2">Order = RCL</td></tr>
<tr><td>Initial</td><td>[1, 1, 1]</td><td>[2, 2, 1]</td></tr>
<tr><td rowspan="8">↓</td><td>[2, 1, 1]</td><td>[1, 2, 1]</td></tr>
<tr><td>[1, 2, 1]</td><td>[2, 1, 1]</td></tr>
<tr><td>[2, 2, 1]</td><td>[1, 1, 1]</td></tr>
<tr><td>[1, 1, 2]</td><td>[2, 2, 2]</td></tr>
<tr><td>[2, 1, 2]</td><td>[1, 2, 1]</td></tr>
<tr><td>[1, 2, 2]</td><td>[2, 1, 1]</td></tr>
<tr><td>[2, 2, 2]</td><td>[1, 1, 2]</td></tr>
<tr><td>[1, 1, 1]</td><td>[2, 2, 1]</td></tr>
</table>
TYPE = DEPALLET
<table>
<tr><td></td><td>INCR = 1</td><td>INCR = -1</td></tr>
<tr><td></td><td colspan="2">Order = RCL</td></tr>
<tr><td>Initial</td><td>[2, 2, 2]</td><td>[1, 1, 2]</td></tr>
<tr><td rowspan="8">↓</td><td>[1, 2, 2]</td><td>[2, 1, 2]</td></tr>
<tr><td>[2, 1, 2]</td><td>[1, 2, 2]</td></tr>
<tr><td>[1, 1, 2]</td><td>[2, 2, 2]</td></tr>
<tr><td>[2, 2, 1]</td><td>[1, 1, 2]</td></tr>
<tr><td>[1, 2, 1]</td><td>[2, 1, 1]</td></tr>
<tr><td>[2, 1, 1]</td><td>[1, 2, 1]</td></tr>
<tr><td>[1, 1, 1]</td><td>[2, 2, 1]</td></tr>
<tr><td>[2, 2, 2]</td><td>[1, 1, 2]</td></tr>
</table>
</td></tr>
</table>

Table 6–24. PALLETIZING-B Pallet Editor Items (Cont'd)

Pal Reg	This item allows you to specify the pallet register that will be used to store the next position to be processed.
Order	This item allows you to specify the order in which the robot will palletize or depalletize. The available options are CRL, CLR, RCL, RLC, or LCR where • C = Column • R = Row • L = Layer
Columns	This item allows you to specify the total number of columns in the stack.
Rows	This item allows you to specify the total number of rows in the stack.
Layers	This item allows you to specify the total number of layers in the stack.
Auxiliary Pos	This item allows you to specify whether or not an auxiliary position will be used. A value of YES will keep the pallet positions that are away from the [1,1,1] corner compact. A value of NO can cause spreading.
APPR	This item allows you to specify how many approach route positions to use. The number you specify will be the number of positions created by the instruction for you to record.
RTRT	This item allows you to specify how many retreat route positions to use in the palletizing instruction. The number you specify will be the number of positions created by the instruction for you to record.

Procedure 6-8 Using the Palletizing Editor To Enter the Palletizing-B Instruction

Conditions

- You have created a program.
- You are editing a program.

Steps

1. **Select the instruction:**
   a. Press F1, [INST].
   b. Select Palletizing. See the following screen for an example.

```
PALLETIZING statem
                      5 PALLETIZING-END
2 PALLETIZING-BX      6
3 PALLETIZING-E       7
4 PALLETIZING-EX      8
Main Pallet
    [END]
Select item
```

c. Select PALLETIZING-B. The Pallet Editor will be displayed. See following screen for an example.

```
Main Pallet
PALLETIZING CONFIGURATION
PALETIZING_1  [                    ]
TYPE = [PALLET]          INCR = [  1]
PAL REG = [  1]         ORDER = [CRL]
COLUMN  = [          1]
ROWS    = [          1]
LAYERS  = [          1]
AUXILIARY POS = [NO]
APPR = [  1]   RTRT = [  1]
```

Note The term Palletizing_1 in the example screen indicates that this is the first palletizing instruction in the program.

2. To return to your program at any time, press F1, PROG.

```
Quit pallet editor?
```

a. Press F4, YES, to return to your program.

b. Press F5, NO, to remain in the Pallet Editor.

3. To set up the palletizing configuration:

a. Enter a Comment.

- Press ENTER.
- Move the cursor to select a method of naming the program.

For example, if you chose Alphabetic, press a function key corresponding to the first letter. Press that key until the letter you want is displayed in the comment field. Press the right arrow key to move the cursor to the next space. Continue until the entire comment is displayed. **To delete a character**, press BACK SPACE.

- When you are finished, press ENTER.

b. Select the TYPE.

- **To palletize**, press F2, PALLET.
- **To depalletize**, press F3, DEPALL.

c. Type the increment value.

d. Type the number of the pallet register you want to use.

e. Type the order. Available orders include: CRL, CLR, RCL, RLC, and LCR.

- **To select rows**, press F2, R.
- **To select columns**, press F3, C.
- **To select layers**, press F4, L.

f. Type the total number of columns.

g. Type the total number of rows.

h. Type the total number of layers.

i. Select the auxiliary position.

- To use an auxiliary position, press F2, YES.
- To not use an auxiliary position, press F3, NO.

j. Type the number of approach positions.

k. Type the number of retreat positions.

l. When you are finished setting up the PALLETIZING B instruction, press F5, DONE. See the following screen for an example.

```
Main Pallet
TEACH BOTTOM POINTS
  1: *P[1,1,1]
  2: *P[10,1,1]
  3: *P[1,2,1]
  4: *P[1,1,2]
```

Note The number of bottom positions you record depends on the number of columns, rows, and layers you entered in the Pallet Editor. In the above screen example, 10 columns, 2 rows, and 2 layers were entered. The order was entered as CRL.

4. **To return to your program at any time**, press >, and then press F1, PROG.

```
Quit pallet editor?
```

   a. Press F4, YES, to return to your program.

   b. Press F5, NO, to remain in the Pallet Editor.

5. **To return to the Pallet Editor at any time**, press F1, BACK.

6. **To record each pallet position**:

   a. Jog the robot to the position.

   b. Hold down the SHIFT key and press F4, RECORD.

   c. When you are finished recording positions, press F5, DONE. See the following screen for an example.

```
Main Pallet
PALLETIZING ROUTE POINTS
    IF PL[1] = [*,*,*}
1: J PAL_1[A_2] 30% FINE
2: J PAL_1[A_1] 30% FINE
3: J PAL_1[BTM] 30% FINE
4: J PAL_1[R_1] 30% FINE
5: J PAL_1[R_2] 30% FINE
```

Note The number of route positions you record depends on the number of approach and retreat positions you entered in the Pallet Editor. In the above example, 2 approach and 2 retreat positions were entered.

7. **To return to your program at any time**, press >, and then press F1, PROG.

```
Quit pallet editor?
```

   a. Press F4, YES, to return to your program.

   b. Press F5, NO, to remain in the Pallet Editor.

8. **To return to the Pallet Editor at any time**, press F1, BACK.

9. **To set up position default information**, press F2, POINT.
10. **To record each route position**:
    a. Jog the robot to the position.
    b. Hold down the SHIFT key and press F4, RECORD.
    c. When you are finished recording positions, press F5, DONE. You are returned to your program and the PALLETIZING B instruction has been inserted.

6.26.3 PALLETIZING-BX Instruction

The PALLETIZING-BX instruction is a basic kind of palletizing instruction that allows you to specify multiple route patterns. You use this instruction when:

- The approach and retreat routes of your robot have different directions and orientations. The route is the number of robot positions your record to move your robot to the pallet and then away from the pallet. See Figure 6–151 .
- Your stacking pattern consists of a line pattern. A line pattern has columns, rows, and layers placed vertically, horizontally, and perpendicularly. For the line pattern, you record three end positions that indicate where the rows, columns, and layers are located. In Figure 6–151 , [1,1,1] is the last route position recorded, [1,1,5], [1,3,1] and [4,1,1] are the three recorded end positions.

Figure 6–151. Example of PALLETIZING-BX Instruction

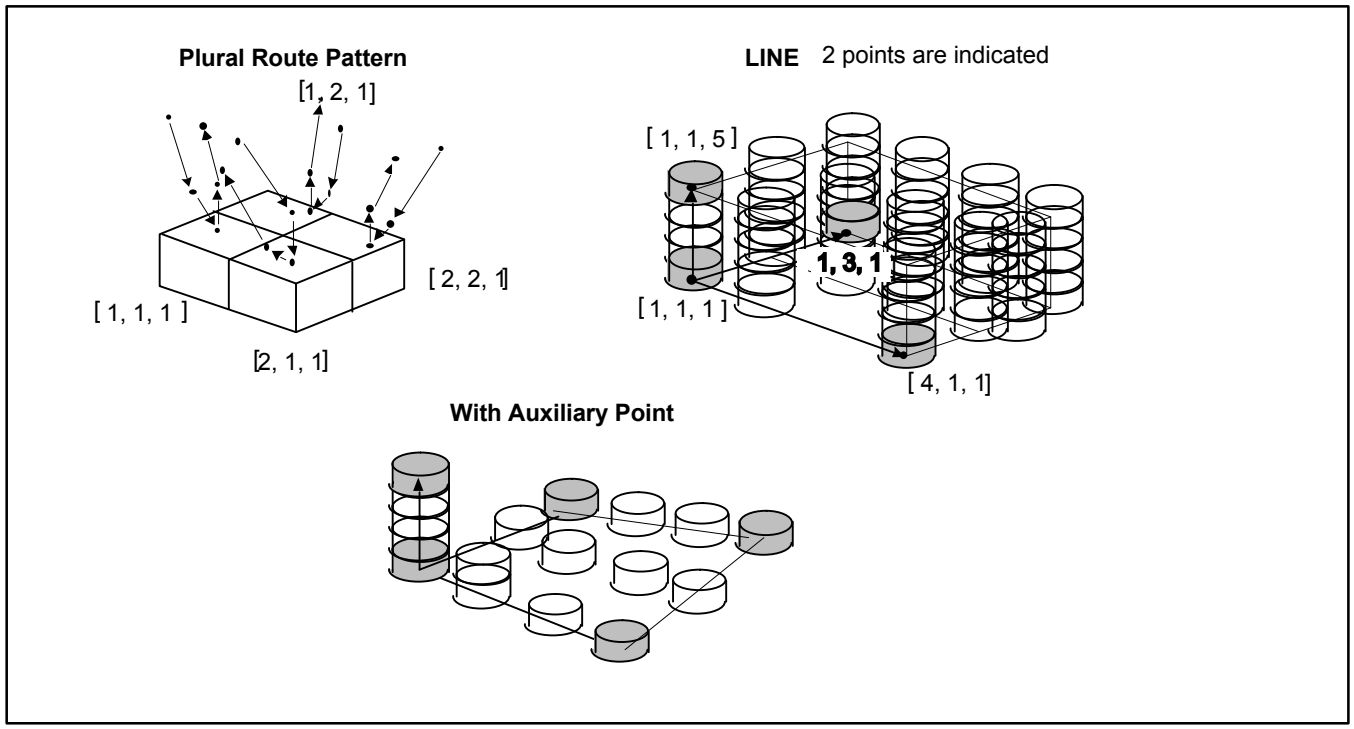
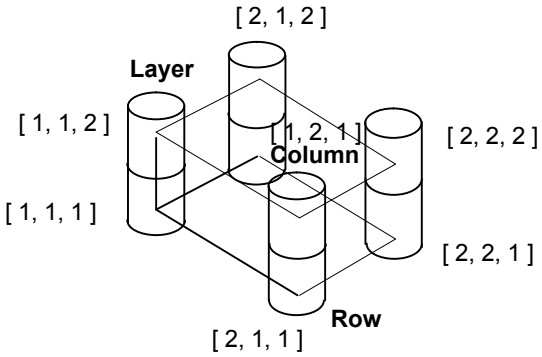


Table 6–25 lists and describes each PALLETIZING-BX instruction item you can specify in the Pallet Editor.

Table 6–25. PALLETIZING-BX Pallet Editor Items

ITEM	DESCRIPTION
Comment	This item allows you to enter a comment about this palletizing instruction.
Type	This item allows you to specify whether this palletizing instruction will palletize or depalletize. If set to **PALLET**, the robot stacks the pallet from bottom to top. If set to **DEPALLET**, the robot unloads the pallet from top to bottom.

Table 6–25. PALLETIZING-BX Pallet Editor Items (Cont'd)

<table>
<tr><td>Incr</td><td>This item allows you to specify how the pallet register will increment or decrement. This value, along with the specified order, controls which position will be processed by the robot next. The value of the pallet register indicates the next position to be processed. The order indicates which portion of the pallet register is updated first, second, and last. In the example, the specified order is row, column, layer. For palletizing, when Incr is set to 1, the instruction initializes the pallet register as the last row, last column, and first layer.
For depalletizing, when Incr is set to 1, the instruction initializes the pallet register as the last row, last column, last layer. When Incr is set to —1 the instruction initializes the pallet register as the first row, first column, and last layer.
 

TYPE = PALLET
<table>
<tr><td></td><td>INCR = 1</td><td>INCR = - 1</td></tr>
<tr><td></td><td colspan="2">order = RCL</td></tr>
<tr><td>Initial</td><td>[1, 1, 1]</td><td>[2, 2, 1]</td></tr>
<tr><td rowspan="8">↓</td><td>[2, 1, 1]</td><td>[1, 2, 1]</td></tr>
<tr><td>[1, 2, 1]</td><td>[2, 1, 1]</td></tr>
<tr><td>[2, 2, 1]</td><td>[1, 1, 1]</td></tr>
<tr><td>[1, 1, 2]</td><td>[2, 2, 2]</td></tr>
<tr><td>[2, 1, 2]</td><td>[1, 2, 1]</td></tr>
<tr><td>[1, 2, 2]</td><td>[2, 1, 1]</td></tr>
<tr><td>[2, 2, 2]</td><td>[1, 1, 2]</td></tr>
<tr><td>[1, 1, 1]</td><td>[2, 2, 1]</td></tr>
</table>
TYPE = DEPALLET
<table>
<tr><td></td><td>INCR = 1</td><td>INCR = - 1</td></tr>
<tr><td></td><td colspan="2">order = RCL</td></tr>
<tr><td>Initial</td><td>[2, 2, 2]</td><td>[1, 1, 2]</td></tr>
<tr><td rowspan="8">↓</td><td>[1, 2, 2]</td><td>[2, 1, 2]</td></tr>
<tr><td>[2, 1, 2]</td><td>[1, 2, 2]</td></tr>
<tr><td>[1, 1, 2]</td><td>[2, 2, 2]</td></tr>
<tr><td>[2, 2, 1]</td><td>[1, 1, 2]</td></tr>
<tr><td>[1, 2, 1]</td><td>[2, 1, 1]</td></tr>
<tr><td>[2, 1, 1]</td><td>[1, 2, 1]</td></tr>
<tr><td>[1, 1, 1]</td><td>[2, 2, 1]</td></tr>
<tr><td>[2, 2, 2]</td><td>[1, 1, 2]</td></tr>
</table>
</td></tr>
<tr><td>Pal Reg</td><td>This item allows you to specify the pallet register that will be used to store the next position to be processed.</td></tr>
</table>

Table 6–25. PALLETIZING-BX Pallet Editor Items (Cont'd)

Order	This item allows you to specify the order in which the robot will palletize or depalletize. The available options are CRL, CLR, RCL, or RLC where • C = Column • R = Row • L = Layer
Columns	This item allows you to specify the total number of columns in the stack.
Rows	This item allows you to specify the total number of rows in the stack.
Layers	This item allows you to specify the total number of layers in the stack.
Auxiliary Pos	This item allows you to specify whether or not an auxiliary position will be used.
APPR	This item allows you to specify how many approach route positions to use. The number you specify will be the number of positions created by the instruction for you to record.
RTRT	This item allows you to specify how many retreat route positions to use in the palletizing instruction. The number you specify will be the number of positions created by the instruction for you to record.
PATTERN	This item allows you to specify the number of approach and retreat routes to use. These routes patterns can be **DIRECT** or **MODULO** . A direct pattern uses the same multiple routes for each layer. A modulo pattern allows you to flip the pattern between layers.

Procedure 6-9 Using the Palletizing Editor To Enter the Palletizing-BX Instruction

Conditions

- You have created a program.
- You are editing a program.

Steps

1. **Select the instruction:**
   a. Press F1, [INST].
   b. Select Palletizing. See the following screen for an example.

```
PALLETIZING statem
1 PALLETIZING-B       5 PALLETIZING-END
2 PALLETIZING-BX      6
3 PALLETIZING-E       7
4 PALLETIZING-EX      8
Main Pallet
   [END]
Select item
```

c. Select PALLETIZING- BX. The Pallet Editor will be displayed. See following screen for an example.

```
Main Pallet
  PALLETIZING CONFIGURATION
  PALETIZING_1  [                    ]
  TYPE = [PALLET]          INCR = [  1]
  PAL REG = [  1]         ORDER = [CRL]
  COLUMN  = [          1]
  ROWS    = [          1]
  LAYERS  = [          1]
  AUXILIARY POS = [NO]
  APPR=[  1] RTRT=[  1] PATTERN=[  1]
```

Note The term Palletizing_1 in the example screen indicates that this is the first palletizing instruction in the program.

2. To return to your program at any time , press F1, PROG.

```
Quit pallet editor?
```

a. Press F4, YES, to return to your program.

b. Press F5, NO, to remain in the Pallet Editor.

3. To set up the palletizing configuration,

a. Enter a Comment.

- Press ENTER.
- Move the cursor to select a method of naming the program.

For example, if you chose Alphabetic, press a function key corresponding to the first letter. Press that key until the letter you want is displayed in the comment field. Press the right arrow key to move the cursor to the next space. Continue until the entire program name is displayed. **To delete a character**, press BACK SPACE.

- When you are finished, press ENTER.

b. Select the TYPE.

- **To palletize**, press F2, PALLET.
- **To depalletize**, press F3, DEPALL.

c. Type the increment value.

d. Type the number of the pallet register you want to use.

e. Type the order. Available orders include: CRL, CLR, RCL, RLC.

- **To select rows**, press F2, R.
- **To select columns**, press F3, C.
- **To select layers**, press F4, L.

f. Type the total number of columns.

g. Type the total number of rows.

h. Type the total number of layers.

i. Select the auxiliary position.

- To use an auxiliary position, press F2, YES.
- To not use an auxiliary position, press F3, NO.

j. Type the number of approach positions.

k. Type the number of retreat positions.

l. Type the number of approach and retreat patterns you want to use.

m. When you are finished setting up the PALLETIZING BX instruction, press F5, DONE. See the following screen for an example.

```
Main Pallet
TEACH BOTTOM POINTS
 1: *P[1,1,1]
 2: *P[10,1,1]
 3: *P[1,2,1]
 4: *P[1,1,2]
```

Note The number of bottom positions you record depends on the number of columns, rows, and layers you entered in the Pallet Editor. In the above screen example, 10 columns, 2 rows, and 2 layers were entered. The order was entered as CRL.

4. **To return to your program at any time**, press >, and then press F1, PROG.

```
Quit pallet editor?
```

   a. Press F4, YES, to return to your program.

   b. Press F5, NO, to remain in the Pallet Editor.

5. **To return to the Pallet Editor at any time**, press F1, BACK.

6. **To record each pallet position**:

   a. Jog the robot to the position.

   b. Hold down the SHIFT key and press F4, RECORD.

   c. When you are finished recording positions, press F5, DONE. See the following screen for an example.

```
Main Pallet
PALLETIZING ROUTE PATTERNS
  PTN [ 1] = [  *,  *,  *]
  PTN [ 2] = [  *,  *,  *]
```

7. **To return to your program at any time**, press >, and then press F1, PROG.

```
Quit pallet editor?
```

   a. Press F4, YES, to return to your program.

   b. Press F5, NO, to remain in the Pallet Editor.

8. **To return to the Pallet Editor at any time**, press F1, BACK.

9. **To set up each route pattern**,

   a. Move the cursor to each asterisk.

   b. **To not alternate the patterns between layers**, press F2, DIRECT.

   c. **To alternate the patterns between layers**, press F3, MODULO.

d. Type the position number for each asterisk and press ENTER.

e. When you are finished recording positions, press F5, DONE. See the following screen for an example.

```
Main Pallet
PALLETIZING ROUTE POINTS
    IF PL[1] = [*,*,*}
1: J PAL_1[A_2] 30% FINE
2: J PAL_1[A_1] 30% FINE
3: J PAL_1[BTM] 30% FINE
4: J PAL_1[R_1] 30% FINE
5: J PAL_1[R_2] 30% FINE
```

Note The number of route positions you record depends on the number of approach and retreat positions you entered in the Pallet Editor. In the above example, 2 approach and 2 retreat positions were entered.

10. To return to your program at any time, press >, and then press F1, PROG.

```
Quit pallet editor?
```

a. Press F4, YES, to return to your program.

b. Press F5, NO, to remain in the Pallet Editor.

11. To return to the Pallet Editor at any time, press F1, BACK.

12. To set up position default information, press F2, POINT.

13. To record each route position:

a. Jog the robot to the position.

b. Hold down the SHIFT key and press F4, RECORD.

c. When you are finished recording positions, press F5, DONE. You are returned to your program and the PALLETIZING BX instruction has been inserted.

6.26.4 PALLETIZING-E Instruction

The PALLETIZING-E instruction is an extended kind of palletizing instruction that allows you to specify different kind of stacking arrangements. You use this instruction when:

- The approach and retreat route of your robot always has the same direction and orientation. The route is the number of robot positions your record to move your robot to the pallet and then away from the pallet. See Figure 6–152 .
- Your stacking pattern consists of a line pattern. A line pattern has columns, rows, and layers placed vertically, horizontally, and perpendicularly. For the line pattern, you record three end positions that indicate where the rows, columns, and layers are located. In Figure 6–152 , [1,1,1] is the last route position recorded, [1,1,5] [1,3,1] and [4,1,1] are the three recorded end positions.
- Your stacking pattern consists of a free pattern. A free pattern has columns, rows and layers placed vertically, horizontally and perpendicularly. For the free pattern, you record free positions where the rows, columns, and layers are located. In Figure 6–152 , [1,1,1] is the last route position recorded, [1,1,5] indicates the position of the top layer, and [2,1,1], [3,1,1], and [4,1,1] indicate the position of each row.
- You want to use an auxiliary position to define the direction of the pattern.
- You want to define the orientation of the tool at the pick/place point.

Figure 6–152. Example of PALLETIZING-E Instruction

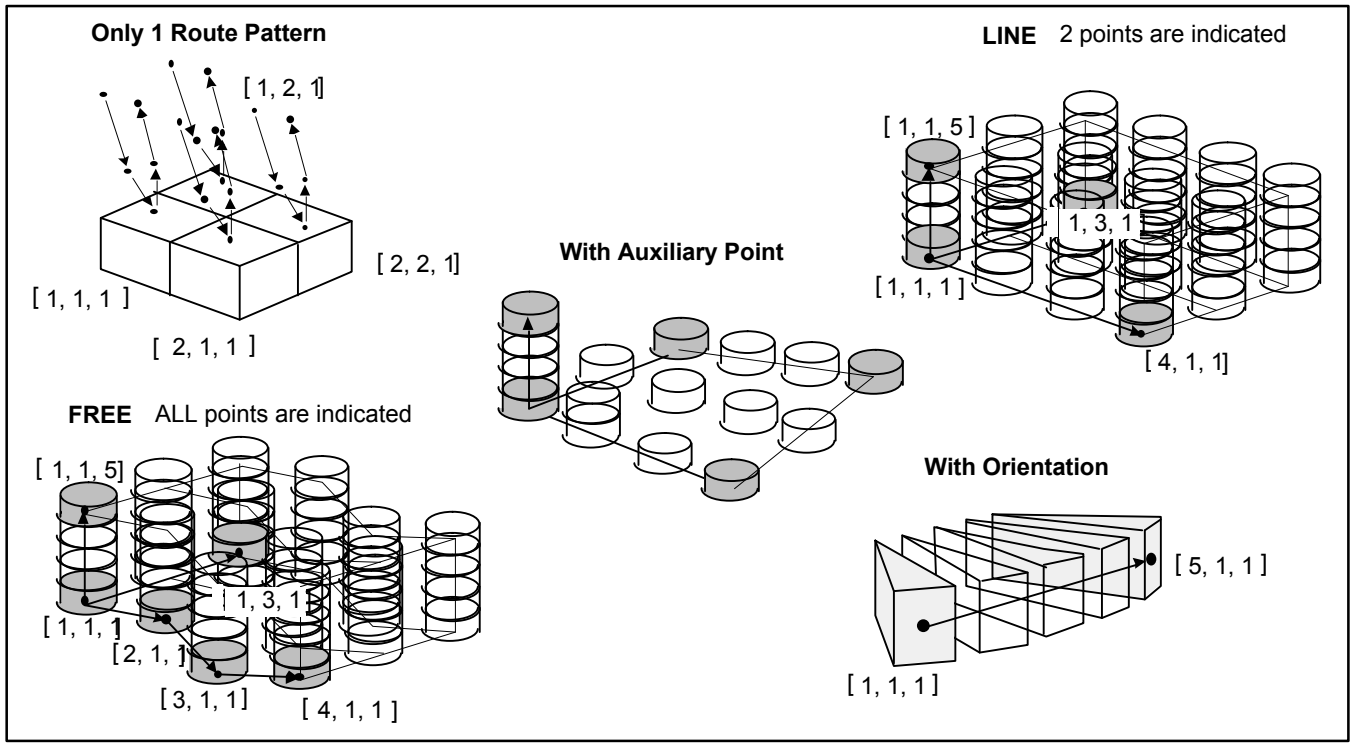
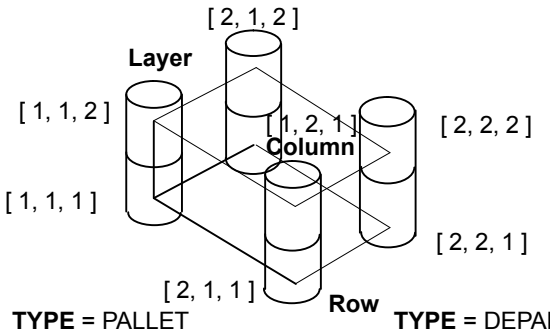


Table 6–26 lists and describes each PALLETIZING-E instruction item you can specify in the Pallet Editor.

Table 6–26. PALLETIZINGE Pallet Editor Items

<table>
<tr><th>ITEM</th><th>DESCRIPTION</th></tr>
<tr><td>Comment</td><td>This item allows you to enter a comment about this palletizing instruction.</td></tr>
<tr><td>Type</td><td>This item allows you to specify whether this palletizing instruction will palletize or depalletize. If set to PALLET, the robot stacks the pallet from bottom to top. If set to DEPALLET, the robot unloads the pallet from top to bottom.</td></tr>
<tr><td>Incr</td><td>This item allows you to specify how the pallet register will increment or decrement. This value, along with the specified order, controls which position will be processed by the robot next. The value of the pallet register indicates the next position to be processed. The order indicates which portion of the pallet register is updated first, second, and last. In the example, the specified order is row, column, layer. For palletizing, when Incr is set to 1, the instruction initializes the pallet register as the first row, first column, first layer. When Incr is set to —1 the instruction initializes the pallet register as the last row, last column, and first layer.
For depalletizing, when Incr is set to 1, the instruction initializes the pallet register as the last row, last column, last layer. When Incr is set to —1 the instruction initializes the pallet register as the first row, first column, and last layer.
 

TYPE = PALLET
<table>
<tr><td></td><td>INCR = 1</td><td>INCR = - 1</td></tr>
<tr><td></td><td colspan="2">order RCL</td></tr>
<tr><td>Initial</td><td>[1, 1, 1]</td><td>[2, 2, 1]</td></tr>
<tr><td rowspan="8">↓</td><td>[2, 1, 1]</td><td>[1, 2, 1]</td></tr>
<tr><td>[1, 2, 1]</td><td>[2, 1, 1]</td></tr>
<tr><td>[2, 2, 1]</td><td>[1, 1, 1]</td></tr>
<tr><td>[1, 1, 2]</td><td>[2, 2, 2]</td></tr>
<tr><td>[2, 1, 2]</td><td>[1, 2, 1]</td></tr>
<tr><td>[1, 2, 2]</td><td>[2, 1, 1]</td></tr>
<tr><td>[2, 2, 2]</td><td>[1, 1, 2]</td></tr>
<tr><td>[1, 1, 1]</td><td>[2, 2, 1]</td></tr>
</table>
TYPE = DEPALLET
<table>
<tr><td></td><td>INCR = 1</td><td>INCR = - 1</td></tr>
<tr><td></td><td colspan="2">order RCL</td></tr>
<tr><td>Initial</td><td>[2, 2, 2]</td><td>[1, 1, 2]</td></tr>
<tr><td rowspan="8">↓</td><td>[1, 2, 2]</td><td>[2, 1, 2]</td></tr>
<tr><td>[2, 1, 2]</td><td>[1, 2, 2]</td></tr>
<tr><td>[1, 1, 2]</td><td>[2, 2, 2]</td></tr>
<tr><td>[2, 2, 1]</td><td>[1, 1, 2]</td></tr>
<tr><td>[1, 2, 1]</td><td>[2, 1, 1]</td></tr>
<tr><td>[2, 1, 1]</td><td>[1, 2, 1]</td></tr>
<tr><td>[1, 1, 1]</td><td>[2, 2, 1]</td></tr>
<tr><td>[2, 2, 2]</td><td>[1, 1, 2]</td></tr>
</table>
</td></tr>
<tr><td>Pal Reg</td><td>This item allows you to specify the pallet register that will be used to store the next position to be processed.</td></tr>
</table>

Table 6–26. PALLETIZINGE Pallet Editor Items (Cont'd)

Order	This item allows you to specify the order in which the robot will palletize or depalletize. The available options are CRL, CLR, RCL, or RLC where •C = Column •R = Row •L = Layer
Columns Rows Layers	This item allows you to specify the total number of columns, rows, or layers in the stack, whether pattern is a linear or free, and whether the orientation of the TCP is a fixed or calculated orientation. • **LINE** - indicates the positions are lying in a straight line. P----P----P----P----P • **FREE** - indicates the positions can be anywhere in a horizontal plane. P PP PP • **FIX** - indicates the TCP orientation equals the orientation recorded at the [1,1,1] position. • **INTER** - indicates the TCP orientation vector is calculated.
Auxiliary Pos	This item allows you to specify whether or not an auxiliary position will be used.
APPR	This item allows you to specify how many approach route positions to use. The number you specify will be the number of positions created by the instruction for you to record.
RTRT	This item allows you to specify how many retreat route positions to use in the palletizing instruction. The number you specify will be the number of positions created by the instruction for you to record.

Procedure 6-10 Using the Palletizing Editor To Enter the Palletizing-E Instruction

Conditions

- You have created a program.
- You are editing a program.

Steps

1. **Select the instruction:**
   a. Press F1, [INST].
   b. Select Palletizing. See the following screen for an example.

```
PALLETIZING statem
1 PALLETIZING-B        5 PALLETIZING-END
2 PALLETIZING-BX       6
3 PALLETIZING-E        7
4 PALLETIZING-EX       8
Main Pallet
     [END]
  Select item
```

c. Select PALLETIZING-E. The Pallet Editor will be displayed. See the following screen for an example.

```
Main Pallet
PALLETIZING CONFIGURATION
  PALETIZING_1   [                    ]
  TYPE = [PALLET]           INCR = [  1]
  PAL REG = [  1]          ORDER = [CRL]
  COLUMN = [1 LINE FIX]
  ROWS   = [1 LINE FIX]
  LAYERS = [1 LINE FIX]
  AUXILIARY POS = [NO]
  APPR=[  1] RTRT=[  1]
```

Note The term Palletizing_1 in the example screen indicates that this is the first palletizing instruction in the program.

2. To return to your program at any time, press F1, PROG.

```
Quit pallet editor?
```

a. Press F4, YES, to return to your program.

b. Press F5, NO, to remain in the Pallet Editor.

3. To set up the palletizing configuration:

a. Enter a Comment.

- Press ENTER.
- Move the cursor to select a method of naming the program.

For example, if you chose Alphabetic, press a function key corresponding to the first letter. Press that key until the letter you want is displayed in the comment field. Press the right arrow key to move the cursor to the next space. Continue until the entire program name is displayed. **To delete a character,** press BACK SPACE.

- When you are finished, press ENTER.

b. Select the TYPE.

- **To palletize**, press F2, PALLET.
- **To depalletize**, press F3, DEPALL.

c. Type the increment value.

d. Type the number of the pallet register you want to use.

e. Type the order. Available orders include: CRL, CLR, RCL, RLC.

- **To select rows**, press F2, R.
- **To select columns**, press F3, C.
- **To select layers**, press F4, L.

f. Set up your column information.

- Type the total number of columns.
- Select the type of positions.
- **To select linear positions**, press F2, LINE.
- **To select free positions**, press F3, FREE.
  - — Select the type of TCP orientation.
- **To select fixed TCP orientation**, press F2, FIX.
- **To select calculated TCP orientation**, press F3, INTER.

g. Set up your row information.

- Type the total number of rows.
- Select the type of positions.
- **To select linear positions**, press F2, LINE.
- **To select free positions**, press F3, FREE.
  - — Select the type of TCP orientation.
- **To select fixed TCP orientation**, press F2, FIX.

- **To select calculated TCP orientation**, press F3, INTER.

h. Set up your layer information.

- Type the total number of layers.
- Select the type of positions.
- **To select linear positions**, press F2, LINE.
- **To select free positions**, press F3, FREE.
- Select the type of TCP orientation.
  - **To select fixed TCP orientation**, press F2, FIX.
  - **To select calculated TCP orientation**, press F3, INTER.

i. Select the auxiliary position.

- **To use an auxiliary position**, press F2, YES.
- **To not use an auxiliary position**, press F3, NO.

j. Type the number of approach positions.

k. Type the number of retreat positions.

l. Type the number of approach and retreat patterns you want to use.

m. When you are finished setting up the PALLETIZING-E instruction, press F5, DONE. See the following screen for an example.

```
Main Pallet
TEACH BOTTOM POINTS
 1: *P[1,1,1]
 2: *P[10,1,1]
 3: *P[1,2,1]
 4: *P[1,1,2]
```

Note The number of bottom positions you record depends on the number of columns, rows, and layers you entered in the Pallet Editor. In the above screen example, 10 columns, 2 rows, and 2 layers were entered. The order was entered as CRL.

4. To return to your program at any time, press >, and then press F1, PROG.

```
Quit pallet editor?
```

a. Press F4, YES, to return to your program.

b. Press F5, NO, to remain in the Pallet Editor.

5. **To return to the Pallet Editor at any time**, press F1, BACK.

6. **To record each pallet position**:

a. Jog the robot to the position.

b. Hold down the SHIFT key and press F4, RECORD.

c. When you are finished recording positions, press F5, DONE. See the following screen for an example.

```
Main Pallet
PALLETIZING ROUTE POINTS
      IF PL[1] = [*,*,*}
 1: J PAL_1[A_2] 30% FINE
 2: J PAL_1[A_1] 30% FINE
 3: J PAL_1[BTM] 30% FINE
 4: J PAL_1[R_1] 30% FINE
 5: J PAL_1[R_2] 30% FINE
```

Note The number of route positions you record depends on the number of approach and retreat positions you entered in the Pallet Editor. In the above example, 2 approach and 2 retreat positions were entered.

7. **To return to your program at any time**, press >, and then press F1, PROG.

```
Quit pallet editor?
```

a. Press F4, YES, to return to your program.

b. Press F5, NO, to remain in the Pallet Editor.

8. **To return to the Pallet Editor at any time**, press F1, BACK.

9. **To set up position default information**, press F2, POINT.

10. **To record each route position**:

a. Jog the robot to the position.

b. Hold down the SHIFT key and press F4, RECORD.

c. When you are finished recording positions, press F5, DONE. You are returned to your program and the PALLETIZING E instruction has been inserted.

6.26.5 PALLETIZING-EX Instruction

The PALLETIZING-EX instruction can be used in the following cases:

- The approach and retreat routes of your robot have different directions and orientations. The route is the number of robot positions you record to move your robot to the pallet and then away from the pallet. See Figure 6–153 .
- If your stacking pattern consists of a line pattern. A line pattern has columns, rows, and layers placed vertically, horizontally, and perpendicularly. For the line pattern, you record three end positions that indicate where the rows, columns, and layers are located. In Figure 6–153 , [1,1,1] is the last route position recorded, [1,1,5] [1,3,1] and [4,1,1] are the three recorded end positions.
- If your stacking pattern consists of a free pattern. A free pattern has columns, rows and layers placed vertically, horizontally, and perpendicularly. For the free pattern, you record two end positions where the rows, columns, and layers are located. In Figure 6–153 , [1,1,1] is the last route position recorded, [1,1,5] indicates the position of the top layer, and [2,1,1], [3,1,1], and [4,1,1] indicate the position of each row.
- Your layer patterns alternate. See Figure 6–153 .
- You want to use an auxiliary position to define the direction of the pattern.
- You want to define the orientation of the tool at the pick/place point.

Figure 6–153. Example of PALLETIZING-EX Instruction

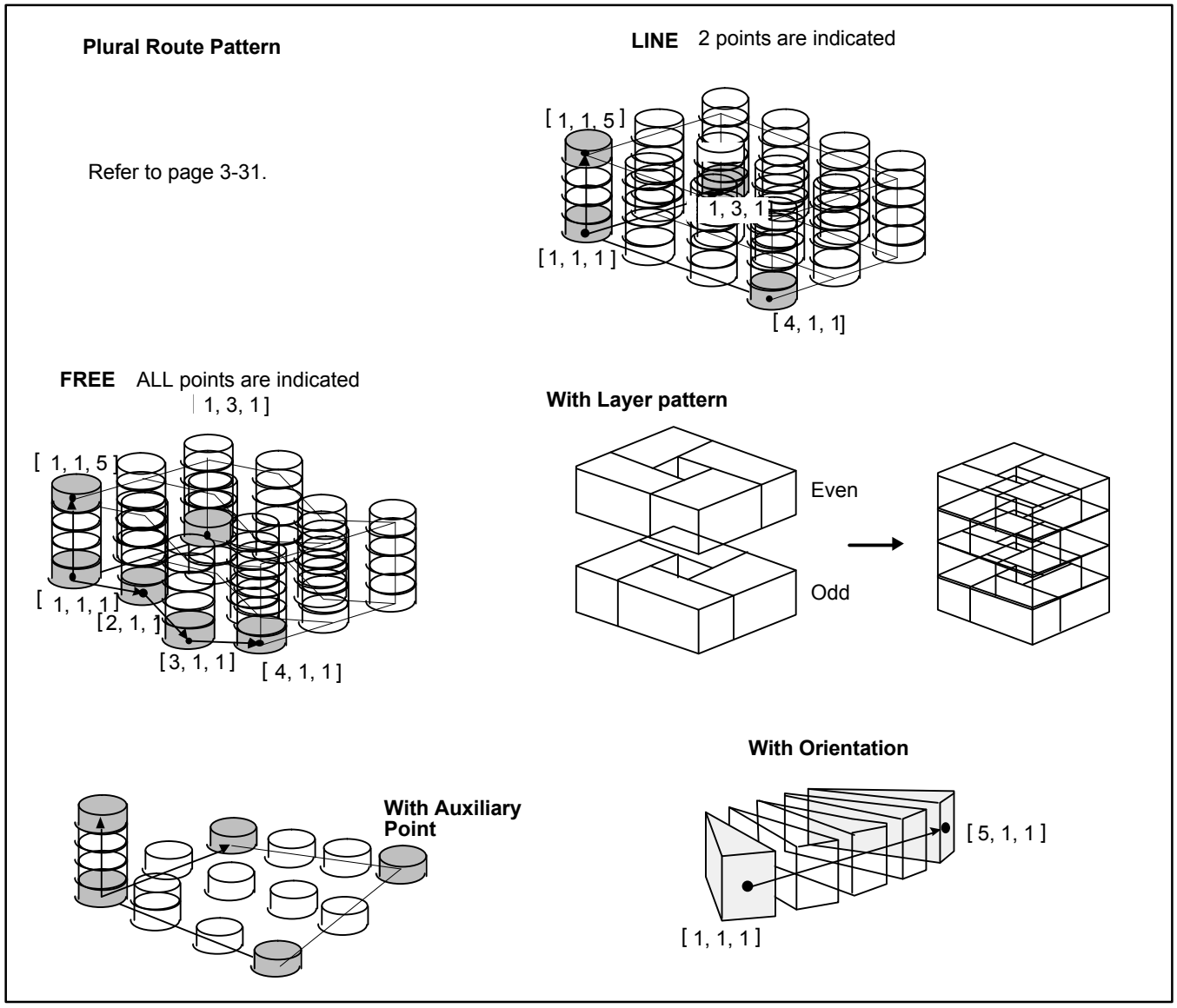
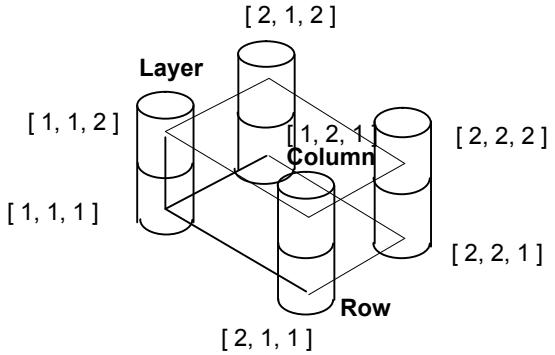


Table 6–27 lists and describes each PALLETIZING-EX instruction item you can specify in the Pallet Editor.

Table 6–27. PALLETIZING-EX Pallet Editor Items

ITEM	DESCRIPTION
Comment	This item allows you to enter a comment about this palletizing instruction.

Table 6–27. PALLETIZING-EX Pallet Editor Items (Cont'd)

<table>
<tr><td>Type</td><td>This item allows you to specify whether this palletizing instruction will palletize or depalletize. If set to PALLET , the robot stacks the pallet from bottom to top. If set to DEPALLET , the robot unloads the pallet from top to bottom.</td></tr>
<tr><td>Incr</td><td>This item allows you to specify how the pallet register will increment or decrement. This value, along with the specified order, controls which position will be processed by the robot next. The value of the pallet register indicates the next position to be processed. The order indicates which portion of the pallet register is updated first, second, and last. In the example, the specified order is row, column, layer. For palletizing, when Incr is set to 1, the instruction initializes the pallet register as the first row, first column, first layer. When Incr is set to —1 the instruction initializes the pallet register as the last row, last column, and first layer.
For depalletizing, when Incr is set to 1, the instruction initializes the pallet register as the last row, last column, last layer. When Incr is set to —1 the instruction initializes the pallet register as the first row, first column, and last layer.
 

TYPE = PALLET
<table>
<tr><th></th><th>INCR = 1</th><th>INCR = - 1</th></tr>
<tr><td>Initial</td><td>[1, 1, 1]</td><td>[2, 2, 1]</td></tr>
<tr><td></td><td>[2, 1, 1]</td><td>[1, 2, 1]</td></tr>
<tr><td></td><td>[1, 2, 1]</td><td>[2, 1, 1]</td></tr>
<tr><td></td><td>[2, 2, 1]</td><td>[1, 1, 1]</td></tr>
<tr><td></td><td>[1, 1, 2]</td><td>[2, 2, 2]</td></tr>
<tr><td></td><td>[2, 1, 2]</td><td>[1, 2, 1]</td></tr>
<tr><td></td><td>[1, 2, 2]</td><td>[2, 1, 1]</td></tr>
<tr><td></td><td>[2, 2, 2]</td><td>[1, 1, 2]</td></tr>
<tr><td></td><td>[1, 1, 1]</td><td>[2, 2, 1]</td></tr>
</table>
TYPE = DEPALLET
<table>
<tr><th></th><th>INCR = 1</th><th>INCR = - 1</th></tr>
<tr><td>Initial</td><td>[2, 2, 2]</td><td>[1, 1, 2]</td></tr>
<tr><td></td><td>[1, 2, 2]</td><td>[2, 1, 2]</td></tr>
<tr><td></td><td>[2, 1, 2]</td><td>[1, 2, 2]</td></tr>
<tr><td></td><td>[1, 1, 2]</td><td>[2, 2, 2]</td></tr>
<tr><td></td><td>[2, 2, 1]</td><td>[1, 1, 2]</td></tr>
<tr><td></td><td>[1, 2, 1]</td><td>[2, 1, 1]</td></tr>
<tr><td></td><td>[2, 1, 1]</td><td>[1, 2, 1]</td></tr>
<tr><td></td><td>[1, 1, 1]</td><td>[2, 2, 1]</td></tr>
<tr><td></td><td>[2, 2, 2]</td><td>[1, 1, 2]</td></tr>
</table>
</td></tr>
<tr><td>Pal Reg</td><td>This item allows you to specify the pallet register that will be used to store the next position to be processed.</td></tr>
</table>

Table 6–27. PALLETIZING-EX Pallet Editor Items (Cont'd)

Order	This item allows you to specify the order in which the robot will palletize or depalletize. The available options are CRL, CLR, RCL, or RLC, where • C = Column • R = Row • L = Layer
Columns Rows Layers	This item allows you to specify the total number of columns, rows, or layers in the stack, whether pattern is a linear or free, and whether the orientation of the TCP is a fixed or calculated orientation. • **FREE** - indicates the positions can be anywhere in a horizontal plane. P PP PP • **LINE** - indicates the positions are lying in a straight line. P----P----P----P----P • **FIX** - indicates the TCP orientation equals the orientation recorded at the [1,1,1] position. • **INTER** - indicates the TCP orientation vector is calculated.
Auxiliary Pos	Auxiliary Pos cannot be specified for the PALLETIZING-BX instruction.
APPR	This allows you to specify how many approach route positions to use. The number you specify will be the number of positions created by the instruction for you to record.
RTRT	This allows you to specify how many retreat route positions to use in the palletizing instruction. The number you specify will be the number of positions created by the instruction for you to record.
PATTERN	This allows you to specify the number of approach and retreat routes to use. These routes patterns can be **DIRECT** or **MODULO**. A direct pattern uses the same multiple routes for each layer. A modulo pattern allows you to alternate the multiple route patterns between layers.

Procedure 6-11 Using the Palletizing Editor To Enter the Palletizing-EX Instruction

Conditions

- You have created a program.
- You are editing a program.

Select the Instruction

1. **Select the instruction:**

a. Press F1, [INST].

b. Select Palletizing. See the following screen for an example.

```
PALLETIZING statem
1 PALLETIZING-B       5 PALLETIZING-END
2 PALLETIZING-BX      6
3 PALLETIZING-E       7
4 PALLETIZING-EX      8
Main Pallet
    [END]
 Select item
```

c. Select PALLETIZING-EX. The Pallet Editor will be displayed. See the following screen for an example.

```
Main Pallet
PALLETIZING CONFIGURATION
  PALETIZING_1  [                          ]
  TYPE = [PALLET]             INCR = [  1]
  PAL REG = [  1]            ORDER = [CRL]
  COLUMN = [1 LINE FIX]
  ROWS   = [1 LINE FIX]
  LAYERS = [1 LINE FIX]
  AUXILIARY POS = [NO]
  APPR=[  1] RTRT=[  1]
```

Note The term Paletizing_1 in the example screen indicates that this is the first palletizing instruction in the program.

2. To return to your program at any time, press F1, PROG.

```
Quit pallet editor?
```

a. Press F4, YES, to return to your program.

b. Press F5, NO, to remain in the Pallet Editor.

3. To set up the palletizing configuration:

a. Enter a Comment.

- Press ENTER.
- Move the cursor to select a method of naming the program.

  For example, if you chose Alphabetic, press a function key corresponding to the first letter. Press that key until the letter you want is displayed in the comment field. Press the right arrow key to move the cursor to the next space. Continue until the entire comment is displayed. **To delete a character,** press BACK SPACE.
- When you are finished, press ENTER.

b. Select the TYPE.

- **To palletize**, press F2, PALLET.
- **To depalletize**, press F3, DEPALL.

c. Type the increment value.

d. Type the number of the pallet register you want to use.

e. Type the order. Available orders include: CRL, CLR, RCL, RLC.

- **To select rows**, press F2, R.
- **To select columns**, press F3, C.
- **To select layers**, press F4, L.

f. Set up your column information.

- Type the total number of columns.
- Select the type of positions.
  - **To select linear positions**, press F2, LINE.
  - **To select free positions**, press F3, FREE.
- Select the type of TCP orientation.
  - **To select fixed TCP orientation**, press F2, FIX.
  - **To select calculated TCP orientation**, press F3, INTER.

g. Set up your row information.

- Enter the total number of rows.
- Select the type of positions.
  - **To select linear positions**, press F2, LINE.
  - **To select free positions**, press F3, FREE.
- Select the type of TCP orientation.

— **To select fixed TCP orientation**, press F2, FIX.

— **To select calculated TCP orientation**, press F3, INTER.

h. Set up your layer information.

- Type the total number of layers.
- Select the type of positions.
  - — **To select linear positions**, press F2, LINE.
  - — **To select free positions**, press F3, FREE.
- Select the type of TCP orientation.
  - — **To select fixed TCP orientation**, press F2, FIX.
  - — **To select calculated TCP orientation**, press F3, INTER.

i. Select the auxiliary position.

- **To use an auxiliary position**, press F2, YES.
- **To not use an auxiliary position**, press F3, NO.

j. Type the number of approach positions.

k. Type the number of retreat positions.

l. Type the number of approach and retreat patterns you want to use.

m. When you are finished setting up the PALLETIZING EX instruction, press F5, DONE. See the following screen for an example.

```
Main Pallet
TEACH BOTTOM POINTS
  1: *P[1,1,1]
  2: *P[10,1,1]
  3: *P[1,2,1]
  4: *P[1,1,2]
```

Note The number of bottom positions you record depends on the number of columns, rows, and layers you entered in the Pallet Editor. In the above screen example, 10 columns, 2 rows, and 2 layers were entered. The order was entered as CRL.

4. To return to your program at any time, press >, and then press F1, PROG.

```
Quit pallet editor?
```

a. Press F4, YES, to return to your program.

b. Press F5, NO, to remain in the Pallet Editor.

5. **To return to the Pallet Editor at any time**, press F1, BACK.

6. **To record each pallet position**:

a. Jog the robot to the position.

b. Hold down the SHIFT key and press F4, RECORD.

c. When you are finished recording positions, press F5, DONE. See the following screen for an example.

```
Main Pallet
PALLETIZING ROUTE PATTERSN
 PTN [ 1] = [  *,  *,  *]
 PTN [ 2] = [  *.  *.  *]
```

7. **To return to your program at any time** , press >, and then press F1, PROG.

```
Quit pallet editor?
```

a. Press F4, YES, to return to your program.

b. Press F5, NO, to remain in the Pallet Editor.

8. **To return to the Pallet Editor at any time**, press F1, BACK.

9. **To set up each palletizing route pattern**,

a. Move the cursor to each asterisk.

b. **To not alternate the patterns between layers**, press F2, DIRECT.

c. **To alternate the patterns between layers**, press F3, MODULO.

d. Enter the position number for each asterisk.

e. When you are finished recording positions, press F5, DONE. See the following screen for an example.

```
Main Pallet
PALLETIZING ROUTE PATTERSN
     IF PL[1] = [*,*,*}
1: J PAL_1[A_2] 30% FINE
2: J PAL_1[A_1] 30% FINE
3: J PAL_1[BTM] 30% FINE
4: J PAL_1[R_1] 30% FINE
5: J PAL_1[R_2] 30% FINE
```

Note The number of route positions you record depends on the number of approach and retreat positions you entered in the Pallet Editor. In the above example, 2 approach and 2 retreat positions were entered.

10. **To return to your program at any time**, press >, and then press F1, PROG.
    a. Press F4, YES, to return to your program.
    b. Press F5, NO, to remain in the Pallet Editor.
11. **To return to the Pallet Editor at any time**, press F1, BACK.
12. **To set up position default information**, press F2, POINT.
13. **To record each route position**:
    a. Jog the robot to the position.
    b. Hold down the SHIFT key and press F4, RECORD.
    c. When you are finished recording positions, press F5, DONE. You are returned to your program and the PALLETIZING EX instruction has been inserted.

6.26.6 PALLETIZING-END Instruction

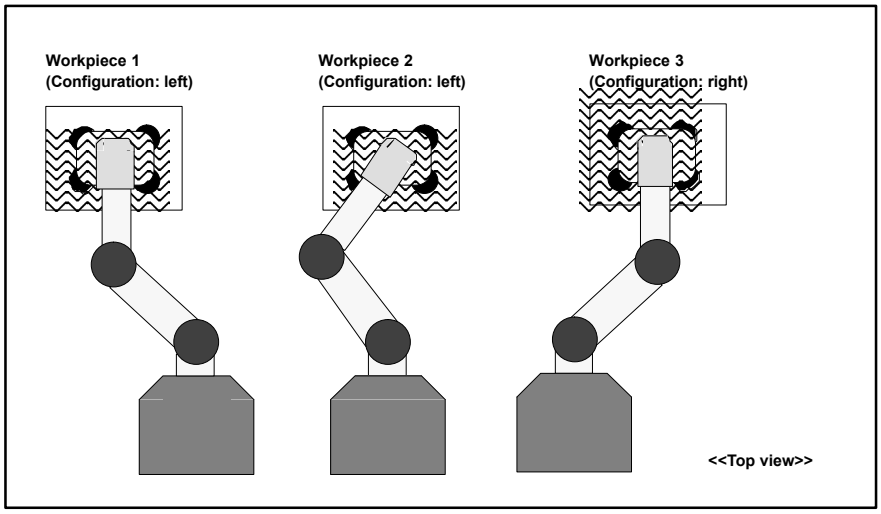
The PALLETIZING-END instruction resets the pallet register after the robot has finished stacking or unstacking a pallet.

6.27 ALL-POINT TEACHING FOR PALLETIZING

Conventional robots are capable of palletizing or depalletizing workpieces in a single configuration. Refer to "Positional Information" for information about configuration.

Figure 6–154 shows an example of palletizing with a horizontally articulated robot with two arm configurations: left and right.

Figure 6–154. Palletizing Example



In this example, the palletizing of three workpieces is taught using different configurations-right for workpiece 3 and left for the other two.

Even after teaching as above, in a conventional system the program is executed so that the robot palletizes or depalletizes workpieces such that the configuration is the same at each bottom point.

This function enables a robot to palletize or depalletize workpieces according to the configurations it has been taught, simply by setting a system variable.

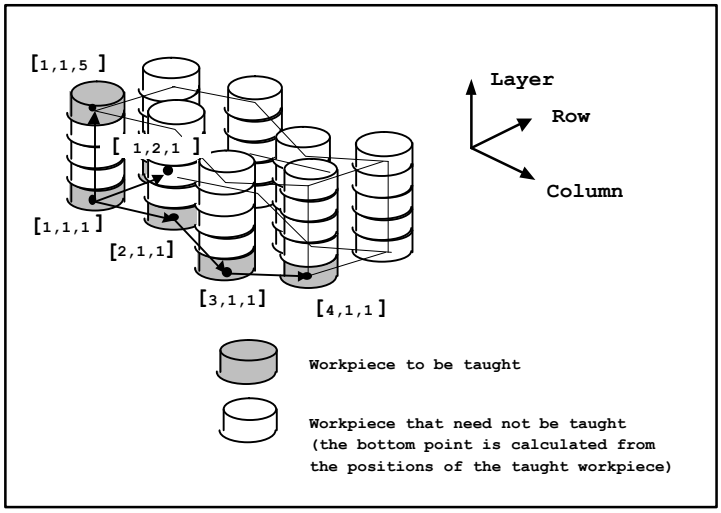
Operation

To palletize or depalletize workpieces, maintaining the same attitude and configuration as that taught, perform the following setup procedure:

1. On the system variable screen, set system variable $PALCFG.$FREE_CFG_EN to TRUE (the initial value is TRUE).
2. On the palletizing initial data screen, set attitude control to INTER for the palletizing direction (one of COLUMN, ROWS, and LAYERS) for which the arrangement method is FREE.

As a result, each workpiece along the specified direction is palletized or depalletized with the same attitude and configuration as that of the reference workpiece used for teaching.

The following is an example of an irregular arrangement of four columns, two rows, and five layers. Set the palletizing initial data as follows:



COLUMN = [4 FREE INTER]

ROWS = [2 LINE FIX]

LAYERS = [5 LINE FIX 1]

In this example, FREE and INTER are set for COLUMN. When system variable $PALCFG.$FREE_CFG_EN is TRUE, workpieces are palletized or depalletized with the following configurations:

- P[1,1,1] for workpieces in the first column- P[2,1,1] for workpieces in the second column- P[3,1,1] for workpieces in the third column- P[4,1,1] for workpieces in the fourth column

Notes

When you use this function, note the following:

- FREE and INTER cannot be set for more than one direction of COLUMN, ROWS, and LAYERS (such setting is possible when system variable $PALCFG.$FREE_CFG_EN is FALSE, that is, the function is not used).

This is because the configuration for a workpiece that need not be taught (whose bottom point is calculated from the positions of the reference workpieces which are taught) cannot be uniquely determined from such a setting.

When the program contains such a setting, the execution will fail and the following error message is displayed:

```
PALT-024 Calculation error occurred
```

- Create a program so that it is not stopped due to an alarm for an unmatched configuration.

When the configuration at the current position and that at the destination position are different, the robot cannot move between the positions with linear movement (program execution is stopped and an unmatched configuration alarm is output).

The configurations at the approach positions and retreat positions for palletizing are specified in the same way as those for the bottom points. If the first palletizing command specifies linear motion type, an unmatched configuration alarm might be output, depending on the configuration of the robot when the command is executed.

This problem does not occur when the first palletizing command specifies joint motion type. For example, the following program does not cause an unmatched configuration alarm to be output for palletizing with three approach positions and two retreat positions:

Palletizing Program that will not Cause an Unmatched Configuration Alarm

```
:
10:  PALLETIZING EX_1
11: J PAL_1[A_3] 100% FINE
12: L PAL_1[A_2] 500mm/sec Cnt50
13: L PAL_1[A_1] 300mm/sec Cnt10
14: L PAL_1[BTM] 100mm/sec FINE
15:  OPEN_HAND1
16: L PAL_1[R_1] 300mm/sec Cnt10
17: L PAL_1[R_2] 500mm/sec Cnt50
18:  PALLETIZING-END-1
  :
```

6.28 PALLET REGISTER INSTRUCTIONS

A pallet register stores layer, row, and column information for palletizing. Thirty-two registers are available for all the programs in the controller combined. Pallet registers are identified by numbers, 1-32. Pallet register instructions manipulate pallet registers arithmetically.

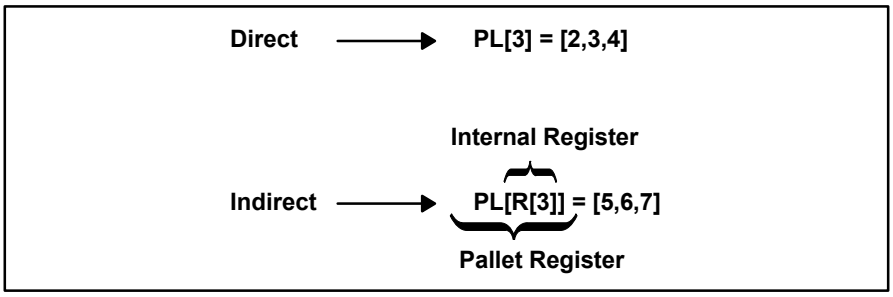
Up to 127 pallet registers are available for all programs in the controller combined. You can increase the number of pallet registers at controlled start. Refer to the “System Operations” appendix of the application-specific *Setup and Operations Manual* for information on performing a Controlled start.

Pallet Register Addressing

Many instructions employ direct or indirect addressing techniques. When direct addressing is used, the actual value is entered into the instruction. For example, if the pallet register instruction PL[2]= [2,3,4] is used, the contents of pallet register 2 is replaced with the value [2,3,4].

When indirect addressing is used, the instruction contains a register within a pallet register. This indicates that the actual value of the internal register becomes the pallet register number of the pallet register. See Figure 6–155 .

Figure 6–155. Direct and Indirect Addressing Example



In Figure 6–155 , the first instruction illustrates direct addressing. This instruction causes the contents of pallet register 3 to be replaced with the value [2, 3, 4].

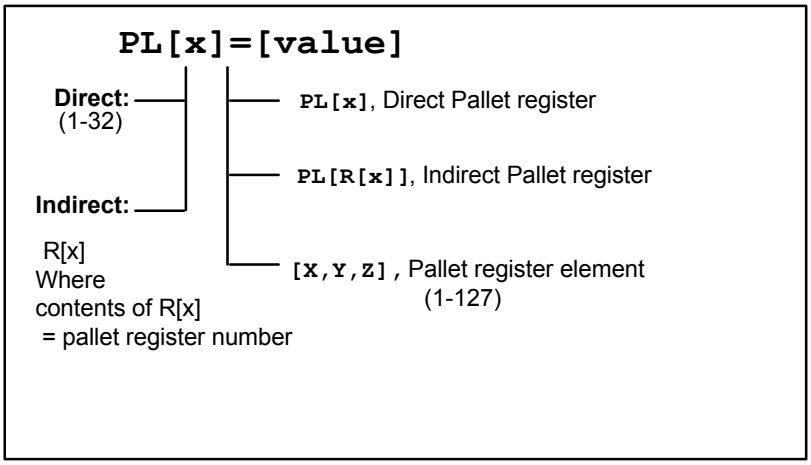
The second instruction in Figure 6–155 illustrates indirect addressing. In this instruction, R[3] is the internal register and PL[R[3]] is the external pallet register. If the value of the internal register R[3] is 3, the pallet register number becomes PL[R[3]=3] or PL[3]. Therefore, the result of the second instruction is that the contents of the pallet register, PL[3], is to be replaced with the value [5, 6, 7].

PL[x] = [value]

The PL[x] = [value] instruction stores a value in a pallet register. See Figure 6–156 .

Figure 6–156. PL[x] = [value]

```
PL[...] = ...
```



PL[x]=[value][operator][value]

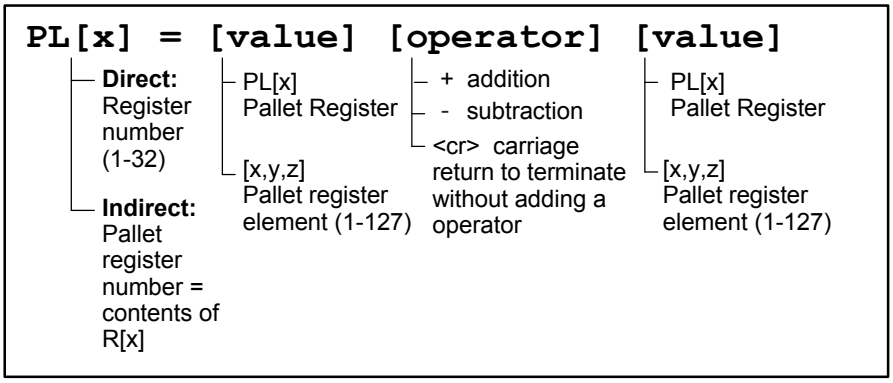
The PL[x] = [value] [operator] [value] instructions store the result of an arithmetic operation in a Pallet register. The arithmetic operations are addition and subtraction. See Figure 6–157 .

You can use multiple arithmetic operators in a single instruction. However, there are the following limitations:

- You cannot mix +, -, or */ in the same instruction.
- The maximum number of arithmetic operators you can have in the same instruction is 5.

Figure 6–157. PL[x] = [value] [operator] [value]

```
PL[...] = ... + ...
PL[...] = ... - ...
```



For conditional branching instruction using PL[], refer to Section 6.9.4 .

6.29 TOUCH SENSE INSTRUCTIONS

6.29.1 Overview

Touch sensing is an option and might not be installed on your controller. Touch sensing instructions are used to implement the touch sensing programming. There are four touch sensing instructions provided:

- Search Start
- Search End

- Touch Offset
- Touch Offset End

Note You can not use the Find command to locate touch sense instructions when you edit a program.

Caution

You can not use the Replace command to replace a motion instruction with a touch sense instruction. Doing so will cause a *memory write failure* error. If you want to replace the motion instruction, first delete the motion instruction, and then insert the touch sense instruction.

6.29.2 Search Start Instruction

Search Start indicates where the search motion begins. The Search Start instruction assigns the touch schedule to be used during the search and the position register where the positional information is to be stored. Each Search Start must have a Search End. A new Search Start can not be executed until a Search End has been executed. See Figure 6–158 .

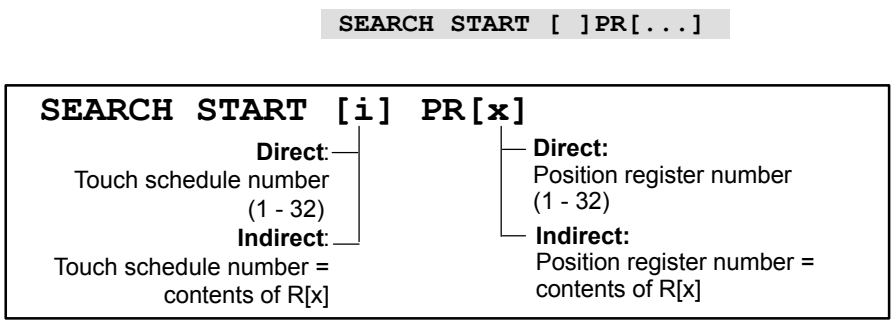
Schedule and position register range checking is done when you run the program.

If the program causes a nested search start error, move the cursor to the beginning of the program and run the program again. The error should clear automatically.

You cannot move the cursor to a search motion statement and execute it unless you also execute the preceding search start statement.

Note Backward execution of Search Start [ ] P[ ] will disable Search Start.

Figure 6–158. SEARCH START [i] PR[x]



6.29.3 Search End Instruction

Search End stops the search. It is important to end a Search Start properly by using Search End in your program. Otherwise, if incremental is set to ON, all the motion afterward will be affected by the incremental search. See Figure 6–159 .

Figure 6–159. SEARCH END

```
SEARCH END
```

```
SEARCH END
```

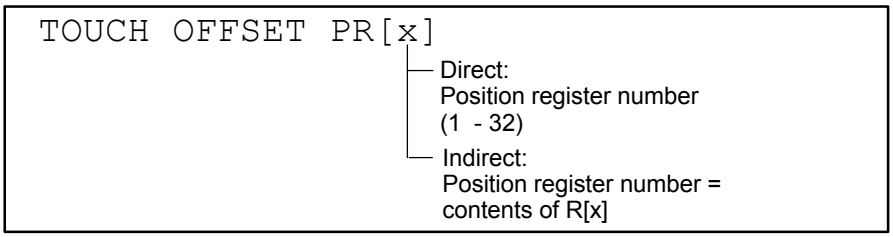
6.29.4 Touch Offset Instruction

Touch Offset indicates where the position shift begins in the program. The shift amount is determined by the information in the specified position register relative to world frame. The shift amount is generated by the search routine. Position register range checking is done when you run the program. See Figure 6–160 .

Note When using a simple search pattern, Touch Offset and Touch Offset End are not used. A simple search stores the actual position of the point being searched into the Position Register.

Figure 6–160. TOUCH OFFSET PR[x]

```
TOUCH OFFSET PR[   ]
```



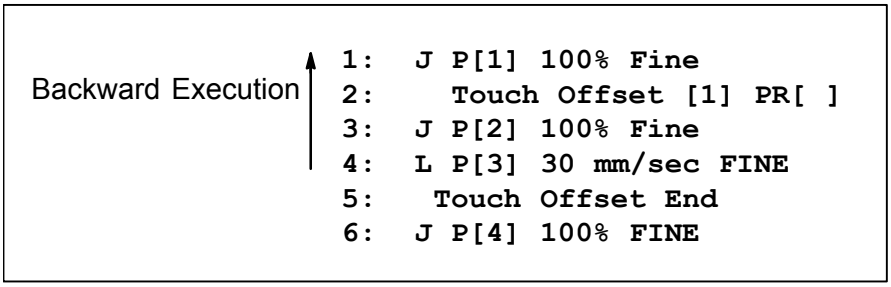
Touch Offset and Touch Offset End allow backward execution of the program with the following conditions:

Note Backward execution of Touch Offset PR [ ] will disable Touch Offset.

- Backward execution of a Touch Offset instruction**will not terminate the offset**. When backward execution is done, any added or "touched up" positions will be the recorded position plus the position register offset. For example, in line 1 of the following program, the robot position is equal to P[1] + PR[1] when backward execution begins at line 2.

- Touch Offset terminates at only two conditions:
  - — The Touch Offset End is executed.
  - — The program aborts.
- The Touch Offset End instruction**will terminate the offset** . However, if backward execution begins prior to executing the Touch Offset End instruction all positions will be offset. For example, both P[2] and P[3] in the example shown in Figure 6–161 will be offset by PR[1] when backward execution begins at line 4. Also, if you scroll to P[4] and execute that instruction, the position will be offset by the PR[1] amount.

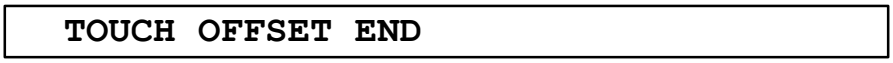
Figure 6–161. Backward Execution Example



6.29.5 Touch Offset End Instruction

Touch Offset End indicates where the position shift ends. It is important to properly end a Touch Offset by using Touch Offset End in your program. Otherwise, all the motion afterward will be affected by the touch offset position register. See Figure 6–162 .

Figure 6–162. TOUCH OFFSET END



Note When using a simple search pattern, Touch Offset and Touch Offset End are not used. A simple search stores the actual position of the point being searched into the Position Register.

6.30 MIXED LOGIC INSTRUCTIONS

6.30.1 Overview

The Mixed Logic Instructions option provides any combination of operators and data in TP program assignment statements, IF statements and WAIT statements. Mixed Logic also supports the NOT operator (!) and parenthesis (“(“ and “)”).

You can specify Mixed Logic instructions through the Register menu, the I/O menu, the IF/SELECT menu, and the WAIT menu.

Mixed Logic instructions must be specified in parentheses, as follows:

- DO[1]=(DI[1] AND !DI[2])
- IF (DI[1]) JMP,LBL[1]
- WAIT (DI[1])

When the statement does not have parentheses, the statement is executed the same as any other Logic Instruction.

Mixed logic also provides the additional boolean data types Flag and *Marker* and the additional statement*TC_Online*.

Mixed logic also provides Background Logic, which allows simultaneous, continuous, repeated execution of as many as eight TP programs containing only assignment statements.

6.30.2 Data Types

The following data types are available for mixed logic instructions:

Table 6–28. Data Types

Type	Value	Data
Numerical	The data can have numerical value. Both integer and real values are available.	Register, Constant, GI/O, AI/O, Element of position register , Argument, System variable
Boolean	The data can be On or Off .	DI/O, RI/O, UI/O, SI/O, WI/O, On, Off, Flag, Marker

Note Position type data and Palletizing type data cannot be specified in mixed logic instructions.

6.30.3 Operators

The following operators are available for mixed logic instructions.

Table 6–29. Arithmetical Operators

Operator	Operation
+	Addition of left side and right side.
-	Subtraction of right side from left side
*	Multiplication of left side and right side.
/	Division of left side and right side.
MOD	Remainder of the division of left side and right side.
DIV	Round off the decimal part of the division of left side and right side.

- Arithmetical operators are used for numerical data only. If numerical operators are used for boolean data, the INTP-205 Variable Type Mismatch error occurs.
- The output data type of arithmetical operators is always numerical.

Table 6–30. Logical Operators

Operator	Operation
AND	Logical AND of left side and right side.
OR	Logical OR of left side and right side.
!	Logical NOT of left side and right side.

- Logical operators are used for boolean data only. If logical operators are used for numerical data, the INTP-205 Variable Type Mismatch error occurs.
- The output data type of logical operators is always boolean.

Table 6–31. Comparison Operators

Operator	Operation
=	When the left side and right side are equal, returns ON; when unequal, returns OFF.
<>	When the left side and right side are not equal, returns ON; when equal, returns OFF.
<	When the left side is less than the right side, returns ON; when greater, returns OFF.

Table 6–31. Comparison Operators (Cont'd)

Operator	Operation
>	When the left side is greater than the right side, returns ON; when less, returns OFF.
<=	When the left side is less than the right side, or they are equal, returns ON; when greater, returns OFF.
>=	When the left side is greater than the right side, or they are equal, returns ON; when less, returns OFF.

- "=" and "<>" can be used for both numerical data and boolean data.
- "<","<sup>","<=", and " >=" are used for numerical data only. If they are used for boolean data, the INTP-205 Variable Type Mismatch error occurs.

The priority of the operators is indicated in the following table.

Table 6–32. Priority of Operators

Priority	Operator
HIGHEST	!
	*, /, DIV , MOD
	+, —
MEDIUM	<, >, <=, >=
	=, <>
	AND
LOWEST	OR

6.30.4 Expressions

Mixed logic instructions can be specified in assignment statements, IF statements, and WAIT statements.

Assignment Statements

The following is an example of a mixed logic assignment statement:

```
R[1] = ((GI[1] + R[1]) * AI[1])
DO[1] = (DI[1] AND (GI[1] — GI[2]))
```

- The first = from the left indicates an assignment statement. The other =s indicate comparison. The result of the right side expression is assigned to the left side data.
- When the left side data type is boolean and the result of the right side expression is numerical, left side data becomes OFF when the right side value is less than 1 and greater than —1; the left side data becomes ON when the right side value is greater than 1 or less than —1. This behavior is the same as a normal assignment statement.
- When the left side data type is numerical and the result of the right side expression is boolean, the left side data becomes 0 when the right side value is OFF, and the left side data becomes 1 when the right side value is ON. This behavior is the same as a normal assignment statement.
- If a real value is assigned to GO, AO, or an integer type system variable, the decimal part is rounded off.
- "Pulse" cannot be specified in mixed logic instructions. You must use normal logic instructions to specify "Pulse".
- Position type data and palletizing type data cannot be specified in the right side or left side of mixed logic instructions. You must use normal logic instructions to specify position or palletizing type data.
- The maximum number of items (data or operators) allowed in an assignment statement is approximately 20. The exact maximum number allowed varies according to data type.

The following data can be specified in the left side of the assignment:

Table 6–33. Data Assignments

Type	Data
Boolean	DO, RO, UO, SO, WO, Flag, Marker
Numerical	Register, GO, AO, element of a position register, system variable

IF Statements

The following is an example of a mixed logic IF statement:

```
IF (R[1] = (GI[1] + R[1]) * AI[1]) JMP LBL[1]
IF (DI[1] AND (!DI[2] OR DI[3])) JMP LBL[1]
```

- You can specify mixed logic expressions in the condition part of an IF statement.
- The result of the IF statement must be boolean.
- When the result of the expression is on, the action part of the statement, for example, JMP LBL, is executed.
- The following statements can be used in the action part of an IF statement when mixed logic is used in the condition part:

```
JMP LBL[ ]
CALL
Mixed Logic assignment statement
Pulse statement
```

- Mixed Logic assignment statements and Pulse statements can be specified in the action part of the IF statement only when the condition part is a Mixed Logic expression. See the following example:

```
IF (DI[1]), DO[1]=(On)
IF (DI[2]), DO[1]=Pulse
```

- The maximum number of items (data or operators) in an IF statement is approximately 20. The exact maximum number of items varies according to data type.

WAIT Statements

The following is an example of a mixed logic WAIT statement:

```
WAIT (DI[1] AND (!DI[2] OR DI[3]))
```

- Mixed Logic expressions can be specified in the condition of a WAIT statement
- The result of the expression must be boolean.
- The WAIT statement waits until the result of the expression becomes ON.
- "On+","Off-" and "ERR_NUM" cannot be specified in mixed logic instructions. You must use normal logic instructions to specify them.
- The maximum number of items (data or operators) in a WAIT statement is approximately 20. The exact maximum number of items varies according to the data type.

6.30.5 Adding Mixed Logic Instructions

Editing mixed logic statements is more complex than editing normal statements because of the greater variety of data types and operators that can be used, and the flexibility with which they can be combined.

To make editing easier, the following functions are provided:

- To begin teaching a mixed logic instruction you must first choose a statement containing parentheses.
- Item selection menu shows the items available according to the location in the statement.

- When the combination of items is invalid, for example two operators adjoin, empty item is inserted automatically, and you are prompted to select an item.
- When an item is deleted, the related items are deleted at the same time. For example, an operator is deleted, the next data is deleted at the same time.
- When the cursor is on the item of Mixed Logic expression, if the expression is wrong, an error message is displayed on the prompt line. The following messages can be displayed.

Table 6–34. Mixed Logic Error Messages

Error Message	Explanation
Parentheses mismatch	The number of left parenthesis and right parenthesis are not same.
Invalid index	Index number is not correct.
Variable type mismatch	Data type is not correct for the operator.
Invalid parameter name	System variable name is not correct.
Untaught element	Empty item (...) exists.
Invalid motion group	The specified motion group of PR[ ] is not available for the program.
Invalid item for output	The left side item of assignment statement is invalid.
Invalid item for Mixed Logic	The item can not be used in Mixed Logical expression.
Syntax error	Invalid statement.

To add Mixed Logic instructions to a program,

1. At the Edit menu, press F1, INST to go to the Instruction menu.
2. Select the kind of instruction you want to add: Register, I/O, IF/SELECT, or WAIT.
3. At the Statement menu, select the Mixed Logic statement, which contains parentheses:

   (. . .)

4. Build the instruction as you normally would.
5. To **change** an item of a Mixed Logic statement, when the cursor is on the item you want to change, press F4, CHOICE. The available items will be displayed.
6. You can **insert** items in any Mixed Logic statement, except for the left side of an Assignment statement. Press F1, INSERT. An empty item,

   " . . . "

   is inserted before the cursor and an item selection menu is displayed.

If you select an operator, an empty item is inserted after the operator, and the data menu is displayed.

7. You can **delete** items in any Mixed Logic statement except for the left side of an assignment statement. Move the cursor to the item and press CHOICE. The DELETE key will be displayed on F2. If an operand is deleted, the following data item will also be deleted.
8. To add or delete a NOT(!) operator, when the cursor is on a Digital I/O item of a Mixed Logic expression, except the left side of an Assignment statement, press F5, (!). The NOT(!) operator will be added or deleted.
9. To change the left side of an Assignment statement when the right side of the Assignment statement is Mixed Logic instructions, move the cursor to the left side item and press F4, CHOICE. A menu that includes the items that can be specified in the left side of an Assignment statement will be displayed.

6.30.6 Background Logic

6.30.6.1 Overview

Background logic allows any teach pendant program that includes only assignment statements to be executed in the background. The program is executed from beginning to end repeatedly. This execution is not affected by E-STOP, HOLD, or any alarms.

There are two execution modes in Background Logic, normal mode and fast mode.

- In normal mode, all mixed logic instructions can be executed in the background. 300 items are processed at every ITP (ITP is normally 8msec). An “item” is data or an operator. If there are more than 300 items, scanning time is increased. For example, when there are 800 operations, scanning time is 24msec.
- In fast mode, only I/O data and logical operators can be executed. 8000 items are processed every 8msec.The maximum number of items in fast mode is 8000. Indirection of index, for example DO[R[1]], can not be used in fast mode.

Refer to Table 6–35 for information on the background logic execution modes.

Use Procedure 6-13 to execute background logic.

Table 6–35. Background Logic Execution Modes

Mode	Max. number of Items	Scanning time	Available data	Available operators
Normal Mode	No limitation	(Num. of items / 300) * ITP Number of items means the total of all background logic programs. ITP is normally 8msec.	F[], M[]*, DI[], DO[], AI[], AO[], GI[],GO[], SI[], SO[], UI[], UO[], RI[], RO[] WI[], WO[] On, Off R[], PR[i.j], AR[] Constant Parameter Timer, Timer overrun	(,), !, AND, OR, =, <>, <, < =, >, >=, +,—, *, /, DIV, MOD
Fast Mode	8,000	8msec	F[], M[]*, DI[], DO[], SI[], SO[], UI[], UO[], RI[], RO[] WI[], WO[]	(,), !, AND, OR

M[ ] cannot be specified as left side of assignment statement in background logic.

- 8 programs can be executed as Background Logic at the same time.
- If the program includes anything but assignment statements, "INTP-443 Invalid item for Mixed Logic" error occurs at execution.
- While a program is being executed in the background, the program cannot be edited, and the program cannot be executed as a normal task. If the program is not running in the background, it can be run as a normal task in order to test it
- When a program is running in the background, the program cannot be loaded as overwrite.
- If a program is executing as Background Logic at power down, the program is executed at next power up automatically in the same execution mode.
- The Background Logic execution occurs before than normal program execution. The Background Logic execution takes about 1msec in every ITP. The background logic execution may affect the cycle time of the normal program.

  To decrease the execution time of Background Logic, change $MIX_LOGIC.$ITEM_COUNT to smaller value. The default value of $MIX_LOGIC.$ITEM_COUNT is 300, which is the number of items processed each ITP.

- Assignment statements with the IF condition can be executed by Background Logic. The assignment statement is not executed when the condition is OFF. In the following example:

  ```
  IF (DI[1]), DO[1]=(DI[2])
  ```

  the value of DO[1] is set to DI[2] when DI[1] is ON, DO[1] is not changed when DI[1] is OFF.

  The IF condition cannot be executed in fast mode, so any program that includes the IF condition is always executed in Normal mode.

- Pulse instruction can be used in Background Logic. It can be combined with the IF condition to create an OFF DELAY TIMER. In the example:

  ```
  IF (DI[1]), DO[1]=Pulse 1sec
  ```

  DO[1] has 1sec pulse when DI[1] stays ON than 1 sec. If DI[1] turns OFF before 1 sec, DO[1] turns OFF immediately. While DI[1] is OFF, this statement does not set DO[1].

  The Pulse instruction cannot be executed in fast mode, so any program that contains the Pulse instruction is always executed in Normal mode.

  To keep DO[1] on for 1 sec even if DI[1] turns OFF, use the following statements.

  ```
  F[1]=(DI[1] OR (F[1] AND DO[1]))
    IF (F[1]), DO[1]=Pulse 1sec
  ```

- If a Pulse instruction does not specify the time, it will be one scan pulse in Background Logic execution. In the example:

  ```
  IF (DI[1]), DO[1]=Pulse
  ```

  In this case, DO[1] becomes ON for only one scan when DI[1] is changed from OFF to ON. This can be used as edge detection.

  Pulse instruction without time means the pulse of $DEFPULSE in normal execution, so it is different in Background Logic execution.

Note FAST Mode is not available in the Mate controller. The mode setting is always NORMAL and it cannot be changed in Background Logic menu.

You can use the Background Logic screen to set up and execute programs as Background Logic. Refer to Table 6–36 for a description of each background logic screen item. Table 6–37 lists the background logic screen operations.

Table 6–36. Background Logic Screen Items

ITEM	DESCRIPTION
PROGRAM	Enter the name of the program you want to execute as background logic.
STATUS	This item displays the status of the background logic program: • Stop: The program is stopped • Running: The program is running in Normal mode. • Running (Fast): The program is running in Fast mode.
MODE	Use this item to select the execution mode: • Auto: If the program can be executed in fast mode, it is executed in fast mode. If it cannot be executed in fast mode, it is executed in normal mode. • Fast: The program is always executed in Fast mode. If the program includes instructions that are not available for Fast mode, the error " INTP-44 Invalid item for fast mode" occurs at execution. • Normal: The program is always executed in Normal mode. Default is "Auto". If you would like to execute the program in Fast mode but it is executed in Normal mode, set this item "Fast". The line number of invalid statement for Fast mode is displayed in error message.

Table 6–37. Background Logic Screen Operations

FUNCTION KEY	DESCRIPTION
RUN	Press this key to execute the program as background logic. If it contains statements that are not available in background mode, an error will be posted.
STOP	Press this key to stop background execution of the program.

Procedure 6-13 Setting Up Background Logic

1. Press MENU
2. Select 6, SETUP
3. Press F1, TYPE
4. Select BACKGROUND. You will see a screen similar to the following:

```
Background logic
   Normal mode scan time: 8 msec
    PROGRAM      STATUS          MODE
    1 LOGIC1     Running         Normal
    2 LOGIC2     Stop            Fast
    3 LOGIC3     Running(Fast)   Auto
    4            Stop            Auto
    5            Stop            Auto
    6            Stop            Auto
    7            Stop            Auto
    8            Stop            Auto
```

5. In the PROGRAM column, enter the name of the program you want to run as Background Logic.

The STATUS column will display the status of the background logic program:

- Stop: The program is stopped
- Running: The program is running in Normal mode..
- Running (Fast): The program is running in Fast mode.

6. In the MODE column, select the execution mode:.

- Auto: If the program can be executed in fast mode, it is executed in fast mode. If it cannot be executed in fast mode, it is executed in normal mode.
- Fast: The program is always executed in Fast mode. If the program includes instructions that are not available for Fast mode, the error " INTP-44 Invalid item for fast mode" occurs at execution.
- Normal: The program is always executed in Normal mode.

Default is "Auto". If you would like to execute the program in Fast mode but it is executed in Normal mode, set this item "Fast". The line number of invalid statement for Fast mode is displayed in error message.

Note The scanning time of normal mode execution is displayed on the upper line of the screen.

7. To run the program as background logic, press F2, RUN.

The program will run. If it contains statements that are not available in background mode, an error will be posted.

8. To stop background execution of the program, press F3, STOP.

6.30.6.2 Flag

Flags (F[ ])are internal I/O ports that can be read and set. They are not connected to any actual I/O device, they are like a Boolean type variable.

The value of flags can be recovered by Power Failure Recovery function (HOT Start). It is the same as the other output ports, for example DO.

The following conditions set all Flags to OFF:

- COLD start
- CONTROL start
- INIT start
- I/O assignment is changed, even though in HOT start.
- I/O configuration is changed, even though in HOT start.

DI, DO, UI, UO, GI and GO can be assigned to flags by configuring them as Rack 34, Slot 1, Start point 1-1024.

When UI/UO are assigned to flags, program execution can be controlled by TP program or Background Logic. For Example:

```
Rack  Slot  Start Pt.
UI[1-18]  34  1     1
```

In this case, when F[6] is changed from ON to OFF, UI[6:START] is changed from ON to OFF, and the selected program is executed.

To display the Flag monitor menu

1. Press MENUS.
2. Select 5 I/O.
3. Press F1, TYPE.
4. Select Flag. You will see a screen similar to the following:

```
Flag
        #      STATUS              1/1024
   F[   1]       ON  [               ]
   F[   2]       OFF [               ]
   F[   3]       OFF [               ]
   F[   4]       ON  [               ]
   F[   5]       OFF [               ]
   F[   6]       OFF [               ]
   F[   7]       OFF [               ]
   F[   8]       OFF [               ]
   F[   9]       OFF [               ]
   F[  10]       OFF [               ]
```

You can change the value of flags in this menu.

5. To display the port detail menu, press F2, DETAIL. You will see a screen similar to the following:

```
Port Detail
    Flag                  [   1]
    Comment:[                    ]
```

You can change flag comments at this screen.

Examples of Edge Detection, Counter and Timer in Fast Mode Background Logic

Fast mode has better performance than Normal mode, and Fast mode does not affect the performance of normal program execution. But Fast mode cannot use numerical operation and Pulse instruction.

Example 1: Edge Detection

The following program is edge detection of DI[1]. DO[1] becomes ON only when DI[1] is changed from OFF to ON.

```
1: DO[1]=(DI[1] AND !F[1])
2: F[1]=(DI[1])
```

F[1] has the DI[1] value of the previous scan. DO[1] becomes ON when DI[1] is ON and the previous value of DI[1] is OFF.

Example 2: Counter

The following is the example of 4 bit counter of DI[1] edge. The counter value is set in F[41-44]. You can read the counter value by GI[1] if GI[1] is assigned as rack 34, slot 1, start pt 41 and points 4.

```
1:  F[2]=(DI[1] AND !F[1]) ;
2:  F[1]=(DI[1]) ;
3:  ! BIT1 ;
4:  F[3]=(F[41]) ;
5:  F[41]=((F[2] AND !F[3]) OR (!F[2] AND F[3])) ;
6:  F[2]=(F[2] AND F[3]) ;
7:  ! BIT2 ;
8:  F[3]=(F[42]) ;
9:  F[42]=((F[2] AND !F[3]) OR (!F[2] AND F[3])) ;
10:  F[2]=(F[2] AND F[3]) ;
11:  ! BIT3 ;
12:  F[3]=(F[43]) ;
13:  F[43]=((F[801] AND !F[3]) OR (!F[801] AND F[3])) ;
14:  F[2]=(F[2] AND F[3]) ;
15:  ! BIT4 ;
16:  F[3]=(F[44]) ;
17:  F[44]=((F[2] AND !F[3]) OR (!F[2] AND F[3])) ;
18:  F[2]=(F[2] AND F[3]) ;
```

The first 2 lines the edge detection of DI[1], F[2] becomes ON in one scan when DI[1] is changed from OFF to ON.

The lines 4-6 process to count up the first bit of the counter (F[41]). F[3] is the work variable to keep the original value. In line 5, F[41] is reversed when F[2] is ON, and F[41] is not changed when F[2] is OFF. In line 6, F[2] is turned OFF when the original F[41] is OFF, it means overflow does not occur.

The lines 8-10 are for the second bit (F[42]), the lines 12-14 are for the third bit (F[43]), and the lines 16-18 are for the fourth bit (F[44]).

Example 3: Timer

Timer can be programmed by using the counter, because the scanning time of Fast mode is always 8msec. The following is the example of the 80msec Pulse. This program works as same as 'IF (DI[1]), DO[1]=Pulse 80msec'.

```
1:  F[1]=(DI[1]);
2:  F[2]=(F[1] AND !F[4]);
3:  DO[1]=F[2]
4:  ! BIT1 ;
5:  F[3]=(F[41]) ;
6:  F[41]=(F[1] AND ((F[2] AND !F[3]) OR (!F[2] AND F[3]))) ;
7:  F[2]=(F[2] AND F[3]) ;
8:  ! BIT2 ;
```

```
9:  F[3]=(F[42]) ;
10:  F[42]=(F[1] AND ((F[2] AND !F[3]) OR (!F[2] AND F[3]))) ;
11:  F[2]=(F[2] AND F[3]) ;
12:  ! BIT3 ;
13:  F[3]=(F[43]) ;
14:  F[43]=(F[1] AND ((F[2] AND !F[3]) OR (!F[2] AND F[3]))) ;
15:  F[2]=(F[2] AND F[3]) ;
16:  ! BIT4 ;
17:  F[3]=(F[44]) ;
18:  F[44]=(F[1] AND ((F[2] AND !F[3]) OR (!F[2] AND F[3]))) ;
19:  F[2]=(F[2] AND F[3]) ;
20:  ! 80msec is 10 * 8msec. 10=0b1010 ;
21:  F[4]=(F[44] AND !F[43] AND F[42] AND !F[41])
```

F[1] is the work variable to keep DI[1] value.

All bits of counter (F[41-44]) are cleared when F[1] is OFF.

F[2] is the work variable, the counter value is increased when F[2] is ON.

When the counter value is 10 (F[41]:ON, F[42]:OFF, F[43]:ON, F[44]:OFF), F[4] becomes ON, and F[2] becomes OFF, so the counter is not increased until DI[1] is turned OFF.

6.30.6.3 Marker

The Marker Screen allows you to monitor the status of Markers.

Marker (M[ ]) is similar to flag, but the value of markers is not set directly. When Marker (M[ ]) is specified in the left side of an assignment (=) in a TP program and the statement is executed, the expressions are defined as Background Logic internally, and the expression is executed repeatedly. The marker (M[ ]) always has the result of the expression.

By default, the Marker function is disabled, the Marker menu is not displayed, and M[ ] can not be taught in TP program. To use the Marker function, set $MIX_LOGIC.$USE_MKR to TRUE.

Example:

```
M[1]=(DI[1] AND DI[2])
```

After this line is executed in a normal TP program (not in Background Logic), M[1] always has the result of the right side expression. When both DI[1] and DI[2] are ON, M[1] is ON, in the other case M[1] is OFF.

- When a Marker assignment statement is executed in a normal TP program, the statement is registered to Background Logic. The statement is executed as Background Logic until another

expression redefines the marker. Execution of the statement does not stop even though the program is paused or aborted, because it is Background Logic.

- By default, there are 8 markers (M[1]-M[8]). The number of markers can be changed by system variable "$MIX_LOGIC.$NUM_MARKERS". Maximum number of markers is 100. One marker takes 300bytes permanent memory pool.
- The scanning time to calculate a marker assignment statement is the same as the scanning time of Normal mode Background Logic. Having marker assignment statements defined affects the scan time of background logic.
- Clear the defined marker expression to stop the calculation. To clear the defined expression, execute the CLEAR operation in the Marker detail menu or execute the following line of TP program.

```
M[1]=()
```

- If a marker is not assigned to an expression and the marker is used in another statement, the "INTP-347 Read I/O value failed" error occurs when the marker value is read.
- M[ ] cannot be specified in the left side of assignment statement in Background Logic.

To display the Marker monitor menu

1. Press MENUS.
2. Select 5, I/O.
3. Press F1, TYPE.
4. Select Marker. You will see a screen similar to the following:

```
Marker
        #       STATUS               1/8
   M[   1]        ON   [               ]
   M[   2]        OFF  [               ]
   M[   3]        OFF  [               ]
   M[   4]        ON   [               ]
   M[   5]        OFF  [               ]
   M[   6]        OFF  [               ]
   M[   7]        OFF  [               ]
   M[   8]        OFF  [               ]
```

5. To display the port detail menu, press F2, DETAIL. You will see a screen similar to the following:

```
Port Detail
 Marker                [   1]
 Comment:[                        ]
 Expression:
  M[1]=((DI[1] OR DI[2]) AND !DI[3]
      AND !(DI[4] AND DI[5]))
 Monitor:
  M [1]                  ON
  DI[1]                  OFF
  DI[2]                  ON
  DI[3]                  OFF
  DI[4]                  OFF
  DI[5]                  ON
```

You can change marker comments in this screen.

- The port detail screen displays the defined expression.

  To clear the defined expression , press F5, CLEAR. When the message

```
Clear expression?
```

  is displayed, press F4, YES.

- Current value of every data item in the defined expression is displayed in monitor area.
- You can change marker comments in this screen.

6.30.6.4 TC_ONLINE

TC_ONLINE is similar to marker. The TC_ONLINE statement defines the expression and the expression is calculated as Background Logic. While the result of the expression is OFF, all TP and KAREL programs that have group motion are stopped.

By default, the TC_ONLINE function is disabled, the TC_ONLINE menu is not displayed and TC_ONLINE statement can not be taught in TP program. To use TC_ONLINE function, please set $MIX_LOGIC.$USE_TCOL to TRUE.

For Example:

```
TC_ONLINE (DI[1] AND DI[2])
```

After this line is executed, all TP programs are stopped while DI[1] or DI[2] is OFF. Refer to Table 6–38 .

Table 6–38. TC Online Instruction

TC_ONLINE (...)	Defines the specified Mixed Logic Instructions as a TC_ONLINE expression and enables the TC_ONLINE function.
TC_ONLINE DISABLE*	Disable TC_ONLINE function. Any TP program is not stopped by TC_ONLINE when TC_ONLINE is disabled.
TC_ONLINE ENABLE*	Enable TC_ONLINE function. This is used to enable TC_ONLINE that is disabled by TC_ONLINE Disable.

* By default, TC_ONLINE DISABLE and TC_ONLINE ENABLE are not available. set $MIX_LOGOC.$USE_TCOLSIM to FALSE to use these instructions.

- When TC_ONLINE (...) statement is executed, the specified expression is defined as a TC_ONLINE expression.

  While TC_ONLINE is enabled, the defined expression is calculated as Background Logic.

  If the result of the expression is OFF, all TP and KAREL programs except NOPAUSE are stopped.
- If a program is started while TC_ONLINE expression is OFF, the program is paused immediately. All types of start are affected by TC_ONLINE except Shif+BWD.
- Only when a program is executed by Shift+BWD, the program can be executed even though TC_ONLINE expression is OFF.
- TC_ONLINE expression is calculated at every ITP (ITP is normally 8msec) even though the scanning time of Background Logic is longer than ITP.
- Programs that do not have motion group or in which ’ignore pause’ parameter is TRUE are not paused even if TC_ONLINE condition is OFF.
- When $MIX_LOGIC.$USE_TCOLSIM is TRUE (default), TC_ONLINE DISABLE and TC_ONLINE ENABLE cannot be taught by Edit menu. The setting of ENABLE/DISABLE of TC_ONLINE should be changed in the TC_ONLINE menu. In this case, TC_ONLINE is enabled automatically when motion statement execution is completed. This means that TC_ONLINE is disabled only when the operator moves the robot temporarily.
- When $MIX_LOGIC.$USE_TCOLSIM is FALSE, TC_ONLINE DISABLE and TC_ONLINE ENABLE can be taught by Edit menu. The setting of ENABLE/DISABLE of TC_ONLINE is not changed automatically when motion statement execution is completed.

Use Procedure 6-16 to add a TC_ONLINE Instruction. Use Procedure 6-17 to display the TC_ONLINE Monitor Menu.

Procedure 6-16 Adding a TC_ONLINE Instruction

Steps

1. At the Edit menu, press F1, INST to go to the Instruction menu.
2. Select TC_ONLINE.

   If $MIX_LOGIC.$USE_TCOLSIM is**TRUE**, select the item and complete the statement as you normally would

   If $MIX_LOGIC.$USE_TCOLSIM is**FALSE**,

   a. Select 1 (...).

   b. Select the item and complete the statement as you normally would.
3. To**change** a TC_ONLINE statement, press F4, CHOICE, on the first '(' in the TC_ONLINE statement, then select 2 ENABLE.

Procedure 6-17 Displaying the TC_ONLINE Monitor Menu

Steps

1. Press MENUS.
2. Select 5, I/O.
3. Select F1, TYPE.
4. Select TC ONLINE. You will see a screen similar to the following:

```
TC_ONLINE
                            1/6
 Status:                            On
 Enable:                            TRUE
 Expression:
  ((DI[1] OR DI[2]) AND !DI[3] AND
  !(DI[4] AND DI[5]))
 Monitor:
  DI[1]                     ON
  DI[2]                     OFF
  DI[3]                     ON
  DI[4]                     OFF
  DI[5]                     OFF
```

- “Enable” line shows whether TC_ONLINE is enabled or not now. User can change this item in this menu.
- “Status“ line shows the status of TC_ONLINE. It is the result of the defined expression.

- The current value of every data in the defined expression is displayed in monitor area.
- The defined expression is displayed in expression area.

5. To clear the defined expression, move the cursor to Monitor.

a. Press F5, CLEAR. You will see the following prompt:

```
Clear expression?
```

b. Press F4, YES.

6.30.7 Backup/Restore

Every data of Mixed Logic Instructions are saved as follows.

- Mixed Logic Instructions are saved in TP file of the program.
- Background Logic program is saved to TP file.
- Setting of Background Logic menu is saved in MIXLOGIC.SV.

  MIXLOGIC.SV has the value of the following system variables.

  — $MIX_LOGIC
  — $MIX_BG
  — $MIX_MKR
  — $DRYRUN
  — $DRYRUN_PORT
  — $DRYRUN_SUB

- Comments of Flag and Marker are saved in DIOCFGSV.IO.
- If DI/O, UI/O or GI/O are assigned to flags, the assignment is saved in DIOCFGSV.IO.

6.31 PROCESS SYNCHRONIZATION

Process synchronization instructions are used to synchronize processes between robots in a Multi-arm environment.

SYNC_SCHED[seq_n]

The SYNC_SCHED instruction sets the synchronization schedule number.

The synchronization schedule number must be defined by this instruction prior to using INPOS/PR_STRT/PR_END/ or PR_SYNC instructions.

The SYNC_SCHED instruction sets the synchronization schedule number for the proceeding process synchronization instructions. The schedule defines the robots to be synchronized. Use the schedule combinations defined in Table 6–39 . See Figure 6–163 for an example of how to use the instruction in a program.

Table 6–39. Schedule and Robot Combinations

Schedule #	Robot #s
1	Robot #1 and #2
2	Robot #2 and #3
3	Robot #1 and #2
4	Robot #1, #2, and #3
5	Robot #1 and #3
6	Robot #2 and #3
7	Robot #1, #2, and #3
8	Robot #1, #2, #3, and #4
9	Robot #1 and #4
10	Robot #2 and #4
11	Robot #1, #2, and #4
12	Robot #3 and #4
13	Robot #1, #3, and #4
14	Robot #2, #3, and #4
15	Robot #1, #2, #3, and #4
16	Robot #1, #2, #3, #4, and #5
17	Robot #1 and #5
18	Robot #2 and #5
19	Robot #1, #2, and #5
20	Robot #3 and #5
21	Robot #1, #3, and #5
22	Robot #2, #3, and #5

Table 6–39. Schedule and Robot Combinations (Cont'd)

23	Robot #1, #2, #3, and #5
24	Robot #4 and #5
25	Robot #1, #4, and #5
26	Robot #2, #4, and #5
27	Robot #1, #2, #4, and #5
28	Robot #3, #4, and #5
29	Robot #1, #3, #4, and #5
30	Robot #2, #3, #4, and #5
31	Robot #1, #2, #3, #4, and #5

Figure 6–163. SYNC_SCHED

```
SYNC_SCHED[sch_n]
```

INPOS[n]

The INPOS instruction indicates the master signal slave, and waits for PR_END. INPOS is used on a Master robot as a motion option only. The seq_n indicates the sequence number. The seq_n must match between robots and be the same for the PR_START and PR_END instructions. See Figure 6–164 for an example of how to use the INPOS instruction in a program.

Figure 6–164. INPOS

```
INPOS[n]
```

Note The INPOS instruction cannot be combined with an Application instruction.

se

PR_STRT[n]

This instruction waits for a Master INPOS **before** application processing. PR_STRT is used on a slave robot only as a motion option. PR_STRT must be followed by a PR_END instruction. See Figure 6–165 for an example of how to use the INPOS instruction in a program.

Note The PR_STRT instruction can be combined with an Application instruction.

Figure 6–165. PR_STRT[n]

```
PR_STRT[n]
```

PR_END[n]

This instruction signals the master after application processing. PR_END is used on a slave robot only as a motion option. PR_END must be used after a PR_START instruction. See Figure 6–166 for an example of how to use the INPOS instruction in a program.

Note The PR_END instruction can be combined with an Application instruction.

Figure 6–166. PR_END[n]

```
PR_END[n]
```

PR_SYNC[n]

PR_SYNC is used on a slave robot only as a motion option. PR_SYNC is equivalent to a PR_STRT and PR_END on one line. See Figure 6–167 for an example of how to use the INPOS instruction in a program.

Note The PR_SYNC instruction can be combined with an Application instruction.

Figure 6–167. PR_SYNC[n]

```
PR_SYNC[n]
```

6.32 VISION INSTRUCTIONS

6.32.1 Overview

This page contains help for the Vision Instruction Set. When editing the teach pendant program, press the F1,[INST] key for a list of teach pendant instructions. The *i*RVision instructions are under the group heading VISION (not to be confused with PC VISION - which is used with a PC based vision system).

After selecting the VISION group the vision instructions will be displayed. Select one of the desired instructions:

- RUN_FIND
- GET_OFFSET
- SET_REFERENCE
- CAMERA_CALIB
- VR[].OFFSET
- VR[].MODELID

6.32.2 RUN_FIND

```
VISION RUN_FIND '...' CAMERA_VIEW[...]
```

This instruction begins with a request to acquire the image. After the image has been acquired, control returns to the robot and the robot can move. The vision process continues to process.

Parameters

Vision Process Name: This indicates the name of the vision process. The name can be selected from the list of valid vision processes.

Camera View Number: An option that is used with a vision program with more than one camera image and the images can not be acquired at the same time. For example, suppose there is one overhead camera, and the robot is holding a part with two target holes. The target holes are too far apart to be in the same FOV. Hole one is acquired, the robot moves and hole two is acquired. In this case there would be two RUN_FIND instructions — one for each hole.

6.32.3 GET_OFFSET

```
VISION GET_OFFSET 'VP1' VR[2] JMP LBL[99]
```

This command provides a destination vision register for the vision process. After the vision process has completed and the data has been transferred to the appropriate vision register, control returns to the robot.

Parameters

Vision Process Name: Type the name of the vision process.

Vision Register Index: Type the index of the vision register to receive the data.

Jump Label: Type the label of a jump statement. The program jumps to this label in the event the vision program fails or no parts are found.

6.32.4 SET_REFERENCE

```
VISION SET_REFERENCE 'VP1'
```

Use this instruction to set the reference position from a program. The steps for training a vision process are to train and find a vision target. Set this vision position to be the zero or reference position thus Set Reference. Vision will return a zero result for the offset. Record the pick/place robot position without an offset. The vision position and the robot position have been linked.

Parameters

Vision Process Name: Use the [CHOICE] key to select the vision process from a list of vision processes.

6.32.5 CAMERA_CALIB

```
VISION CAMERA_CALIB 'TWOPTCAL' Request=1
```

This instruction allows you to calibrate the system from a teach pendant program. This is especially useful for a robot mounted camera. Checks can be made periodically. When calibration is required, the calibration program can be run to re-calibrate vision. This is useful after the camera has been bumped or replaced.

Parameters

Camera Name : Type the name of the camera calibration.

Request Number: Type the plane number for the grid to be calibrated when used with grid calibration.

6.32.6 VR[].OFFSET

```
PR[1]=VR[2].OFFSET
```

This instruction transfers the vision offset from the Vision Data register to a position register so that it can be used with a Move instruction with an offset clause.

Parameters

Position Register Index: Specify an integer to indicate the index of the destination position register.

Vision Register Index: Specify an integer to indicate the source vision register.

6.32.7 VR[]. MODELID

```
R[33]=VR[2].MODELID
```

This instruction transfers the vision process model ID from the Vision Data register to a register so that it can be used to determine which model was found.

Parameters

Register Index: Specify an integer to indicate the index of the destination register.

Vision Register Index: Specify an integer to indicate the source vision register.

6.32.8 SAMPLE PROGRAM

The following example illustrates how the vision instructions are used together:

Line 2 finds the vision process VP1

Line 3 the results are stored in Vision register 5. If there were no parts found, the program will jump to LBL 999.

Line 4 the offset is transferred to PR[1]

Line 5 the model ID is saved in R[4]

Line 7 is the approach point

Line 8 is the pick position

Line 9 actuates the gripper

Line 10 Move through approach.

```
 1:  LBL[1]
 2:  VISION RUN_FIND 'VP1'
 3:  VISION GET_OFFSET 'VP1' VR[5]
  :  JMP LBL[999]
 4:  PR[1]=VR[5].OFFSET
 5:  R[4]=VR[1].MODELID
 6:
 7:L @P[1] 500MM/SEC CNT75 Offset,PR[1]
 8:L @P[2] 100MM/SEC FINE Offset,PR[1]:
 9:  CALL PICK
10:L @P[1] 300mm/sec CNT75 Offset,PR[1]
  :  Offset,PR[1]
 11: END
 12:
 13: LBL[999:NO PART]
 14: UALM[1]
[End]
```

Chapter 7

TESTING A PROGRAM AND RUNNING PRODUCTION

Contents

Chapter 7	TESTING A PROGRAM AND RUNNING PRODUCTION	7–1
7.1	OVERVIEW	7–2
7.2	PROGRAM PAUSE AND RECOVERY	7–3
7.2.1	Program Pause and Recovery Overview	7–3
7.2.2	EMERGENCY STOP and Recovery	7–3
7.2.3	HOLD and Recovery	7–5
7.2.4	Setting Tolerance for Resuming a Program	7–6
7.2.5	Disable Fault Checking	7–11
7.3	TEST CYCLE	7–16
7.3.1	Test Cycle Overview	7–16
7.3.2	Test Cycle Setup	7–16
7.3.3	Single Step Testing	7–21
7.3.4	Continuous Testing	7–28
7.3.5	Monitoring Programs	7–34
7.4	RELEASE WAIT	7–34
7.5	PRODUCTION OPERATION	7–35
7.5.1	Production Operation Overview	7–35
7.5.2	Standard Operator Panel Cycle Start Production	7–38
7.5.3	User Operator Panel Start	7–39
7.5.4	Robot Service Request (RSR) Production Start	7–41
7.5.5	Program Number Select (PNS) and UOP Production Start	7–42
7.6	ADJUSTING A PROGRAM (PROG ADJUST)	7–44
7.6.1	Overview	7–44
7.6.2	Adjusting a Program or a Schedule	7–44
7.6.3	Error Handling	7–52
7.7	MAINTENANCE AND REPAIR	7–54

7.1 OVERVIEW

This section provides an overview of testing a program and running production.

Testing Operations

During test operations you should

- Run a program by
  - — **Stepping** through each line of the program
  - — **Continuously** running the program for a single cycle
  - — Enabling and disabling robot motion and the pelletizing, spot welding, arc welding, painting, or dispensing process during testing to verify each instruction of the program
- Monitor your program
- **Adjust program information** without stopping the program or production, if desired.
- Manually controlling wire feed and weld enable (ArcTool only)
- Performing test arc welds (ArcTool only)
- Performing test spot welds (SpotTool+ only)

Production Operations

During production operations you should

- Run a thoroughly tested program **continuously and repeatedly with all production conditions enabled**
- Perform maintenance procedures when necessary
- Perform automatic purge (DispenseTool only)
- Perform error recovery procedures when necessary
- Execute multiple programs

Note During testing and running production, you must know how to **stop the program** if there is a safety problem or adjustment to make, and know how to **restart the program.**

7.2 PROGRAM PAUSE AND RECOVERY

7.2.1 Program Pause and Recovery Overview

You can EMERGENCY STOP or HOLD any program running in production using

- The **EMERGENCY STOP button** on the teach pendant or operator panel
- The **HOLD button** on the teach pendant

Note If you are using ArcTool and if you have paused a running program, you can jog linear and circular motion instructions using the PATH jog coordinate system.

Note If you are using the Error Recovery option, you can perform error recovery procedures if an error occurs while you are running production.

If you have paused a running program, you can jog linear and circular motion instructions using the PATH jog coordinate system. Refer to Chapter 2 *TURNING ON AND JOGGING THE ROBOT* .

7.2.2 EMERGENCY STOP and Recovery

Press the EMERGENCY STOP button on the operator panel or teach pendant to stop the robot immediately. When you press the EMERGENCY STOP button you

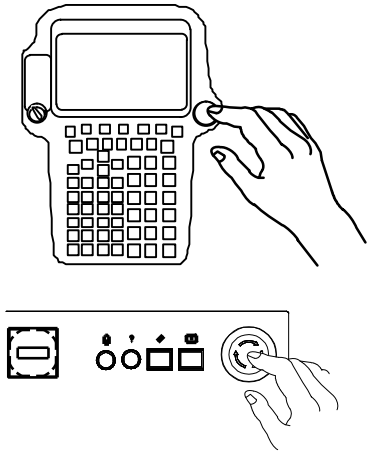
- Stop the running program
- Turn off drive power to the robot servo system
- Apply the brakes to the robot
- Stop the application

Use Procedure 7-1 to perform an EMERGENCY STOP. Use Procedure 7-2 to recover from an EMERGENCY STOP.

Procedure 7-1 EMERGENCY STOP

Step

1. Press the EMERGENCY STOP button on the teach pendant, or operator panel. This causes an EMERGENCY STOP fault.

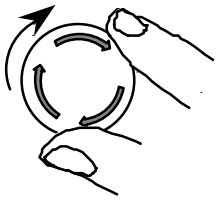


2. Fix the problem that caused you to press the EMERGENCY STOP button.

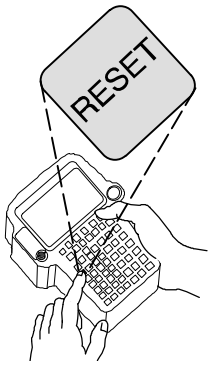
Procedure 7-2 Recovery from EMERGENCY STOP

Steps

1. Turn the EMERGENCY STOP button clockwise to release it.



2. Press the RESET button on the teach pendant, or Fault Reset on the operator panel.



3. Depending on the application you are using, press CYCLE START on the controller. The robot will resume operation using the Original Path Resume feature.

Note The Original Path Resume feature is used when resuming a stopped motion in a paused program. This feature allows the robot to remember the stop position and return to that position upon resuming, before continuing along the taught path. The motion to return to the stop position will be a linear, FINE termtype motion at 250 mm/sec.



Caution

The paused program must remain at the same line number as when the emergency stop occurred; otherwise, the original path resume feature will not be used to resume the program.

Note If using SpotTool+: when resuming production in auto mode, the previously paused program will be resumed, unless a user aborts the previously running production program.

Note For all tools except PaintTool: If the position where the robot resumes is outside of the range defined by the resume tolerance feature and you are in AUTO mode, a prompt box message will be displayed. You must either continue or stop and press ENTER.

7.2.3 HOLD and Recovery

Press the HOLD button on the teach pendant to pause a running program. Pressing the HOLD button

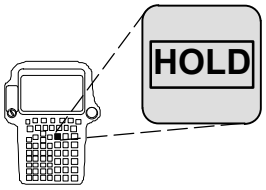
- Pauses a running program
- Causes the robot to decelerate and come to a controlled stop

Use Procedure 7-3 to recover from a held program.

Procedure 7-3 HOLD and Recovery

Steps

1. Press the HOLD button on the teach pendant.



2. Fix the problem that caused you to press HOLD.
3. Press the RESET button on the teach pendant to reset the fault.
4. Press CYCLE START to resume program execution. The program will resume using the Original Path Resume feature.

Note The Original Path Resume feature is used when resuming a stopped motion in a paused program. This feature allows the robot to remember the stop position and return to that position upon resuming, before continuing along the taught path. The motion to return to the stop position will be a linear, FINE termtype motion at 250 mm/sec.

Caution

The paused program must remain at the same line number as when the emergency stop occurred; otherwise, the original path resume feature will not be used to resume the program.

Note Hold is released when program execution begins.

7.2.4 Setting Tolerance for Resuming a Program

If you resume a program while in AUTO mode, and the distance between the resume position and the stop position is greater than the *stop tolerance*, a prompt box is displayed. When this occurs, you must perform specific actions to restart the program.

Stop tolerance is the amount of distance allowed between the resume robot position and the robot stop position. You can specify the following stop tolerances:

- **Distance tolerance** , for the location components of the position (x,y,z)
- **Orientation tolerance** , for the orientation components of the position (w,p,r)

- **Axes tolerances** for the joint angle (**rotary axes**) or distance location (**translational axes**) of any extended axes, if they are used.

- **Orientation tolerance**, for the orientation components of the position (w,p,r)
- **Axes tolerances** for the joint angle (**rotary axes**) or distance location (**translational axes**) of any extended axes, if they are used.

Operation

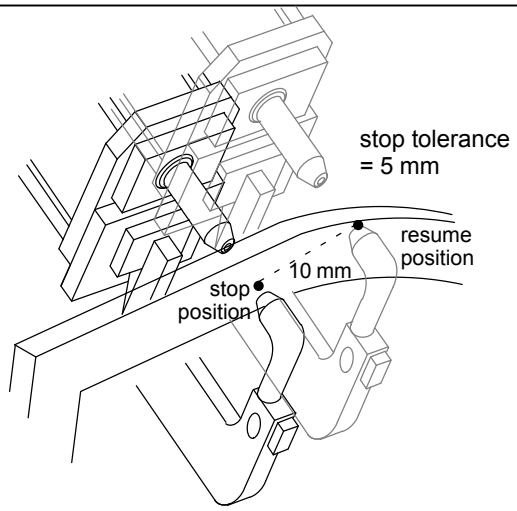
If you have set the tolerance, the following steps occur when the program is resumed. See Figure 7–1 .

1. A running program is paused. The position in which the robot stops is called the **stop position** .
2. The robot is moved to another position prior to resuming the program. This is called the **resume position** .
3. CYCLE START is issued to resume the program.
4. If the distance between the stop position and the resume position is **greater than** the stop tolerance
   a. A prompt box is displayed and the program is not resumed.
   b. You choose the next action:
      - Stop the program.

        Note In some applications Abort can be used in place of Stop.

      - Restart the program in its current position.
      - Change the mode to T1 or T2 and move the robot to another position by jogging it. Then, restart the program. The current position will be rechecked for tolerance.
5. If the distance between the stop position and the resume position is **less than** the stop tolerance, the program is resumed.

Figure 7–1. Resume Tolerance Example



- Distance between resume position and stop position = 10 mm
- Stop tolerance = 5 mm
- Resume position has exceeded the stop tolerance by 5 mm
- When you resume the program, the following screen will be displayed.

```
The robot position is out of
stop tolerance.
Please select action.
Choosing CONTINUE will
require cycle start.

      STOP      CONTINUE
```

Note In some applications Abort may be used in place of Stop.

Limitations

You cannot set tolerance for resume in the following cases:

- In line tracking programs
- In programs that use constant joint path (CJP)
- KAREL motion programs

Setting Up Tolerance for Resuming a Program

You can define the tolerances that will be used when programs run in AUTO mode are paused and then resumed. If you do not define the tolerances, the default values will be used.

Table 7–1 lists the tolerances you can set. Use Procedure 7-4 to set up tolerance for resuming a program.

Table 7–1. Tolerance Setup Items

ITEM	DESCRIPTION
Group Default: 1 Min: 1 Max: 8	This item is the motion group number of the axes for which you are setting tolerances. If you have a multiple motion group system, this is a number from 1 to 8. If you do not have a multiple motion group system, make sure Group is set to 1.
Enable Tolerance Checking Default: YES	This item specifies whether tolerances will be checked during program resume. • YES means axes will be checked for tolerance during program resume. • NO means axes will not be checked for tolerance during program resume.
Distance Tolerance Default: 250 mm Min: 0.1 mm Max: 1000 mm	This item specifies the location distance between the resume robot position and the position in which the robot stopped when the program was stopped.
Orientation Tolerance Default: 20 degrees Min: 0.1 degree Max: 80 degrees	This item specifies the angular or orientation distance between the resume robot tool approach vector and the stop position of the tool approach vector.
Axes Tolerance Rotational Axes Default: 20 degrees Min: 0.1 degree Max: 80 degrees	If the robot has a rotary extended axes, this item specifies the angular or orientation distance between the resume extended axes position and the corresponding extended axes of the robot stop position.
Axes Tolerance Translational Axes Default: 250 mm Min: 0.1 mm Max: 1000 mm	If the robot has a linear extended axes, this item specifies the location distance between the resume extended axes position and the corresponding extended axes of the robot stop position.

Procedure 7-4 Setting Up Tolerance for Resuming a Program

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select RESUME TOL. You will see a screen similar to the following.

```
SETUP RESUME TOL.
1  Group:                          1
2  Enable tolerance checking:      YES
3  Distance Tolerance (mm)         250.0mm
4  Orientation Tolerance (deg)     20.0deg
Axes Tolerance
5  Rotational axes (deg)           20.0deg
6  Translational axes (mm)         250.0mm
```

5. Select each item and set as desired.

Resuming a Program that Exceeds the Stop Tolerance

If you resume a program in which the current robot position exceeds the stop tolerance, you must perform specific procedures appropriate to the conditions. Use Procedure 7-5 to resume a program that exceeds the stop tolerance.

Procedure 7-5 Resuming a Program that Exceeds the Stop Tolerance

Conditions

- The robot is in AUTO mode.
- The program has been paused. You will see a screen similar to the following.

```
The robot position is out of
stop tolerance.
Please select action.
Choosing CONTINUE will
require cycle start.
```

Steps

1. Select the appropriate action:
   - To **abort** , go to Step 2 .
   - To **continue** , go to Step 3 .
   - To **move the robot back into tolerance and restart** , go to Step 4 .
2. **To Stop the program,** move the cursor to STOP, and press ENTER. To abort the program, press the FCTN key and select " Abort All."

The program is aborted.

Note In some applications Abort may be used in place of Stop.

3. **To continue the program from the current position,**
   a. Move the cursor to CONTINUE, and press ENTER.

      The program is still paused.

   b. Input the start signal again to restart the program at its present position.

      The robot will move from the current position to the stop position and continue the program. The robot does not check whether it is out of tolerance, and the prompt box is not displayed again.

      Note If the restarted program is the paused program, the appropriate recovery functions are executed. Refer to Section 10.19 for more information.

4. **To move the robot into tolerance and restart** the program,
   a. Set the MODE SELECT switch to the T1 or T2 position.
   b. Jog the robot to a position that is within the tolerance.
   c. Set the MODE SELECT switch to the AUTO position.
   d. Input the start signal again to restart the program at its present position.

      The robot will check again whether it is out of tolerance. If it is out of tolerance, the prompt box is displayed again. Repeat Step 4a through Step 4d until the robot is within the resume tolerance.

7.2.5 Disable Fault Checking

During program execution, certain alarms involving inputs are monitored. If one of these alarms occurs, you will be given a choice of whether to disable the fault for a certain number of program execution cycles.

If you have SpotTool+, or HandlingTool with the "MH RS4 Compliance" option , you will see a screen similar to the one shown in Figure 7–2 . If you press F4, [CHOICE], and then select Disable Alarm, the fault that is being disabled is automatically added to the Disabled Faults screen.

Figure 7–2. ALARM Recovery Without Specific Error and Action Text

```
ALARM Recovery
  ERROR:   Part Present  1  is not
           reading the part.
  ACTION:  Check the wiring.
           Press F4 [CHOICE] to select
           recovery option.
Perform action, then press [CHOICE].
```

Disabled Faults Screen

The Disabled Faults screen gives a status of the alarms that are currently disabled. You can also use the Disabled Faults screen to clear the disabled faults manually.

The fault table is cleared (no faults are disabled) when you Cold start the controller. Otherwise, faults are not cleared automatically and the fault count will continue to increase. You can change the fault count in the Status Disabled Faults screen. A fault's count is incremented at the beginning of each job cycle after the fault is added to the table.

All faults use the same maximum fault count. At the end of each job cycle, every fault count in the fault table is incremented by one. Each time, when certain faults occur, the table is checked:

- If the fault is in the table and the count is less than the current maximum number of cycles disabled, the alarm is ignored.
- If the count is greater than the current maximum number of cycles disabled, the alarm is displayed.
- If the fault is not in the table, the Alarm Recovery screen is displayed with disable alarm as one of the choices.

Table 7–2 lists and describes the items that are displayed on the Status Disabled Faults screen. Use Procedure 7-6 to display the Status Disabled Faults screen and to clear faults manually using this screen.

Table 7–2. Status Disabled Faults Screen Items

ITEM	DESCRIPTION
FAULT NAME	This item displays the name of the fault. You cannot change the fault name on this screen.
TYPE	This item displays the type of input signal that failed. You cannot change the type on this screen.
#	This item displays the port number of the input signal that failed. You cannot change the # on this screen.

Table 7–2. Status Disabled Faults Screen Items (Cont'd)

ITEM	DESCRIPTION
SIM	This item displays the simulated status of the input signal that failed: • **U** is unsimulated • **S** is simulated You cannot change the simulated status on this screen.
STATUS	This item displays the current status of the input signal that failed. You cannot change the status on this screen.
CNT	This item displays the number of cycles that have run since this fault was disabled. You can change the cycle count on this screen.
Configuration Menu	
Disabled Faults Option Default: ENABLED	This item allows you to specify whether you want to use the Disabled Faults option: • **DISABLED** means that the Disabled Faults option will not be used. • **ENABLED** means that the Disabled Faults option will be used.
Max Cycles Disabled Min: 0 Max: 9999 Default: 20	This item allows you to specify the number of cycles a fault will be disabled. If you set this to 0, the fault will be posted every time it occurs. If you set this to a number other than 0, the fault will be posted only when the number of cycles equals the Max Cycles Disabled you specified.

Faults that can be Disabled

You can only disable specific faults, which are shown in Table 4–31 .

Note You can only disable a fault after it has occurred, and an error message has been displayed.

Table 7–3. Faults that can be Disabled

Material Handling Gripper	PP*n* not read (*n* = 1 through 10)
	PP*n* still read (*n* = 1 through 10)
	Clp*n* not open (*n* = 1 through 20)
	Clp*n* still cls (*n* = 1 through 20)
	Clp*n* not clsd (*n* = 1 through 20)
	Clp*n* still opn (*n* = 1 through 20)
	Vac*n* not on (*n* = 1 through 2)
	Vac*n* not off (*n* = 1 through 2)

If you are using SpotTool+, use Table 7–4 for information regarding faults that can be disabled.

Table 7–4. SpotTool+ Faults that can be Disabled

Spot Welding	Water saver OK
	Wtr flow OK
	X-former
	Gun detect OK
	Backup detect

Procedure 7-6 Disabling Fault Checking

Steps

1. Press MENUS.
2. Select STATUS.
3. Press F1, [TYPE].
4. Select DisableFault. You will see a screen similar to the following.

```
Status Disabled Faults
     FAULT NAME    TYPE #   SIM STATUS CNT
1 PP 3 still read  DI[ 10]    U  OFF    20
2 Clp 5 not open   DI[ 11]    U  ON     15
3                  DI[  0]    U  OFF     0
4                  DI[  0]    U  OFF     0
5                  DI[  0]    U  OFF     0
6                  DI[  0]    U  OFF     0
```

5. Review the fault status.
6. **To clear a single fault**,
   a. Move the cursor to the fault you want to clear.
   b. Press F3, CLEAR.
7. **To clear all of the faults listed**,
   a. Press NEXT, >.
   b. Press F3, CLEARALL.
8. **To enable or disable the Disabled Faults option**,
   a. Press F2, CONFIG. You will see a screen similar to the following.

```
Status Disable Faults
Configuration Menu
  1  Disable Faults Option:      ENABLED
  2  Max Cycles Disabled:        20
```

   b. Move the cursor to Disable Faults Option.
   c. Press the appropriate key:

   - **To enable** the Disable Faults option, press F4, ENABLE.
   - **To disabled** the Disable Faults option, press F5, DISABLE.

9. **To set the maximum number of cycles a fault will be disabled**,
   a. Move the cursor to the fault you want.
   b. Press F2, CONFIG. You will see a screen similar to the following.

```
Status Disable Faults
Configuration Menu
  1  Disable Faults Option:      ENABLED
  2  Max Cycles Disabled:        20
```

   c. Move the cursor to Max Cycles Disabled.

d. Type the number of cycles you want and press ENTER.

7.3 TEST CYCLE

7.3.1 Test Cycle Overview

You must test your program before you run production. For each step in the following test plan you must set up test cycle conditions as appropriate for the kind of testing you are performing.

A typical testing plan will

1. Single step through the program using the teach pendant while the process is disabled to check the robot motion, other instructions, and I/O.
2. Continuously run the program using the teach pendant at a low speed with the process disabled.
3. Continuously run the program using test cycle at high speed with the process disabled to check the robot positions and timing.
4. Depending on the application you are using, continuously run the program using the operator panel with the process enabled to verify the process.
5. Continuously run the program, using the operator box or operator panel, at a high speed, with the process enabled, to check the robot positions and timing.

Note You can change test cycle conditions only if a program is not running.

Note If you are using ArcTool, during testing, you might want to use the on-the-fly utility. On-the-fly permits real time editing of schedule data during program execution.

Note If you are using SpotTool+, you cannot execute a SpotTool+ program that has been created prior to software version 4.10 unless you have converted it to the appropriate format.

7.3.2 Test Cycle Setup

Setting up the test cycle allows you to control the conditions for test running a program. These conditions are in effect any time a program is run until you change the conditions.

You can set the test cycle conditions listed and described in Table 7–5 . Use Procedure 7-7 to set up test cycle conditions.

Table 7–5. Test Cycle Conditions

Test Cycle Condition	Description
Group	This item specifies the motion group number of the program for which the test cycle conditions are being set.
Group Motion	This item determines whether the robot will move during the test cycle. If set to **ON**, the **robot will move**. If set to **OFF**, the brakes are set, servo power is turned off, and the **robot will not move**.
Dry Run	This item is only displayed if $AP_DSPDRYRN=TRUE. This item determines whether the EOAT will function during the test cycle. If set to **OFF** , the **EOAT will function**. If set to **ON**, the **EOAT will not function**.
Cartesian Dry Run Speed	This item is only displayed if $AP_DSPDRYRN=TRUE. This item determines the speed at which the robot will move during program execution when using Cartesian motion (linear or circular moves) when dry run is set to ON.
Joint Dry Run Speed	This item is only displayed if $AP_DSPDRYRN=TRUE. This item determines the speed at which the robot will move during program execution when using joint motion when dry run is set to ON. A joint dry run speed of 100 will test run the program at 100% of the programmed speed.
Jog Dry Run Speed	This item is only displayed if $AP_DSPDRYRN=TRUE. This item determines the speed at which the robot will jog when dry run is set to ON. A jog dry run speed of 100% indicates that any jogging done during dry run test cycle will be at the normal jog speed. A jog dry run speed of less than 100% indicates that any jogging done during dry run test cycle will be reduced by that percentage. The jog dry run speed is independent of the Cartesian dry run speed and the Joint dry run speed.
Digital/Analog I/O	This item determines whether digital/analog input and output signals will turn on and off during the test cycle. If set to **ENABLED**, the input and output signals **will turn on and off**. If set to **DISABLED**, the input and output signals will not function.

Table 7–5. Test Cycle Conditions (Cont'd)

Test Cycle Condition	Description
Step Statement Type	This item allows you to select at which statements the robot will pause between steps. There are the following statement types: • **TPP LINE OR KAREL STATEMENT** - The program pauses when the execution of each program is completed. The program pauses after executing each step in a routine. • **MOTION** - The program pauses when the execution of each motion statement is completed. • **ROUTINE** - The program pauses after each statement is executed. However, when executing a routine, the program pauses only after every motion statement and when returning from the routine. • **STATEMENT** - The program pauses when the execution of each program statement is completed. The program pauses after executing each step in a routine. • **TP & MOTION** - The program pauses when the execution of a teach pendant statement or when a KAREL motion statement is completed.
Step Path Node	This item specifies whether to pause the program after each path node during a KAREL MOTION ALONG statement. When set to ON, the program will pause after each path node. When set to OFF, each path will be executed as one continuous step.

The Test Cycle Spot menu allows you to set up test run conditions that are specific to SpotTool+. These conditions are in effect any time a program is run until you change the conditions.

Table 7–6. Test Cycle Spot Description

TEST CYCLE SPOT ITEM	DESCRIPTION
Gun operation Cannot be changed if a program is running	This item allows you to select the type of gun mode that will be used when running the program. • **STROKE** - opens and closes the gun at each weld point. Passing current depends on the setting of weld controller mode. • **NOSTROKE** - does not close and open the gun, and does NOT pass current through the gun. Backup stroke will still function. The weld controller mode must be set to NOWELD.

Table 7–6. Test Cycle Spot Description (Cont'd)

TEST CYCLE SPOT ITEM	DESCRIPTION
Weld controller mode Cannot be changed if a program is running	This item allows you to select the type of weld mode that will be used when running the program • **WELD** - when the gun closes, the weld controller executes the appropriate schedule and passes current through the gun. • **NOWELD** - no current passes through the gun when the weld controller executes the weld sequence.

Tryout mode
$SHELL_WRK.$tryout_mode
If the tryout input is not defined ($CELL_SETUP.$di_tryout _i=0), you cannot change the tryout_mode state.
 	
If the tryout mode input is defined ($CELL_SETUP.$di_tryout _i > 0) and the TP is enabled, you can change the tryout_mode state via the softpanel screen (or direct sysvar manipulation).
If the tryout mode input is defined ($CELL_SETUP.$di_tryout _i > 0) and the TP is disabled, you cannot change the tryout_mode state via the softpanel screen. Set or reset DIN[$CELL_SETUP.$di_tryout _i] to change the tryout_mode state in this case.
	This item enables and disables the ability of the robot to execute material handling commands in a tryout, or dry run mode. Tryout Mode is controlled by the input signal in Auto Mode. The Enabled setting also internally sets the part thickness to 0 for Servo Guns to prevent pressure shortage alarms.
	• **ENABLED** - The robot goes into tryout mode and operates all material handling outputs but ignores all part present inputs.
	• **DISABLED** - The robot is not in tryout mode.
	
	The Tryout cell output signal reports the status of tryout mode.
	
	In addition, you can set Tryout mode within a program, using the system variable or the Tryout cell output signal, as follows:
	
	`If $SHELL_WRK.$TRYOUT_MODE = 1 or`
	`DOUT[MH_TRYOUT] = ON`
	`JUMP LABEL xxxx`

Table 7–6. Test Cycle Spot Description (Cont'd)

TEST CYCLE SPOT ITEM	DESCRIPTION
Force process complete	This item allows you to force the process complete signal. • **ENABLED** - Process Complete is turned on immediately. Process Complete will be cleared at the start of the next style. The Process Alert signal is turned on whenever Force Process Complete is enabled, and a warning error is displayed on the ALARMS screen to indicate that this abnormal operation was performed. • **DISABLED** - Process Complete is turned off.
Update weld pressures	If the weld controller controls weld pressure (the Controls weld pressure item is enabled at controlled start), Update weld pressures is displayed. • **ENABLED** - This will force an update of the weld pressure table from the weld controller. • **DISABLED** - Update Weld Pressures is turned off

The Test Cycle Dispense menu allows you to set up test run conditions that are specific to dispensing applications. These conditions are in effect any time a program is run until you change the conditions.

Procedure 7-7 Setting Up Test Cycle Conditions

Steps

1. Press SELECT.
2. Select the program you want to test and press ENTER.
3. Press MENUS.
4. To set up **general** Test Cycle conditions, select **Setup**. You will see a screen similar to the following.

```
TEST CYCLE Setup
Group 1
 1 Group Motion:            DISABLE
 2 Digital/Analog I/O:      ENABLE
 3 Step statement type:     MOTION
 4 Step path node:          OFF
```

5. To set up additional Test Cycle conditions for **spot welding**, select **Spot Weld**. You will see a screen similar to the following.

```
TEST CYCLE SPOT
                                1/7
 1 Gun Operation:            STROKE
 2 Weld controller mode:      WELD
 3 Tryout mode:             DISABLED
 4 Weld I/O sim:            ENABLED
 5   Weld Duration (ms):    500   ms
 6   Inform output to plc    DO[  0]
 7 Force process complete: DISABLED
```

6. To set up additional Test Cycle conditions for **dispensing**, select **Dispense**. You will see a screen similar to the following.

```
TEST CYCLE DISPENSE
                                2/2
 1 Dispense controller mode:  DRY
 2 Force process complete:    ENABLED
```

7. **To display help information** , press NEXT,>, and then press F1, HELP. When you are finished displaying help information, press PREV.
8. Set Test Cycle conditions as desired.

   Note You can change test cycle conditions only if there is not a program running.

   Note If you are using ArcTool, and if you disable I/O from the TEST CYCLE SETUP screen, I/O might appear to be simulated when it actually is not. For simulation to occur, you must enable I/O on the TEST CYCLE SETUP screen.

7.3.3 Single Step Testing

Single step testing is the process of running individual program instructions one at a time. You use the teach pendant to single step the current program displayed on the teach pendant screen.

The MODE SELECT switch must be in the T1 or T2 position to test a program in single steps using the teach pendant. If you test a program in T1 mode, the robot speed will be no greater than 250mm/sec, regardless of any other speed settings. Refer to Section 1.3.4 for more information on the MODE SELECT switch.

If you set the singularity stop system variable, $PARAM_GROUP[n].$T1T2_SNGSTP, to TRUE, the robot will stop at singularity points while in T1 or T2 mode. If you change the value of this variable, you must cycle power for the change to take effect.

Note During single step testing, the @ indicator will be displayed on the screen when the robot reaches the end of the motion. The @ indicator will be displayed on all lines in which the current position is approximately equal to the taught position.

Single step testing can be done two ways:

- Forward
- Backward

Note If you are using ArcTool, using single step testing turns off tracking. Do not use single step testing during tracking because it will cancel tracking on the next motion instruction, and the desired motion will not be obtained for the next resumed motion. Also, Arc welding is never enabled when single stepping through a program regardless of the conditions set in Test Cycle or whether Weld Enable is set to ON.

Note You cannot use backward execution to call the sub program from the main program.

- Does not perform any SPOT/BACKUP commands regardless of STROKE/NOSTROKE/WELD/NOWELD. Only the motion portion of the instruction is executed in BWD.

Forward

- Executes the next instruction when the SHIFT and FWD keys are pressed and FWD is released.
- Stops when the step is completed or SHIFT is released.
- Executes subprograms one step at a time.

Backward

- Executes the previous instruction when the SHIFT and BWD keys are pressed and then BWD is released.
- Steps backward from a sub-program to the main program. Before you can do this, you must step forward from the main program to the sub-program. When you return to the main program from the sub program, the cursor pauses on the CALL instruction in the sub program.
- Stops when the step is completed or the SHIFT key is released.
- Can only be done for motion instructions.

Note You cannot use backward execution to call the sub program from the main program.

Note If you are using DispenseTool, dispensing is**never** enabled when single stepping through a program regardless of the conditions set in Test Cycle.

Figure 7–3 contains an example program which shows how to perform backward execution from the 4th line of the sub program SUB_PROG.

Figure 7–3. Example Program Showing Backward Execution

```
MAIN_PROG
1:
2: R[1] = R[1] + 1
3: J P[1] 100% FINE
4:
5: CALL SUB_PROG
6:
[END]
SUB_PROG
 1: SDO[1] = ON
 2: SDO[2] = ON
 3: L P[2] 1000mm/sec FINE
 4: L P[3] 1000mm/sec FINE
 [END]
```

1. Start to do the backward execution from 4th line of the SUB_PROG.

   Motion: The robot moves from P[3] to P[2].

   Cursor: The cursor is on the 3rd line of the SUB_PROG.

2. Do the backward execution again.

   Motion: No motion.

   Cursor: The cursor is on the 5th line of the MAIN_PROG.

3. Do the backward execution again.

   Motion: The robot moves from P[2] to P[1].

   Cursor: The cursor is on the 3rd line of the MAIN_PROG.

DISABLE FWD/BWD

DISABLE FWD/BWD allows you to disable the ability to execute program instructions when the SHIFT and FWD keys or SHIFT and BWD keys are pressed. To use DISABLE FWD/BWD, press

FCTN and then select DISABLE FWD/BWD. The ability to use SHIFT FWD and SHIFT BWD will be disabled until you press FCTN and select DISABLE FWD/BWD again.

When the teach pendant FWD and BWD keys are disabled and the teach pendant is enabled, "FBD" is displayed in the upper left hand corner of the teach pendant screen to indicate that you cannot use the teach pendant to run the program.

Use Procedure 7-8 to single step test a program.

Procedure 7-8 Single Step Testing

Note If the MODE SELECT switch is in the T1 position, the robot speed will be no greater than 250mm/sec, regardless of any other speed settings.

Note You can perform single step testing from the teach pendant only with the MODE SELECT switch in the T1 or T2 position.

Note Refer to the *SpotTool+ Setup and Operations Manual* for information on enabling or disabling the Joint Motion Warning used with DispenseTool.

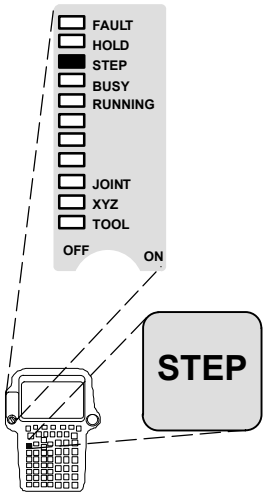
Conditions

- A program has been created and positions have been recorded.
- Test cycle conditions have been set. Refer to Procedure 7-7 .
- All personnel and unnecessary equipment are out of the workcell.
- The MODE SELECT switch is in the T1 or T2 position.

Note If you are using ArcTool, using single step testing turns off tracking. Do not use single step testing during tracking because it will cancel tracking on the next motion instruction, and the desired motion will not be obtained for the next resumed motion.

Steps

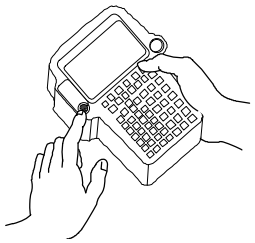
1. Press SELECT.
2. Select the program you want to test and press ENTER.
3. Press STEP to enable single step testing. The STEP indicator will turn on.



4. Move the cursor to the first line of the program you want to test. The program will start at the current cursor position. Depending on the application you are using, you will see a screen similar to the following.

```
PRGWELD
  1  J P[1] 100% CNT50
  2: PALLET[1 :pallet_sample]
  3: J P[1:A_1] 70% CNT100
  4: J P[1:A_2] 50% CNT50
  5: L P[1:A_3] 500mm/s FINE
  6: L P[1:BTM] 300mm/s FINE
  7:   CLOSE HAND [1]
  8:   WAIT SDI[1]
  9: If PL[3]=[5,*,*] JUMP[11]
  [End]
```

5. Continuously press the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.



Note If you compress the DEADMAN switch fully, robot motion will not be allowed and an error occurs. This is the same as when the DEADMAN switch is released. To clear the error, press the DEADMAN switch in the center position and press RESET.

6. Set the speed override or test cycle speed to the value you want. A low speed is recommended.

7. Check the program status on the top line of the teach pendant screen. If it is PAUSED, press FCTN and select ABORT (ALL).

Warning

The next step causes a program instruction to run. This could cause the robot to move and other unexpected events to occur. Make sure all personnel and unnecessary equipment are out of the workcell and that all safeguards are in place; otherwise, you could injure personnel or damage equipment.

In the next step of this procedure, if you want to stop the program instruction before the instruction has finished executing, release the SHIFT key, release the DEADMAN switch, or press the EMERGENCY STOP button.

Warning

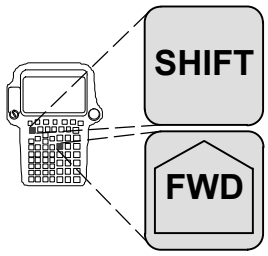
If you execute motion instructions that contain the remote TCP (RTCP) motion option and you skip motion instructions during testing, the robot might have to change orientation dramatically to reach the destination position. This will cause it to move in a large area. Be aware that this might happen before you skip motion instructions during testing; otherwise, you could injure personnel or damage equipment.

Note If you have set the singularity stop system variable, $PARAM_GROUP[n].$T1T2_SNGSTP, to TRUE, the robot will stop at

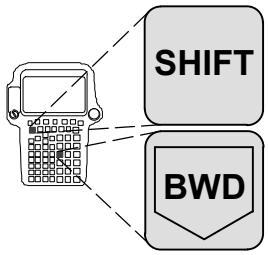
singularity points while in T1 or T2 mode. If you change the value of this variable, you must cycle power for the change to take effect.

8. Test a program instruction.

- **To execute an instruction in the forward direction,** press and hold in the SHIFT key and press and release the FWD key. You must hold in the SHIFT key continuously until the instruction has finished executing.



- **To execute an instruction in the backward direction,** press and hold in the SHIFT key and press and release the BWD key. You must hold in the SHIFT key continuously until the instruction has completed executing.

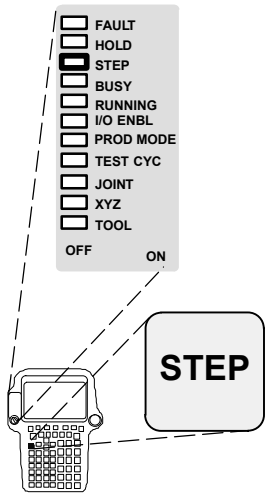


- When the robot reaches the end of the motion, the @ indicator will be displayed as shown in line 4 below.

```
PRGWELD
  1  J P[1] 100% CNT50
  2: PALLET[1 :pallet_sample]
  3: J P[1:A_1] 70% CNT100
  4: J@P[1:A_2] 50% CNT50
  5: L P[1:A_3] 500mm/s FINE
  6: L P[1:BTM] 300mm/s FINE
  7:   CLOSE HAND [1]
  8:   WAIT SDI[1]
  9: If PL[3]=[5,*,*] JUMP[11]
  [End]
```

9. Repeat Step 8 for as many instructions as you want to test. When the robot reaches the end of the motion, the @ indicator will be displayed on the screen indicating the robot is approximately at the new position.

10. Press STEP to disable single step testing. The STEP indicator will turn off.



11. Turn the teach pendant ON/OFF switch to OFF and release the DEADMAN switch.

7.3.4 Continuous Testing

Continuous testing is running a program from beginning to end without stopping. You can test a program continuously using the teach pendant, operator box, or operator panel CYCLE START button.

To test a program continuously using the teach pendant the MODE SELECT switch must be in the T1 or T2 position. To test a program continuously using the CYCLE START button on the operator panel, the MODE SELECT switch must be in the AUTO position. Refer to Section 1.3.4 for more information on the MODE SELECT switch.

If you set the singularity stop system variable, $PARAM_GROUP[n].$T1T2_SNGSTP, to TRUE, the robot will stop at singularity points while in T1 or T2 mode. If you change the value of this variable, you must cycle power for the change to take effect.

Use Procedure 7-9 to test continuously using the teach pendant. Use Procedure 7-10 to test a program continuously using the operator panel CYCLE START button.

Procedure 7-9 Continuous Testing Using the Teach Pendant

Note If the MODE SELECT switch is in the T1 position, the robot speed will be no greater than 250mm/sec, regardless of any other speed settings.

Note You can perform continuous testing from the teach pendant only with the MODE SELECT switch in the T1 or T2 position.

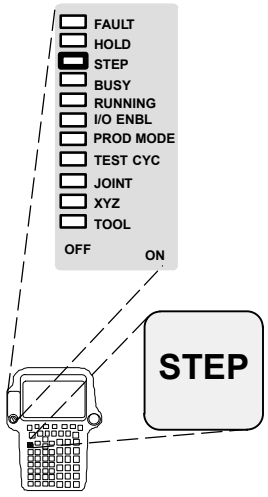
Note Refer to the*SpotTool+ Setup and Operations Manual* for information on enabling or disabling the Joint Motion Warning used with DispenseTool.

Conditions

- A program has been created and positions have been recorded.
- Test cycle conditions have been set. Refer to Procedure 7-7 .
- All personnel and unnecessary equipment are out of the workcell.
- You have tested the program in single steps. (Procedure 7-8)
- The MODE SELECT switch is in the T1 or T2 position.
- The Remote/Local Setup item in the System Configuration Menu is set to REMOTE. Refer to Section 3.15 .

Steps

1. Press SELECT.
2. Select the program you want to test and press ENTER.
3. Disable single step testing. If the STEP indicator is ON, press STEP to disable it.



4. Move the cursor to line 1. The program will start at the current cursor position.
5. Continuously press the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.

Note If you compress the DEADMAN switch fully, robot motion will not be allowed and an error occurs. This is the same as when the DEADMAN switch is released. To clear the error, press the DEADMAN switch in the center position and press RESET.

6. Set the speed override to the value you want. First run the speed at 5% - 10% value.
7. Check program status on the top line of the teach pendant screen. If it is PAUSED, press FCTN and select ABORT (ALL).

Warning

The next step causes a program to run. This could cause the robot to move, the spot welding process to run, and other unexpected events to occur. Make sure all personnel and unnecessary equipment are out of the workcell and that all safeguards are in place; otherwise, you could injure personnel or damage equipment. In the next step of this procedure, if you want to stop the program instruction before the instruction has finished executing, release the SHIFT key, release the DEADMAN switch, or press the EMERGENCY STOP button.

Warning

If you execute motion instructions that contain the remote TCP (RTCP) motion option and you skip motion instructions during testing, the robot might have to change orientation dramatically to reach the destination position. This will cause it to move in a large area. Be aware that this might happen before you skip motion instructions during testing; otherwise, personnel could be injured and equipment damaged.

Note You can test a program continuously in the forward direction only.

Note If you have set the singularity stop system variable, $PARAM_GROUP[n].$T1T2_SNGSTP, to TRUE, the robot will stop at singularity points while in T1 or T2 mode. If you change the value of this variable, you must cycle power for the change to take effect.

8. Press and hold down the SHIFT key and press and release the FWD key. You must hold in the SHIFT key continuously until the instruction has finished executing.

   Run the program in 5% - 10% intervals, up to 100%. If positions with continuous termination type are changed, restart the process at a low speed.

9. Turn the teach pendant ON/OFF switch to OFF and release the DEADMAN switch.

Procedure 7-10 Continuous Testing Using the Operator Panel CYCLE START Button

Note You can perform continuous testing using the CYCLE START button only with the MODE SELECT switch in the AUTO position.

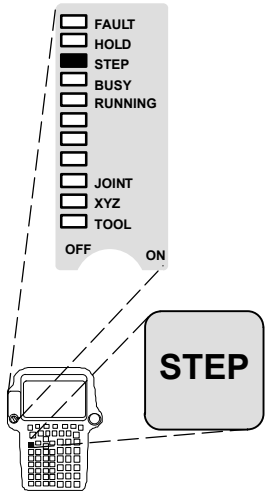
Conditions

- The program has been created and positions recorded.
- Test cycle conditions have been set.
- All personnel and unnecessary equipment are out of the workcell.
- You have tested the program in both single step (Procedure 7-8) and continuous (Procedure 7-9) using the teach pendant.
- The MODE SELECT switch is in the AUTO position.
- The Remote/Local setup item on the system config menu is set to LOCAL. Refer to Section 3.15 .

Steps

1. Press SELECT.

2. Disable single step testing. If the STEP indicator is on, press STEP to turn it off.

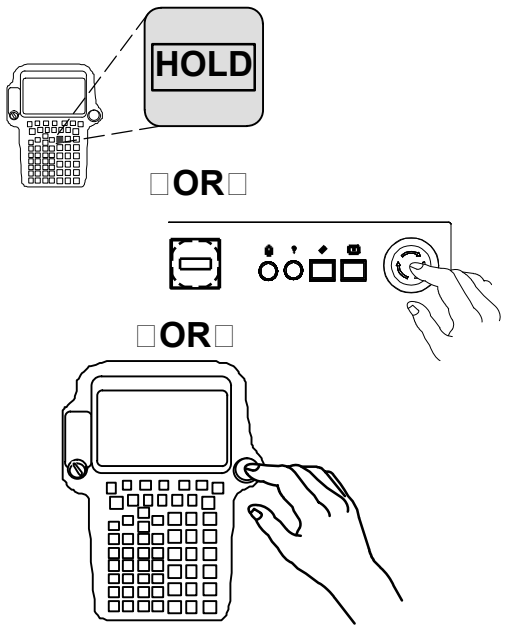


3. Select the program you want to test and press ENTER.

4. Set the speed override to the value you want.

When CYCLE START is used, the speed override might automatically be set to 100%.

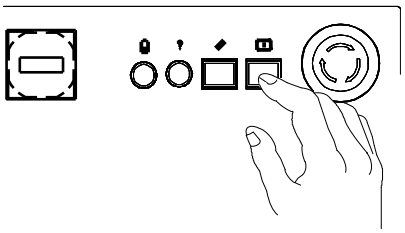
> **Warning**
>
> **The next step causes a program to run. This could cause the robot to move, the spot welding process to run, and other unexpected events to occur. Make sure all personnel and unnecessary equipment are out of the workcell and that all safeguards are in place; otherwise, personnel could be injured and equipment damaged. If you want to stop the program before it has finished executing, press the HOLD button for a decelerated stop, or the EMERGENCY STOP button for an immediate stop.**



Warning

If you execute motion instructions that contain the remote TCP (RTCP) motion option and skip motion instructions during testing, the robot might have to change orientation dramatically to reach the destination position, causing it to move in a large area. Be aware that this might happen before you skip motion instructions during testing; otherwise, you could injure personnel or damage equipment.

5. Press the CYCLE START button on the operator panel . You will be prompted to confirm program execution. If you select YES, then you must press CYCLE START again to run the program.



7.3.5 Monitoring Programs

You can monitor a running program from the SELECT menu. When you monitor a running program, the program is displayed and the cursor highlights the line currently being executed.

Use

Procedure 7-11 Monitoring a Running Program

Conditions

- The program you want to monitor is currently executing.

Steps

1. Press SELECT.

```
Select
                    50983 BYTES FREE
 No.  Program name              Comment
  1    SUB1               [               ]
  2    MAIN25             [               ]
  3    PRG7               [               ]
  4
  5
  6
```

2. Press F4, MONITOR.The program will be displayed on the screen. The cursor will be on the line number of the instruction that is currently being executed. The cursor will move to each instruction as it is executed.
3. **To look at another area of the program while the program is being displayed,** press F2, LOOK. When you want the cursor to return to the line number of the instruction currently being executed, press F2, MONITOR.

7.4 RELEASE WAIT

During program execution, wait release allows you to override pauses in the program when the robot is waiting for I/O conditions to be satisfied. Release wait works only when a program is running.

Perform Procedure 7-12 to use release wait.

Procedure 7-12 Using Release Wait

Conditions

- A program is running.
- The running program is waiting for I/O conditions to be satisfied.

Steps

1. Press the FCTN key.

> **Warning**
>
> **Be careful when you use wait release. When you override I/O or override wait periods, you can cause the robot to move or equipment to operate unexpectedly.**

2. Select RELEASE WAIT.
   - **If an active wait is pending,** the program will pause. Resume the program when you are ready, using the method you used to run the program.
   - **If no active wait is pending,** nothing will happen.

7.5 PRODUCTION OPERATION

7.5.1 Production Operation Overview

Production operation is automatic execution of a designated program or series of programs. The program runs continuously and repeatedly with full speed, the current application, I/O, and motion conditions enabled.

The MODE SELECT switch must be in the AUTO position to perform production operation. Refer to Section 1.3.4 for more information on the MODE SELECT switch.

If you set the singularity stop system variable, $PARAM_GROUP[n].$AUTO_SNGSTP, to FALSE, the robot will pass through singularity points while in AUTO mode.

Depending on the application you are using, you can run production using the following methods:

Table 7–7. Program Select and Program Start

Program Select Method	Std. SOP Cycle Start SI[6]	Std. UOP Production Start UI[18] pr *Start	Production Start Method		
			UOP	DIN	Other
System Config:	Remote/Local: Local	Remote/Local: Remote *Start for Continue Only: FALSE	Remote/Local: Remote * Start for Continue only: FALSE	Remote/ Local: Remote	N/A
Select Menu (Manual Selection)	Run	Run	Run Production start UI[18] or *Start UI[6]	Run PNS Start DI[#] (If Program select is PNS, using DIN Production Start method)	N/A
RSR	Run : If RSR program is already selected.	Run : If RSR program is already selected.	Select & Run : RSR Input UI[9-16]	Select & Run : RSR Input DI[#] #Eight user selected DI Inputs	N/A
PNS	Run : If PNS program is already selected.	Run : If PNS program is already selected (No - if Production Start method is DIN).	Select : PNS Input UI[9-16] + PNStrobe UI[17] Run : Production start UI[18] or *Start UI[6]	Select : PNS Input GI[#] + PNS DIN Strobe DI[#] Run : PNS Start DI[#] #User selected DI / GI Inputs.	N/A
Style	Run : If Style program is already selected.	RUN : If Style program is already selected.	Select :Style Select UI[9-16] Run : Production start UI[18] or Start UI[6] *	Select : Style Select GI[#] Run : Initiate Style DI[#] #User selected DI / GI Inputs.	Select : Style Select GI[#] Run : Change the value of $SHELL_WRK. $CUST_START @@ from false to true.

Table 7–7. Program Select and Program Start (Cont'd)

Program Select Method	Std. SOP Cycle Start SI[6]	Std. UOP Production Start UI[18] pr *Start	Production Start Method		
			UOP	DIN	Other
Other	Run : If Other (see @ below) program is already set.	Run : If Other (see @ below) program is already set.	Select & Run : Production start UI[18] or Start UI[6] * If another program is already selected, the program defined in $SHELL_WRK. $CUST_NAME (see @) is selected and run.	Select & Run : Initiate Style DI[#] If another program is already selected, the program defined in $SHELL_WRK. $CUST_NAME (see @) is selected and run.	Select : Change the value of $SHELL_WRK. $CUST_NAME@ to the program name you need to run. Run : Change the value of $SHELL_WRK. $CUST_START @@ from false to true.
@- When Program Select method is " Other", you may specify the program to run by setting the program name in the Program Select setup screen, Program Select method, by pressing F3, DETAIL. This sets the system variable $SHELL_WRK.$CUST_NAME. The Production Start method, UOP or DIN, will only execute this program. This allows a PC to select the program remotely. @@- When the Production Start method is set to "Other", execution is initiated by toggling the variable $SHELL_WRK.$CUST_START to True. The system will change it back to FALSE once the program has started. This allows PC interfaces to start production.					

- SOP (Standard Operator Panel) CYCLE START
  - — Program Number Select (PNS)
  - — Robot Service Request (RSR)
  - — Style Program Select (optional)
  - — OTHER Program Select Mode
- Production Start method
  - — UOP PRODUCTION START input
  - — DIN[ ]
  - — OTHER
- Default program on SELECT menu method
  - — SOP CYCLE START input

Note If you are using ArcTool during production , you might want to use the ON-the fly utility. On-the-fly permits real time editing of weld and weave schedule data during program execution.

Note If you are using the Material Handling Shell option, refer to the "Material Handling Shell (Option)" chapter in the *Setup and Operations Manual* for production operation information.

7.5.2 Standard Operator Panel Cycle Start Production

A*Standard Operator Panel Cycle Start* is a method of automatically running the selected program. If you select Cycle Start, it implies that you are not using RSR or PNS, but will use the CYCLE START input on the Standard Operator Panel (SOP) to initiate production operation. You select the program to be run on the teach pendant SELECT menu.

The CYCLE START input will run the selected program for one cycle unless

- The program contains a loop.
- Your system is set up to send the CYCLE START input again as soon as the program has finished a cycle.

The MODE SELECT switch must be in the AUTO position to perform SOP CYCLE START. Refer to Section 1.3.4 for more information on the MODE SELECT switch.

If you set the singularity stop system variable, $PARAM_GROUP[n].$AUTO_SNGSTP to FALSE, the robot will pass through singularity points while in AUTO mode. If you change the value of this variable, you must cycle power for the change to take effect.

Refer to Chapter 4 *INPUT/OUTPUT (I/O) SETUP* for more information about setting up to run production using SOP cycle start.

Use Procedure 7-13 to run production using the Standard Operator Panel (SOP) Cycle Start.

Procedure 7-13 Running Production Using Standard Operator Panel (SOP) Cycle Start

Conditions

- The robot is powered up and all faults have been corrected.
- All personnel and unnecessary equipment are out of the workcell.
- The program has been tested thoroughly and found to operate correctly.
- All safeguards have been installed and are functioning correctly.
- Any other conditions related to the application or robot have been satisfied.
- Test cycle conditions are set properly to allow robot motion, I/O, and full production speed.
- Single step testing is disabled and the STEP LED is not illuminated.

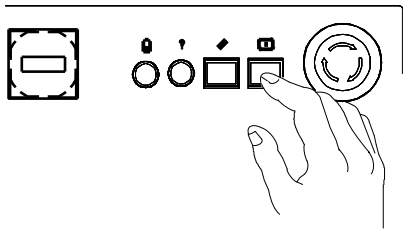
- The MODE SELECT switch is in the AUTO position.

> **Warning**
>
> **This procedure starts production. Make sure all safety barriers are in place, all personnel are outside of the workcell, all equipment is in place, and all production conditions have been met before you continue; otherwise, you could injure personnel and damage equipment.**

- The Remote/Local Setup item in the System Configuration Menu is set to LOCAL. Refer to Section 3.15 .

Steps

1. Select the program using the SELECT menu.
2. Press CYCLE START on the standard operator panel to start the application program.



7.5.3 User Operator Panel Start

A*User Operator Panel Start* is a method of automatically running the selected program. Selecting Start implies that you are not using RSR or PNS, but will use the START input on the User Operator Panel (UOP) to initiate production operation.

The START input will run the selected program for one cycle unless

- The program contains a loop, or
- Your system is set up to send the START input again as soon as the controller determines that the program has finished a cycle.

Refer to Chapter 4 *INPUT/OUTPUT (I/O) SETUP* for more information about setting up to run production using a UOP cycle start.

The MODE SELECT switch must be in the AUTO position to perform DIN CYCLE START. Refer to Section 1.3.4 for more information on the MODE SELECT switch.

If you set the singularity stop system variable, $PARAM_GROUP[n].$AUTO_SNGSTP to FALSE, the robot will pass through singularity points while in AUTO mode. If you change the value of this variable, you must cycle power for the change to take effect.

Use Procedure 7-14 to run production using User Operator Panel (UOP) Cycle Start.

Procedure 7-14 Running Production Using User Operator Panel (UOP) Start

Conditions

- The robot is powered up and all faults have been corrected.
- The program has been tested thoroughly and found to operate correctly.
- All personnel and unnecessary equipment are out of the workcell.
- All safeguards have been installed and are functioning correctly.
- Any other conditions related to the application or robot have been satisfied.
- UOP has been correctly installed and configured.
- The UOP UI enable signal *ENBL is ON.
- The UOP UI safety fence digital signal *SFSPD is ON.
- Test cycle conditions are set properly to allow robot motion, arc welding, I/O, and full production speed.
- Single step testing is disabled and the STEP LED is not illuminated.
- RSR and PNS are disabled.
- The MODE SELECT switch is in the AUTO position.



Warning

This procedure starts production. Make sure all safety barriers are in place, all personnel are outside of the workcell, all equipment is in place, and all production conditions have been met before you continue; otherwise, you could injure personnel and damage equipment.

- The Remote/Local Setup item in the System Configuration Menu is set to REMOTE. Refer to Section 3.15 .

Steps

1. Select the program using the SELECT menu.
2. Press the START or Production Start button on your UOP to start production.

7.5.4 Robot Service Request (RSR) Production Start

A*robot service request*(RSR) is a request for service from an external device. That request comes from a digital input signal on a preassigned RSR input line.

You can use up to eight robot service request signals: RSR1 through RSR8.

The MODE SELECT switch must be in the AUTO position to perform RSR Production Start. Refer to Section 1.3.4 for more information on the MODE SELECT switch.

If you set the singularity stop system variable, $PARAM_GROUP[n].$AUTO_SNGSTP, to FALSE, the robot will pass through singularity points while in AUTO mode.

Refer to Section 3.2.2 for more information about setting up to run production using RSR.

Procedure 7-15 Running Production Using Robot Service Requests (RSR)

Conditions

- The robot is powered up and all faults have been corrected and cleared.
- The program has been tested thoroughly and found to operate correctly.
- All personnel and unnecessary equipment are out of the workcell.
- All safeguards have been installed and are functioning correctly.
- Any other conditions related to the application or robot have been satisfied.
- UOP has been correctly installed and configured.
- The UOP UI enable signal *ENBL is ON.
- The UOP UI signal *IMSTP is ON.
- The UOP UI safety fence digital signal *SFSPD is ON.
- The UOP UI hold signal *HOLD is ON.
- Test cycle conditions are set properly to allow robot motion, arc welding, I/O, and full production speed.
- Single step testing is disabled and the STEP LED is not illuminated.
- RSR setup has been completed. Refer to Section 3.2.2 .
- PNS is disabled.
- All necessary cell I/O points, including the required input and output groups, have been defined and configured in the Cell I/O screen. (DispenseTool only)
- A cold start has been performed without any error messages appearing on the teach pendant error line (DispenseTool only)
- The MODE SELECT switch is in the AUTO position.

Warning

This procedure starts production. Make sure all safety barriers are in place, all personnel are outside of the workcell, all equipment is in place, and all production conditions have been met before you continue; otherwise, you could injure personnel and damage equipment.

Steps

1. Set the LOCAL/REMOTE item on the System Config screen to REMOTE. When the RSR input is received, production operation begins as long as all UOP UI conditions are satisfied.

   Note If the REMOTE LED is not illuminated, check each Condition and reset if necessary (applies to DispenseTool only).

2. Have the cell controller send RSR signals by raising RSR input signals.
3. If all conditions are satisfied when the RSR signals are received, the job set up for that RSR signal will be executed immediately.

7.5.5 Program Number Select (PNS) and UOP Production Start

A*program number select (PNS)* is a method of selecting a program to be run by some external device. The name of the program to be run is received by the controller as a group of input signals from an external device on a total of eight PNS input lines.

The MODE SELECT switch must be in the AUTO position to perform PNS Production Start. Refer to Section 1.3.4 for more information on the MODE SELECT switch.

If you set the singularity stop system variable, $PARAM_GROUP[n].$AUTO_SNGSTP, to FALSE, the robot will pass through singularity points while in AUTO mode.

Refer to Section 3.2.3 for more information about setting up to run production using PNS.

Procedure 7-16 Running Production Using Program Number Select (PNS) and UOP Production Start

Conditions

- The robot is turned on and all faults have been corrected.
- The program has been tested thoroughly and found to operate correctly.
- All personnel and unnecessary equipment are out of the workcell.
- All safeguards have been installed and are functioning correctly.
- Any other conditions related to the application or robot have been satisfied.

- UOP has been correctly installed and configured.
- The UOP UI enable signal *ENBL is ON.
- The UOP UI safety fence digital signal *SFSPD is ON.
- Test cycle conditions are set properly to allow robot motion, arc welding, I/O, and full production speed.
- Single step testing is disabled and the STEP LED is not illuminated.
- PNS setup has been completed.
- RSR is disabled.
- The MODE SELECT switch is in the AUTO position.



Warning

This procedure starts production. Make sure all safety barriers are in place, all personnel are outside of the workcell, all equipment is in place, and all production conditions have been met before you continue; otherwise, you could injure personnel and damage equipment.



Warning

If you do not follow this procedure, temporary memory in the controller will be filled. This causes the process CPU to be locked into a busy and running condition. Make sure your PLC logic is correct and does not contain a high rate of production start calls; otherwise, you could injure personnel or damage equipment.

- The Remote/Local Setup item in the System Configuration Menu is set to REMOTE. Refer to Section 3.15 .

Steps

1. Set the 8 bit PNS input to the number that when added to the base number will determine which program is selected. Refer to Chapter 3 *GENERAL SETUP* , for more information about PNS.
2. Strobe the PNSTROBE input. When the controller receives the input signal, the selected program will be displayed on the teach pendant screen. The ACK UOP signal indicates what binary input is being received. This stays ON until a new program is selected.
3. Press the appropriate production start button on the user operator panel to start production operation or, if your system uses a PLC, production operations will begin as soon as the PROD_START input is received.

7.6 ADJUSTING A PROGRAM (PROG ADJUST)

7.6.1 Overview

Program adjust allows you to modify positional data of a program relative to different frames and handling rail axes, multiple groups, and line tracking. You can adjust positional offsets (values that specify how much of a difference there is between the current positional value and the positional value you want), robot speed, and rail axes. This can also be used to edit a program while another program is running.

Program adjust allows you to:

- Adjust positional offsets
- Adjust robot speed - linear and joint
- Support adjustment reference to both USER frame and TOOL frame

Note Program adjust cannot correct for errors in mastering.

7.6.2 Adjusting a Program or a Schedule

When you make program adjustments, the changes you make are grouped together into a *program adjust schedule*. The default number of program adjust schedules is 10, but you can use up to 99 program adjust schedules to adjust program information during program or production run. You can change the number of schedules in PROGRAM SETUP at Controlled Start. Refer to the *Software Installation Manuall*.

Program Adjust Schedules

The program adjust schedule contains:

- A **number** you assign to **identify the schedule** .
- The **name of the program** being adjusted.
- The **starting and ending line numbers** to be affected by the adjustment.
- The **frame adjustment reference**: either USER or TOOL.
- The **positional offset value** in x, y, z, w, p, and r in USER frame adjustment, or x, y, z, x rotation, y rotation, and z rotation in TOOL frame adjustment.
- A **linear robot speed**.
- A **circular robot speed**.
- A **joint robot speed**.

- The **motion group** selection.
- A **rail mode adjustment** if you have rail axis as an integrated linear axis in the first group.

Program Adjust Guidelines

Use the following program adjust sequence as a guideline for your program adjustments:

1. Edit the schedule to make the program adjustments you want. Use Procedure 7-17 .
2. ENABLE the program adjust schedule. The changes will take effect as soon as the robot motion system can process the new information. Allow for one complete cycle through the program after enabling the adjustment to ensure that all positions are adjusted.
3. **To remove the adjustment**, DISABLE the offset. Allow for one complete cycle of the program for all positions to disable the adjustments.
4. **To make additional adjustments,** clear the schedule (CLEAR_ADJ) and confirm. This will lock in the adjustments to the program and reset the program adjustment values to zero.
5. **If the offset is a temporary adjustment,** continue to use the offset until the temporary condition no longer exists and then DISABLE the offset.

You can display program adjust schedule information from the UTILITIES Prog Adj screen. Refer to Table 7–8 and Table 7–9 for more information.

Use Procedure 7-17 to adjust program information during program or production run.

Table 7–8. UTILITIES Prog Adj Screen Items

ITEM	DESCRIPTION
Schedule #	This item is the number that identifies the schedule.
Program	This item is the name of the program you are adjusting.
Lines	This item is the starting and ending line numbers that are affected by the adjustment.
Status	This item is the status of the item currently selected.

Table 7–9. UTILITIES Prog Adj DETAIL Screen Items

ITEM	DESCRIPTION
Current schedule	This item is the number that identifies the schedule.
Status	This item is the status of the item currently selected.
Program name	This item is the name of the program you are adjusting.
Starting line number	This item identifies the starting line number that will be affected by the adjustment.

Table 7–9. UTILITIES Prog Adj DETAIL Screen Items (Cont'd)

Ending line number	This item identifies the ending line number that will be affected by the adjustment.
Offset relative to	This item indicates the positional offset is relative to either USER or TOOL frame.
X, Y, Z, W, P, R adjustment	These items are the positional offset values.
Linear/Circular speed	This item is the linear and circular robot speed.
Joint speed	This item is the joint robot speed.
Motion group	This item indicates the motion group for multiple group processes.

Procedure 7-17 Adjusting Programs During Program or Production Run

Conditions

- The program you want to adjust is currently selected.

Steps

1. Press MENUS.
2. Select UTILITIES.
3. Press F1, [TYPE].
4. Select Prog Adjust. Depending on the application you are using, you will see a screen similar to the following.

```
UTILITIES Prog Adj
 Program Adjust Schedules
 # Program       Lines        Status
 1 PRG123        1-23         EDIT
 2 PRG123        39-49        ENABLED
 3 PRG345        10-14        DISABLED
 4 PRG678        12-22       DISABLED
 5 ********      1 - 0       ********
 6 ********      1 - 0       ********
 7 ********      1 - 0       ********
 8 ********      1 - 0       ********
 9 ********      1 - 0       ********
10 ********      1 - 0       ********
```

5. Select a program and line numbers to adjust.

To adjust program parameters for the current program if it is not listed on the screen, select an unused schedule (********) and press F2, DETAIL. The current program name will be entered automatically.

6. Press F2, DETAIL. You will see a screen similar to the following.

```
UTILITIES Prog Adj
Current Schedule: 1    Status:  EDIT
  1 Program name:              PRG123
  2 Starting line number:      1
  3 Ending Line number:        23
  4 Offset relative to:        USER
  5 X adjustment:              0.000 mm
  6 Y adjustment:              0.000 mm
  7 Z adjustment:              0.000 mm
  8 W adjustment:              0.000 deg
  9 P adjustment:              0.000 deg
  10 R adjustment              0.000 deg
  11 Linear/Circular Speed:    0 mm/sec
  12 Joint Speed:              0%
  13 Motion Group:             Single Group
  14 Adjust Y for:             Robot
```

7. **To display the DETAIL for a different program**,
   a. Move the cursor to Program name.
   b. Press F4, [CHOICE].
   c. Move the cursor to the name of the program you want and press ENTER.
8. **To display the DETAIL for a different schedule**, press F3, SCHED, and type the schedule number.
9. **To display the next schedule DETAIL automatically**, press SHIFT and F3, SCHED.
10. Select item 2, Starting line number, and type the starting line number where you want the changes to take effect. If you enter a line number that is out of the range of program lines, the last line number in the program will be entered automatically.
11. Select item 3, Ending line number, and type the ending line number where you want the changes to stop taking effect. If you enter a line number that is out of the range of program lines, the last line number in the program will be entered automatically.

Note The ending line must be greater than or equal to the starting line number specified in item 2. **To change only one line number,** the ending line number must be the same as the starting line number.

12. **To select the units** (inches or millimeters) for x, y, and z offsets, press F2, UNITS.
13. **To toggle between USER frame and TOOL frame**, move the cursor to item 4, Offset relative to, and press either F4, USER, or F5, TOOL.
14. **To adjust x, y, and z offsets,** select the item and type the new offset value. To indicate negative offsets, use the minus sign. The range of x, y, and z offsets is +/- 26.00 mm.
15. **To adjust w, p, and r offsets,** select the item and type the new offset value. These offsets are always shown in degrees. To indicate negative offsets, use the minus sign. The range of w, p, and r offsets is +/- .500 dg.

    Note If **Offset relative to** is set to TOOL frame, instead of adjusting w, p, and r values, you will adjust the "x rotation," "y rotation," and "z rotation."

16. **To change linear or circular speed**, select Linear/Circular Speed and type the new speed value. A value of 0 indicates no change.
17. **To change joint speed**, select Joint Speed and type the new speed value. A value of 0 indicates no change.
18. **To select the motion group for multiple group processes**, select Motion group and press F4, [CHOICE]. You can select Single Group, All, or an individual group that is loaded on the controller.

    - Single Group indicates that only the first is being adjusted in this process. If only the first group is loaded, then Single Group is the only choice on the pop-up menu.
    - All indicates that all motion groups in the teach pendant program are being adjusted in this process.
    - If more than one group is loaded on the controller, the value for $SCR_GRP.$robot_id is displayed as a selection for individual groups from a pop-up menu. This selection allows you to adjust any individual group.

19. **To adjust y for the robot rail or all,** select the item and press F2, [CHOICE], to make a selection.

    Note Adjust Y for is displayed only if rail axis is loaded as an integrated linear axis in the first motion group.

    Note If Frame Adj refer to is set to TOOL frame, only ROBOT is available.

Caution

The next step describes how to change position values in the program permanently. Be sure you want to change program values permanently before you perform this step; otherwise, unexpected results could occur.

Caution

If you use CLR_ADJ on an Enabled Schedule the current adjustment values default to zero and the schedule status is set to EDIT. The program adjustments become permanent and the original values are removed.

20. **If you want to clear the x, y, z, w,p, and r portion of this schedule**, press NEXT, >, and then press F2, CLR_ADJ. This

    - Changes the x, y, z, w/x rotation, p/y rotation, and r/z rotation offset values in the schedule to 0
    - Retains the program name and line numbers
    - Resets the position to include the adjusted positional information
    - **To clear**, press F4, YES. You will see the message "Schedule cleared successfully."
    - **To cancel**, press F5, NO.

21. **When you are finished adjusting program parameters and the program is not currently running**, press F4, ENABLE.

    This activates the program adjustments you made and changes the program. The adjustments take effect and become permanent to the program as soon as the robot motion system processes them.

Caution

Do not modify schedule values while the schedule is enabled. Unexpected motion could occur when the schedule is either re-enabled or disabled, and the schedule will not return to the original values.

Note If a motion instruction contains a PR[n] (position register) or INC (Incremental motion option), it will not be adjusted.

22. **To test the adjustments if the program is not running**, refer to the " Test Cycle" section.

23. **If you are not satisfied with the adjustments**, press F5, DISABLE. (F5, DISABLE, appears after you have enabled adjustments.) This returns the program positions to the values they had before you enabled the schedule. The disable feature takes effect as soon as the robot motion system can process it. The changes are permanent.

If you are not satisfied with speed changes, you must EDIT the schedule to enter new speed values and then press F4, ENABLE.

> **Caution**
>
> CLR_ALL makes the current program adjustment changes permanent and removes the original values. Do not use CLR_ALL unless you want the changes to be permanent.

Note DISABLE does not change linear speed or joint speed. You must change them to the original values by selecting each value and entering a new value.

Before you can run the program the new program adjustments must be complete.

24. **To save the changes to your program permanently**,

a. Press F4, ENABLE, to enable the changes.

b. Press NEXT,>.

c. Press CLR_ALL.

- This activates the program adjustments you made and makes permanent changes to the program.
- This changes the x, y, z, w/x rotation, p/y rotation, and r/z rotation offset values displayed on the screen to 0.
- This changes the linear speed and joint speed values displayed on the screen to 0.
- This clears the program name and line numbers displayed on the screen.

- To **clear**, press F4, YES.
- To **cancel**, press F5, NO.

25. **To copy schedule information from one program to another,** press NEXT, >, and then press F1, COPY. Type the schedule number you want to copy to a specified schedule and press ENTER.

26. The message "Please wait, copying schedule..." is displayed for the moment. You will see the message "Schedule copied successfully."

27. If the adjustment fails, the specific cause of error is posted and the line number where the error occurred is displayed. The schedule status is changed to PARTENABLED (or partial enabled). See the following screen for an example.

```
UTILITIES Prog Adj
Current Schedule: 2    Status: PARTENABLED
  1 Program name:               PRG123
  2 Starting line number:       1
  3 Ending Line number:         29
  4 Offset relative to:         USER
  5 X adjustment:               5.000 mm
  6 Y adjustment:               0.000 mm
  7 Z adjustment:               -2.500 mm
  8 W adjustment:               0.000 deg
  9 P adjustment:               0.000 deg
  10 R adjustment               0.000 deg
 Adjustment failed at line  2
```

a. Press F5, DISABLE, to disable the schedule.

b. Determine the cause of the error.

- Fix the error in the program.

OR

- Change the ending line number to be one less than where the error occurred.

c. Press F4, ENABLE, to enable the schedule.



Caution

Do not use a partially adjusted program because it could cause unexpected motion.

Note When a schedule status is either PARTENABLED or PARTDISBLED, you will not be able to use CLR_ADJ or CLR_ALL to make partial adjustments permanent. Also, if you change any adjusted fields, the status changes to EDIT and you will be unable to recover positions lost due to power failure.

Procedure 7-18 Modifying the Number of Program Adjust Schedules

Conditions

- You have just performed a Controlled start. Refer to Appendix C for more information.

Steps

1. Press MENUS.
2. Press NEXT.
3. Select Program Setup. You will see a screen similar to the following.

```
Program Limits          CONTROLLED START MENUS
 Program Limits Setup
  1 User Tasks                    PRG123
  2 Numeric Registers             200
  3 Position Registers            100
  4 Macros                        120
  5 User Alarms                   10
  6 Trace Length                  200
  7 Num. Dig. Ports               512
  8 Error Severity Table          20
  9 Program Adjust Schedules      10
```

4. Move the cursor to Program Adjust Schedules and type the desired number of Program Adjust schedules. You can enter a value between 1 and 99.
5. Press FCTN.
6. Select START (COLD).

7.6.3 Error Handling

The program adjust utility has the ability to detect and recover from the following kinds of errors:

- Limit and position not reachable
- Power failure and data recovery

In addition, the program adjust utility will protect positional data of your programs.

Motion Error Prechecks

Limit errors and position not reachable errors are pre-checked before any adjustments are made to the program. After an error condition is detected the program adjust utility will attempt an adjustment by performing a motion error precheck. If the intended adjustment will not cause a motion error, the adjustment will be made. If the intended adjustment will cause a motion error, the adjustment will be aborted after the pre-check.

The detected error type and the line number at which the error might occur will be displayed. Alarm messages will also be written to the alarm log and can be posted to the PLC.

Power Failure and Data Recovery

In case of power failure, only some of the intended adjustment positions are updated. This results in a condition known as PARTENABLED. PARTENABLED (or partial enabled) means that an adjustment was in progress during power failure, and is a state from which you can recover.

For example, if you are in the middle of adjusting lines 1 to 100 when the power fails, and only lines 1 to 60 were updated with their new positional data, the robot would be in a PARTENABLED state.

To recover from this condition and restore the original positional data back to lines 1 to 60, press F4, DISABLE, when you turn on the controller. You will see a screen similar to the following.

```
UTILITIES Prog Adj
Current Schedule: 1    Status:  PARTENABLED
  1 Program name:               PRG123
  2 Starting line number:       1
  3 Ending Line number:         4
  4 Offset relative to:         USER
  5 X adjustment:               3.000 mm
  6 Y adjustment:               25.000 mm
  7 Z adjustment:               10.500 mm
  8 W adjustment:               0.500 deg
  9 P adjustment:               0.000 deg
  10 R adjustment               0.000 deg
  11 Linear/Circular Speed:     0 mm/sec
  12 Joint Speed:               0%
  13 Motion Group:              Single Group
  14 Adjust Y for:              Robot
```

If power fails a second time during the process of recovering original data, and only lines 1 to 20 were recovered, the status will change to PARTDISBLED (partial disabled) when you turn on the controller again. You will see a screen similar to the following.

```
UTILITIES Prog Adj
Current Schedule: 1    Status:  PARTDISBLED
  1 Program name:               PRG123
  2 Starting line number:       1
  3 Ending Line number:         4
  4 Offset relative to:         USER
  5 X adjustment:               3.000 mm
  6 Y adjustment:               25.000 mm
  7 Z adjustment:               10.500 mm
  8 W adjustment:               0.500 deg
  9 P adjustment:               0.000 deg
  10 R adjustment               0.000 deg
  11 Linear/Circular Speed:     0 mm/sec
  12 Joint Speed:               0%
  13 Motion Group:              Single Group
  14 Adjust Y for:              Robot
```

In this case, press F4, DISABLE, again to recover original positional data in lines 21 to 60.

Note If you encounter a limit error or position unreachable error during the recovery process, the message "enable failed" will be displayed and the status will change to EDIT.

7.7 MAINTENANCE AND REPAIR

You can use *macro commands* to appear on the MANUAL FCTNS to perform maintenance and repair procedures. Refer to Section 3.3 for more information about how to set up macro commands.

If you are using DispenseTool, you can also use the MANUAL FCTNS screen to execute the predefined programs that direct the robot to move to the home positions, move to the purge position, and move to the service position. You must define these programs before you can execute them from the MANUAL FCTNS screen.

Use Procedure 7-19 to perform a manual function.

Procedure 7-19 Using the MANUAL FCTNS Screen

Conditions

- Macro commands that perform maintenance and repair functions have been set up to appear on the MANUAL FCTNS screen. Refer to Section 5.2 .
- If you are using DispenseTool, the programs MOV_HOME, MOV_PURG, and MOV_SERV have been defined.

Steps

1. Press MENUS.
2. Select MANUAL FCTNS.
3. Press F1, [TYPE].
4. Select Macros. You will see a screen similar to the following.

```
Manual Macros
    Instruction
  1  OPEN HAND 1
  2  CLOSE HAND 1
  3  RELAX HAND 1
  4  OPEN HAND 2
  5  CLOSE HAND 2
  6  RELAX HAND 2
  7  GO TO REPAIR POS
```

5. Move the cursor to the instruction you want to execute.
6. Continuously press and hold in the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.



Warning

In the next step, the robot will move. Make sure that personnel and unnecessary equipment are out of the workcell; otherwise, you could injure personnel or damage equipment.

7. Press and hold the SHIFT key and press F3, EXEC. The F3 key can be released, but the SHIFT key must be held continuously until the instruction has finished executing.

Chapter 8

STATUS DISPLAYS AND INDICATORS

Contents

Chapter 8	STATUS DISPLAYS AND INDICATORS	8–1
8.1	STATUS INDICATORS	8–3
8.1.1	Overview	8–3
8.1.2	Teach Pendant Status Indicators	8–3
8.1.3	Standard Operator Panel Status Indicators	8–5
8.2	USER SCREEN STATUS	8–7
8.3	REGISTER STATUS	8–8
8.4	POSITION REGISTER STATUS	8–10
8.5	STRING REGISTER STATUS	8–14
8.6	PALLET REGISTER STATUS	8–17
8.7	SYSTEM VARIABLE STATUS	8–18
8.8	SAFETY SIGNAL STATUS	8–20
8.9	PROGRAM TIMER STATUS	8–22
8.10	SYSTEM TIMER STATUS	8–25
8.11	CLOCK STATUS	8–26
8.12	VERSION IDENTIFICATION STATUS	8–27
8.13	MEMORY STATUS	8–31
8.13.1	Checking Memory	8–31
8.14	POSITION STATUS	8–33
8.15	TURN NUMBER DISPLAY	8–36
8.15.1	Overview	8–36
8.15.2	Usual Configuration	8–36
8.15.3	$SCR_GRP[group].$turn_axis[i] System Variable	8–39
8.16	EXECUTION HISTORY	8–40
8.17	PROCESS AXES STATUS (OPTION)	8–42

8.18 PROGRAM or PRODUCTION STATUS 8–43
8.19 ROBOT AXES STATUS 8–45
8.19.1 Overview 8–45
8.19.2 Status 1 Screen 8–45
8.19.3 Status 2 Screen 8–49
8.19.4 Pulse Screen 8–50
8.19.5 Torque Monitor Screen 8–51
8.19.6 Tracking Screen 8–52
8.19.7 Disturbance Torque Screen 8–53
8.19.8 Servo Diagnosis 8–53
8.19.9 Procedure 8–60
8.20 PROGRAM MONITOR STATUS 8–61
8.21 VISUAL DIAGNOSTICS 8–62
8.21.1 Overview 8–62
8.21.2 Creating and Editing Visual Diagnostic Screens 8–64
8.21.3 Managing Screens 8–78
8.21.4 Viewing Visual Diagnostic Screens 8–81

8.1 STATUS INDICATORS

8.1.1 Overview

Teach pendant and standard operator panel status indicators show various conditions of the system. Your system can also have other indicators on user operator panels. See your supervisor for information about user operator panel indicators.

8.1.2 Teach Pendant Status Indicators

Teach pendant status indicators indicate the system condition when you are using the teach pendant to control the system.

Figure 8–1 and Figure 8–2 show the teach pendant status indicators. You might have different indicators depending on the application that you are using. Table 8–1 lists and describes each teach pendant status indicator.

Figure 8–1. Teach Pendant Status Indicators

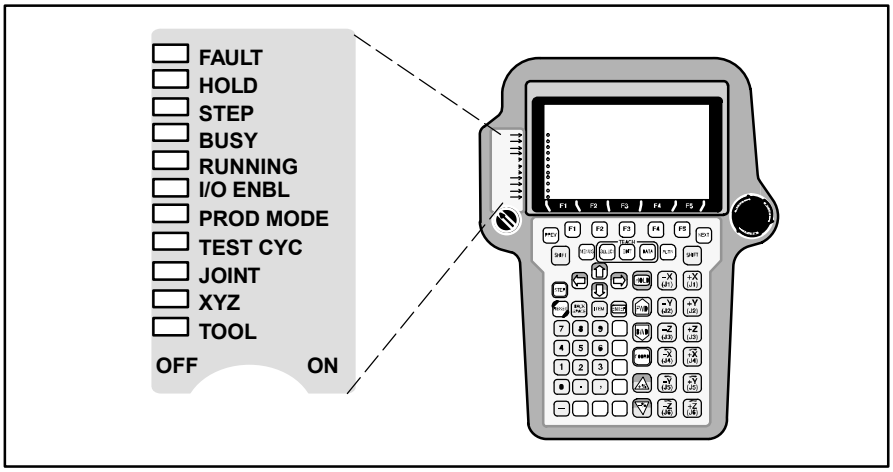


Figure 8–2 shows the status indicators on the iPendant.

Figure 8–2. iPendant Status Indicators

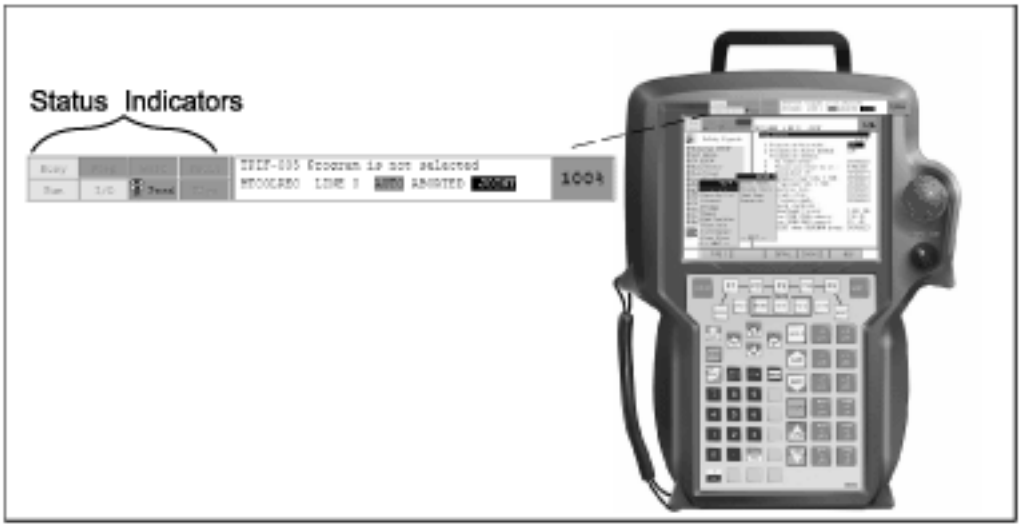


Table 8–1. Teach Pendant Status Indicators

INDICATOR	DESCRIPTION
FAULT	This item indicates that a fault condition has occurred.
HOLD	This item indicates that the robot is in a hold condition. HOLD is not on continuously during a hold condition.
STEP	This item indicates that the robot is in step mode.
BUSY	This item indicates that the controller is processing information.
RUNNING	This item indicates that a program is being executed.
I/O ENBL	This item indicates that I/O is enabled.
PROD MODE	This item indicates that the system is in production mode and CYCLE START will start the process. For PalletTool, this item indicates that the system is running in production mode with the Remote/Local item on the System Config screen set to REMOTE.
TEST CYC	This item indicates that the system is test running a teach pendant or KAREL program with the Remote/Local item on the System Config screen set to LOCAL.
JOINT	This item indicates that the current jog coordinate system is JOINT.
XYZ	This item indicates that the current jog coordinate system is JOG frame, USER frame, or WORLD.
TOOL	This item indicates that the current jog coordinate system is TOOL.

Table 8–1. Teach Pendant Status Indicators (Cont'd)

INDICATOR	DESCRIPTION
GUN ENBL for SpotTool+	This item indicates that the gun will stroke when spot welding commands are executed.
WELD ENBL for Spotool+	This item indicates that welding is enabled, and that spot welds will be performed when SPOT[] commands are executed.
SEAL ENBL for DispenseTool	This indicates that the system is ready for production and wet run is set to ENABLE. **SEAL ENBL** indicates that material can and will be dispensed.
TEST CYCLE for DispenseTool	This item indicates that the system is in test cycle mode.
MAN ENBL for PaintTool	This item indicates that you can perform manual functions.
WELD ENBL for ArcTool	This item indicates that arc welding is enabled. If the arc welding program is not running by remote, this LED is controlled by the teach pendant key **WELD ENBL** . If the arc weld program is running by remote, this LED can be controlled by a digital input.
ARC ESTAB	This item indicates that the robot is welding.
DRY RUN	This item indicates that the program will execute without welding. If the test cycle dry run conditions has been set to ON, then the **DRY RUN** LED will be ON. The robot motion speed is controlled by the amount specified in the test cycle screen.
I/O ENBL for SpotTool+	This item indicates that I/O is enabled. I/O cannot be disabled.

8.1.3 Standard Operator Panel Status Indicators

Standard operator panel (SOP) status indicators indicate the system condition when you are using the operator panel to control the system.

Figure 8–3 shows the standard operator panel status indicators. Table 8–2 lists and describes each operator panel status indicator.

Figure 8–3. Standard Operator Panel

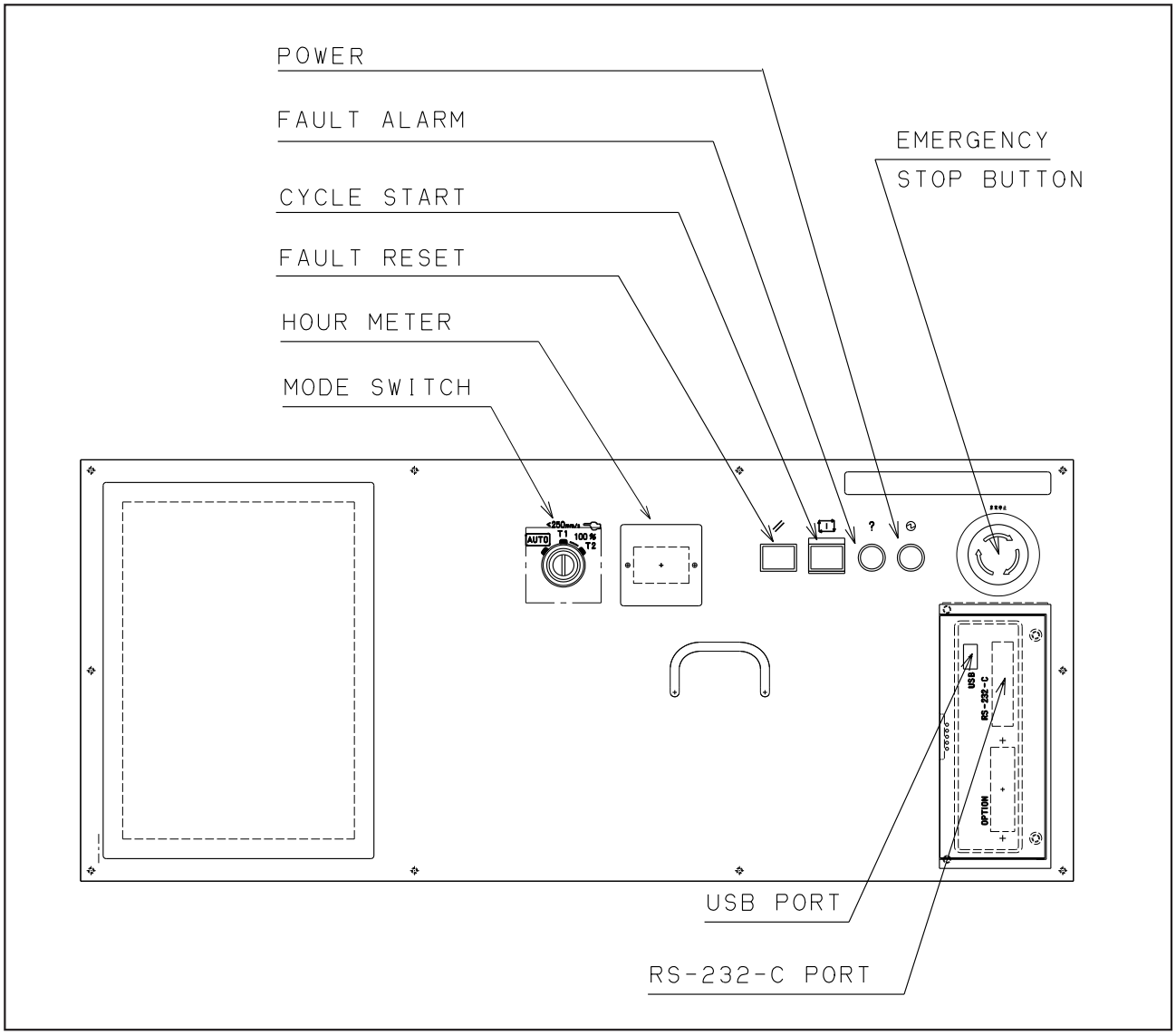


Table 8–2. Standard Operator Panel Status Indicators

INDICATOR	DESCRIPTION
FAULT ALARM	This item indicates a fault condition has occurred.
POWER	This item indicates controller power is on.
CYCLE START	This LED is lit when the currently selected program is being started.

8.2 USER SCREEN STATUS

The user screen displays messages sent to the user from a running program. You cannot use this screen to change information.

User messages are controlled using the MESSAGE instruction in your program. Each time the MESSAGE instruction is used, one line containing up to and including 23 characters, is written to the user screen.

A maximum of nine message lines can be displayed. If more than nine message lines are used, the tenth line is added to the bottom of the screen and the top line scrolls off.

For KAREL programs, user messages are controlled by using the write statement in your program.

Refer to the *KAREL Reference Manual* for information on the write statement. Use Procedure 8-1 to display the user screen.

Procedure 8-1 Displaying the User Screen

Steps

1. Press MENUS.
2. Select USER. You will see a screen similar to the following.

```
USER
THE_SYSTEM_HAS_POWERED
UP_SUCCESSFULLY
MOVE_THE_ROBOT_TO_HOME
POSITION_BEFORE_RUNNING
PRODUCTION
THE_ROBOT_IS_AT_THE
HOME_POSITION
```

Note This screen is blank if no messages were sent to it from a running program. The screen saves messages even after the program has aborted.

USER2 Menu

The USER2 menu is available in all application tools. When the KAREL option is loaded you get a menu choice of USER2 when the MENUS key is pressed. The USER2 menu displays forms, or messages that are initiated from KAREL programs.

Refer to the *KAREL Reference Manual* for more information. Use Procedure 8-2 to display the USER2 menu.

Procedure 8-2 Displaying the User2 Menu

Steps

1. Press MENUS.
2. Select USER2. See the following screen for an example.

Note This screen is blank if no forms were written.

```
USER
THE_SYSTEM_HAS_POWERED
UP_SUCCESSFULLY
MOVE_THE_ROBOT_TO_HOME
POSITION_BEFORE_RUNNING
PRODUCTION
THE_ROBOT_IS_AT_THE
HOME_POSITION
```

8.3 REGISTER STATUS

The DATA Registers screen displays the current value of each register in the system. You can change the value of any register and add comments using the register screen.

Table 8–3. DATA Registers Screen Items

ITEM	DESCRIPTION
R #	This item is the register number.
Comment field	This item is a text field into which you can type a comment for the selected register. To type a comment, move the cursor to the appropriate register, press ENTER, and type your comment.
Value	This item is the value of the selected register. To change the value, move the cursor to the appropriate register, type the new value, and press ENTER.

Refer to Section 6.5 for information on registers. Use Procedure 8-3 to display the register screen.

Procedure 8-3 Displaying and Setting Registers

Steps

1. Press DATA.
2. Press F1, [TYPE].

3. Select Registers. You will see a screen similar to the following.

```
Data Registers
R[  1:                ]     =0
R[  2:                ]     =0
R[  3:                ]     =0
R[  4:                ]     =0
R[  5:                ]     =0
R[  6:                ]     =0
R[  7:                ]     =0
R[  8:                ]     =0
R[  9:                ]     =0
R[ 10:                ]     =0
Press ENTER
```



Caution

Registers are used in programs. Do not modify register values unless you are sure how the register is used in the system; otherwise, you could affect how programs are executed.

4. To add a comment

a. Move the cursor to the register number and press ENTER.

b. Move the cursor to select a method of naming the comment: Upper Case, Lower Case, Punctuation, Options.

c. Press the function keys whose labels correspond to the name you want to give to the comment. These labels vary depending on the naming method you chose in Step 4b .

For example, if you chose Upper Case, press a function key corresponding to the first letter. Press that key until the letter you want is displayed in the comment field. Press the right arrow key to move the cursor to the next space. Continue until the entire comment is displayed. **To delete a character**, press BACK SPACE.

d. When you are finished, press ENTER.

Note The comment appears in the teach pendant programs that are using that register.

5. To change the value of the register

a. Move the cursor to the register value.

b. Type the new value and press ENTER.

6. To save the register values to the default device

a. Press MENUS.
b. Select FILE.
c. Press F1, [TYPE].
d. Select File.
e. Press F5, [UTIL].
f. Select Set Device.
g. Move the cursor to the device you want and press ENTER.
h. Press DATA.
i. Press F1, [TYPE].
j. Select Registers.
k. Press FCTN.
l. Select SAVE. The registers will be saved to the file, NUMREG.VR, on the default device.

8.4 POSITION REGISTER STATUS

The DATA Position Reg screen displays the current value of each position register in the system. You can change the value of any position register and add comments using the DATA Position Reg screen.

For information on position registers, refer to Section 6.6 .

Table 8–4. DATA Position Reg Screen Items

ITEM	DESCRIPTION
PR #	This item is the position register number.
Comment field	This item is a text field into which you can type a comment for the selected register. To type a comment, move the cursor to the appropriate register, press ENTER, and type your comment.
Value	This item is the value of the selected register. To change the value, move the cursor to the appropriate register, type the new value, and press ENTER.

Note If you are using SpotTool+, you must define PR[1] as the home position. SpotTool+ reserves PR[1] as the home position.

If your system is configured to have more than one motion group, you can set the *group mask* when you use any position register instruction. The group mask allows you to use function keys to specify:

- Whether the group mask will be used. If the group mask is not used, the position register instruction affects the default group only.
- The group or groups that the position register instruction will affect.

Use Procedure 8-4 to display the position register screen.

Procedure 8-4 Displaying and Setting Position Registers

Steps

1. Press DATA.
2. Press F1, [TYPE].
3. Select Position Reg. You will see a screen similar to the following.

Note If you are using application tool packages that use the Common Shell, PR[1] is used as the home position. You cannot use PR[1] for any other position in your programs.

```
DATA Position Reg
PR[  1:Home             ]     =R
PR[  2:                 ]     =*
PR[  3:                 ]     =*
PR[  4:                 ]     =*
PR[  5:                 ]     =*
PR[  6:                 ]     =*
PR[  7:                 ]     =*
PR[  8:                 ]     =*
PR[  9:                 ]     =*
PR[ 10:                 ]     =*
Press ENTER
```

R indicates the position has been recorded.

* indicates the position has not been recorded.



Caution

Position registers are used in programs. Do not modify position register values unless you are sure how the position register is used in the system; otherwise, you could affect how programs are executed.

4. **To add a comment**
   a. Move the cursor to the position register number and press ENTER.
   b. Move the cursor to select a method of naming the comment.
   c. Press the function keys whose labels correspond to the name you want to give to the comment. These labels vary depending on the naming method you chose in Step 4b .

   For example, if you chose Upper Case, press a function key corresponding to the first letter. Press that key until the letter you want is displayed in the comment field. Press the

right arrow key to move the cursor to the next space. Continue until the entire comment is displayed.

To delete a character, press BACK SPACE.

d. When you are finished, press ENTER.

Note The comment appears in all teach pendant programs using that position register.

5. To change the value of the position register

a. Move the cursor to the position register value.

b. Enter the new value by **recording a position** (Step 6), or **entering positional information** (Step 8).

6. To record a position,

a. Press and hold the DEADMAN switch and turn on the teach pendant.

b. Jog the robot to the position you want.

c. Hold down the SHIFT key and press F3, RECORD. The * (asterisk) will change to an R to indicate the position has been recorded.

The user frame, UF, and the tool frame, UT, will be set to 15 (FHex), which indicates that the currently active user frame and tool frame will be used. Refer to Section 3.1 for information on setting up frames.

Note If you want to change the user frame or tool frame value after you have recorded the position register, you must select the new user frame or tool frame, record the position register, and then manually enter the desired coordinates in order for the changed frame to take effect. Refer to Step 8 for information on entering position register information manually.

7. To move to a recorded position register,

a. Press and hold the DEADMAN switch and turn on the teach pendant.

b. Press and hold the SHIFT key and press F2, MOVE_TO.

Note Recording position registers in a multiple motion group system records position values for **ALL** axes, regardless of the default group mask.

8. To enter positional information manually,

a. Press F4, POSITION. You will see a screen similar to the following.

```
Position Detail
PR[2]   UF:F  UT:F          CONF: NDF 000
 X  0.500        mm     W  0.00       deg
 Y  1.320        mm     P 90.00       deg
 Z  0.750        mm     R  0.00       deg
DATA Position Reg
PR[  1:                ] =R
PR[  2:                ] =*
PR[  3:                ] =*
PR[  4:                ] =*
PR[  5:                ] =*
PR[  6:                ] =*
Enter value
```

R indicates the position has been recorded.

* indicates the position has not been recorded.

Note **If you are using Application tools that use the common shell**, PR[1] is used as the home position. You do not use it for any other position in your programs.

b. **To change the format of the position from Cartesian coordinates to joint angles or from joint angles to Cartesian coordinates**, press F5, [REPRE] and select the coordinate system. The proper joint angles or Cartesian coordinates will be displayed. The position is automatically converted.

Note Joint angles are useful to zero-position the robot or to control the motion of a positioning table.

c. **To change a position component**, move the cursor to the component, type the value, and press ENTER.

d. **If you have multiple motion groups, and you want to change the motion group number**, press F1, GROUP, type the group number, and press ENTER. This only applies to systems that have been set up for multiple groups.

e. **To change the configuration**, press F3, CONFIG. Select the proper configuration by pressing the up or down arrow key.

f. **If you have the extended axis option, and you want to display the extended axis position information,** press F2, PAGE. This only applies to systems that include extended axes.

g. If you need to change a position component in another group, press F1, GROUP, key.

h. When you are finished, press F4, DONE.

9. To clear a position register press F5, CLEAR. This converts all positional information to all asterisks (*******).

10. To save the position register values to a file:

a. Press MENUS.

b. Select FILE.

c. Press F1, [TYPE].

d. Select File.

e. Press F5, [UTIL].

f. Select Set Device.

g. Move the cursor to the device you want and press ENTER.

h. Press DATA.

i. Press F1, [TYPE].

j. Select Position Reg.

k. Press FCTN.

l. Select SAVE. The position registers will be saved to the file, POSREG.VR, on the default device.

8.5 STRING REGISTER STATUS

The DATA String Reg screen displays the current value of each string register in the system. You can change the value of any string register and add comments using the DATA String Reg Screen.

Table 8–5. Data String Reg Screen Items

ITEM	DESCRIPTION
SR#	This item is the number of the string register.
Comment field	This item is a text field into which you can type a comment for the selected register. To type a comment, move the cursor to the appropriate register, press ENTER, and type your comment.
Value	This item is the value of the selected register. To change the value, move the cursor to the appropriate register, type the new value, and press ENTER.

Procedure 8-5 Displaying and Setting String Registers

Steps

1. Press DATA.

2. Press F1, [TYPE].
3. Select String Reg. You will see a screen similar to the following.

```
DATA String Regist            JOINT 100 %
SR[ 1:                 ]=
SR[ 2:                 ]=
SR[ 3:                 ]=
SR[ 4:                 ]=
SR[ 5:                 ]=
SR[ 6:                 ]=
SR[ 7:                 ]=
SR[ 8:                 ]=
SR[ 9:                 ]=

[ TYPE ] DETAIL IMPORT
```

4. **To view string register values too long for the screen**
   a. Move the cursor to the string register number.
   b. Press F2 DETAIL.
   c. Press PREV to return from the DETAIL screen.
5. **To add a comment**
   a. Move the cursor to the string register number and press ENTER.
   b. Move the cursor to select a method of naming the comment.
   c. Press the function keys whose labels correspond to the name you want to give to the comment. These labels vary depending on the naming method you chose in Step 5b

      For example, if you chose Upper Case, press a function key corresponding to the first letter. Press that key until the letter you want is displayed in the comment field. Press the right arrow key to move the cursor to the next space. Continue until the entire comment is displayed.

      To delete a character, press BACK SPACE.
   d. When you are finished, press ENTER.

      Note The comment appears in all teach pendant programs using that string register.
6. **To change the value**
   a. Move the cursor to the string register number on the right side of the = and press ENTER.
   b. Move the cursor to select a method of naming the value.
   c. Press the function keys whose labels correspond to the name you want to give to the value. These labels vary depending on the naming method you chose inStep 6b

For example, if you chose Upper Case, press a function key corresponding to the first letter. Press that key until the letter you want is displayed in the comment field. Press the right arrow key to move the cursor to the next space. Continue until the entire comment is displayed.

To delete a character, press BACK SPACE.

d. When you are finished, press ENTER.

7. To save the register values to the default device

a. Press MENUS.

b. Select FILE.

c. Press F1, [TYPE].

d. Select File.

e. Press F5, [UTIL].

f. Select Set Device.

g. Move the cursor to the device you want and press ENTER.

h. Press DATA.

i. Press F1, [TYPE].

j. Select Registers.

k. Press FCTN.

l. Select SAVE. The registers will be saved to the file, STRREG.VR, on the default device.

8. To import a string register text file

a. Press F3 IMPORT.

b. Move the cursor to the file to be imported.

c. Press ENTER.

Note An error message will be printed if any problems occur while importing the text file into the string registers.

Note Importing a string register text file will overwrite all current comments and values stored in the string registers.

The string register import text file can have two formats. The simple format is a file containing only values for each string registers. The full format is a file containing string register number, comment and values.

The simple format string register import text file contains a list of strings to be imported as the string register values. The values will be imported starting with string register number 1 and increment to the next string register number on the end of the line. Figure 8–4 shows the contents of an example file that will set value for string registers 1 through 5.

Figure 8–4. String Register Simple Import Text File

```
String Register Value 1
String Register Value 2
String Register Value 3
String Register Value 4
String Register Value 5
```

The full format string register import text file contains information to set a specific string register comment and value. The string register number is selected by the number followed by the colon (:) character at the beginning of the line. A comment is delimited by /* and */. The rest of the text after the comment close delimiter is considered as the string register value. Figure 8–5 shows the contents of an example file that will set comments and values for string registers 1, 4, 10 and 20.

Figure 8–5. String Register Full Import Text File

```
1:/*Comment 1*/String Register 1 with comment
4:/*Comment 4*/String Register 4 with comment
10:/*Comment 10*/String Register 10 with comment
20:/*Comment 20*/String Register 20 with comment
```

8.6 PALLET REGISTER STATUS

Pallet registers allow you to define the number of layers, rows, and columns that will be used to palletize. There are 127 pallet registers available.

Procedure 8-6 Displaying the Pallet Register Screen

Steps

1. Press MENUS.
2. Select DATA.
3. Press F1, [TYPE].
4. Select Pallet Reg. You will see a screen similar to the following.

```
DATA Pallet Reg
PL[  1:                ] =[  1,  1,  1]
PL[  2:                ] =[  1,  1,  1]
PL[  3:                ] =[  1,  1,  1]
PL[  4:                ] =[  1,  1,  1]
PL[  5:                ] =[  1,  1,  1]
PL[  6:                ] =[  1,  1,  1]
PL[  7:                ] =[  1,  1,  1]
PL[  8:                ] =[  1,  1,  1]
PL[  9:                ] =[  1,  1,  1]
PL[ 10:                ] =[  1,  1,  1]
```



Caution

Pallet registers are used in programs. Do not modify pallet register values unless you are sure how the pallet register is used in the system; otherwise, you could affect how programs are executed.

5. To add a comment
   a. Move the cursor to the pallet register number and press ENTER.
   b. Move the cursor to select a method of naming the comment.
   c. Press the function keys whose labels correspond to the name you want to give to the comment. These labels vary depending on the naming method you chose in Step 4b .

   For example, if you chose Alphabet, press a function key corresponding to the first letter. Press that key until the letter you want is displayed in the comment field. Press the right arrow key to move the cursor to the next space. Continue until the entire comment is displayed.
6. **To delete a character**, press BACK SPACE and then press ENTER.
7. To change the value of the pallet register
   a. Move the cursor to the pallet register value.
   b. Enter the new value.

8.7 SYSTEM VARIABLE STATUS

The SYSTEM Variables status screen displays system variables. You can change the value of several system variables using this screen. You can also change the value of a system variable in a program using the Parameter name instruction.

Refer to Section 6.11.10 . Use Procedure 8-7 to display and set system variables.

Refer to the*Software Reference Manual* for more information about system variables.

Procedure 8-7 Displaying and Setting System Variables



Warning

System variables control how the robot and controller operate. Do not set system variables unless you are certain of their effect; otherwise, you could injure personnel, damage equipment, or disrupt the normal operation of the robot and controller.

Steps

1. Press MENUS.
2. Select SYSTEM.
3. Press F1, [TYPE].
4. Select Variables. You will see a screen similar to the following.

```
SYSTEM Variables
 1 $ANGTOL
 2 $APPLICATION          [3] of STRING [21]
 3 $AP_MAXAX              0
 4 $AP_PLUGGED            2
 5 $AP_TOTALAX            16777216
 6 $AP_USENUM             [32] of BYTE
 7 $ASCII_SAVE            FALSE
 8 $AUTOINIT              2
 9 $BLT                   0
10 $CHECKCONFIG           FALSE
```

5. **To change the value of a system variable**
   a. Move the cursor to the variable you want to change.

      To move the cursor a group of lines at a time, press and hold the SHIFT key and press the up or down arrow key.

   b. Type the new value.
   c. Press ENTER, or press a function key as prompted.

6. **If the variable is an array**, a list of array elements is displayed or if the variable is a structure, a list of fields is displayed.
   a. Move the cursor to the element or field you want to set and press ENTER.
   b. Press PREV to return to the top level SYSTEM Variables screen.
   c. Enter the necessary information.
7. To save the variables to a file
   a. Press MENUS.
   b. Select FILE.
   c. Press F1, [TYPE].
   d. Select File.
   e. Press F5, [UTIL].
   f. Select Set Device.
   g. Move the cursor to the device you want and press ENTER.
   h. From any of the SYSTEM Variables screens, press FCTN.
   i. Select SAVE. All the system variables will be saved to the file, SYSVARS.SV, on the default device.



Warning

You must turn off the controller and turn on the controller to use the new information; otherwise, you could injure personnel or damage equipment.

8. When you are finished, turn off the controller. Then, turn on the controller so it can use the new information.

8.8 SAFETY SIGNAL STATUS

The STATUS Safety signal screen displays the status of safety-related control signals coming into the controller.

The safety signal screen displays the current state (TRUE or FALSE) of each safety signal. You cannot change the condition of the safety signal using this screen. Table 8–6 lists and describes each safety signal.

Use Procedure 8-8 to display safety signal status.

Table 8–6. Safety Signals

Safety Signal	Description
SOP E-Stop	This item indicates whether the EMERGENCY STOP button on the operator panel has been pressed. The status is TRUE if the operator panel EMERGENCY STOP button has been pressed.
TP E-Stop	This item indicates whether the EMERGENCY STOP button on the teach pendant has been pressed. The status is TRUE if the teach pendant EMERGENCY STOP button has been pressed.
Ext E-Stop	This item indicates whether an external emergency exists. The status is TRUE if the external emergency stop contacts are open on the cell connector EES1, EES11, EES2, or EES21.
Fence Open (AUTO STOP)	This item indicates whether the safety fence switch is open. The status is TRUE if the safety fence contacts are open on the cell connector EAS1, EAS11, EAS2, or EAS21.
TP Deadman	This item indicates when either the left or right teach pendant DEADMAN switch is pressed. The status is TRUE if either DEADMAN switch is pressed.
TP Enable	This item indicates whether the teach pendant ON/OFF switch is ON. The status is TRUE when the teach pendant ON/OFF switch is ON.
Hand Broken	This item indicates whether the safety joint switch in the robot hand has been tripped and the hand might be damaged. The status is TRUE when the safety joint switch has been tripped.
Overtravel	This item indicates whether the robot has moved beyond its overtravel limits. The status is TRUE when the robot has moved beyond its overtravel limits tripping the overtravel switch.
Low Air Alarm	This item indicates whether the air pressure has decreased below the acceptable limit. ***Low Air Alarm*** is usually connected to an air pressure sensing device. The status is TRUE when the air pressure is below the acceptable limit. You must set the $PARAM_GROUP[1].$PPABN_ENBL system variable to TRUE to use this signal.
Belt Broken	This item indicates whether a robot belt is broken. The status is TRUE when a robot belt is broken. This turns RDI7 on or off depending on how your system is set up. You must set the Use PPABN signal in the System/Config menu to TRUE to use this signal.
SVOFF Input (General Stop)	This item indicates whether the SVON input switch is open. The status is TRUE if the SVON input contacts are open on the cell connector EGS1, EGS11, EGS2, or EGS21.
Servo Disconnect	This item indicates whether the SERVO DISCONNECT input switch is open. The status is TRUE if the SERVO DISCONNECT input contacts are open on the operator panel PCB TBOP4 - SD4, SD41, SD5, or SD51.
Non Teacher Enabling Device (NTED)	This item indicates whether the NTED input switch is open. The status is TRUE if the NTED input contacts are open on CRM27 on the operator panel PCB.

Procedure 8-8 Displaying Safety Signal Status

Steps

1. Press MENUS.

2. Select STATUS.
3. Press F1, [TYPE].
4. Select Safety Signal. You will see a screen similar to the following.

```
STATUS Safety
     SIGNAL NAME        STATUS
  1  SOP E-Stop:        TRUE
  2  TP E-Stop:         FALSE
  3  Ext E-Stop:        FALSE
  4  Fence Open:        FALSE
  5  TP Deadman:        FALSE
  6  TP Enable:         FALSE
  7  Hand Broken:       FALSE
  8  Overtravel:        FALSE
  9  Low Air Alarm:     FALSE
  10 Belt Broken:       FALSE
  11 SVON Input:        FALSE
```

8.9 PROGRAM TIMER STATUS

The program timer screen displays the execution time for a program that contains TIMER instructions. TIMER instructions allow you to specify in your program when you want the timer to start, stop, or reset.

There are two teach pendant screens:

- Program timer listing screen
- Program timer DETAIL screen

Table 8–7 lists and describes each item on the program timer listing screen. Table 8–8 lists and describes each item on the program timer DETAIL screen.

Use Procedure 8-9 to display the program timer screen.

Table 8–7. Program Timer Listing Screen Items

ITEM	DESCRIPTION
Timer[ ]	This item indicates the number of the timer. You use this number in the TIMER instruction in your program. There are ten timers available.
Count	This item indicates the length of time, in seconds, that the program or section of program took to execute.
Comment	This item allows you to name or enter a comment about a timer.

Table 8–8. Program Timer DETAIL Screen Items

ITEM	DESCRIPTION
Timer[ ]	This item indicates the number of the timer. You use this number in the TIMER instruction in your program. There are ten timers available.
Count	This item indicates the length of time, in seconds, that the program (or section of program) took to execute.
Comment	This item allows you to name or enter a comment about a timer.
Start Program	This item indicates the name of the program that contains the TIMER start instruction.
Line	This item indicates the line number of the program where the counter started.
Stop Program	This item indicates the name of the program that contains the TIMER stop instruction.
Line	This item indicates the line number of the program where the counter stopped.

Procedure 8-9 Displaying the Program Timer Screen

Steps

1. Press STATUS.
2. Press F1, [TYPE].
3. Select Prg Timer. You will see a screen similar to the following. If you do not see this screen, press F2, LISTING.

```
PRG TIMER LISTING
                                    1/10
                 count      comment
1  Timer[ 1]      0.00(s)    [               ]
2  Timer[ 2]      0.00(s)    [               ]
3  Timer[ 3]      0.00(s)    [               ]
4  Timer[ 4]      0.00(s)    [               ]
5  Timer[ 5]      0.00(s)    [               ]
6  Timer[ 6]      0.00(s)    [               ]
7  Timer[ 7]      0.00(s)    [               ]
8  Timer[ 8]      0.00(s)    [               ]
9  Timer[ 9]      0.00(s)    [               ]
10 Timer[10]      0.00(s)    [               ]
```

4. **To add or change a comment,**
   a. Move the cursor to the comment line and press ENTER.
   b. Select a method of naming the comment.
   c. Press the appropriate function keys to add the comment.
   d. When you are finished, press ENTER.
5. **To display detailed information about a single timer,**
   a. Move the cursor to the timer you want.
   b. Press F2, DETAIL. You will see a screen similar to the following.

```
PRG TIMER DETAIL

 Timer[ 1]
    Comment    :    [****************]
    Count                  :        0.00 (sec)
    Start program          :  [         ]
    line                   :            0
    Stop program           :  [         ]
    line                   :            0
```

   c. **To display the listing screen**, press F2, LISTING.

8.10 SYSTEM TIMER STATUS

The system timer screen displays lengths of time system power has been on, running time, and waiting time. Table 8–9 lists and describes each item on the system timer screen.

Table 8–9. System Timer Screen Items

ITEM	DESCRIPTION
Group	This item allows you to display system timers for other groups. This item only functions if your system uses multiple groups.
Timer Type	This item indicates the different kinds of time that are counted: On power time, running time, waiting time, and servo on time.
Total(h)	This item indicates the total amount of time, in hours, for each of kind of timer. These totals cannot be changed or reset.
Lap(m)	This item indicates the amount of time, in minutes, a single cycle for each kind of timer takes to complete. Lap counts can be turned ON or OFF, and RESET.

Use Procedure 8-10 to display the system timer screen.

Procedure 8-10 Displaying the System Timer Screen

Steps

1. Press STATUS.
2. Press F1, [TYPE].
3. Select Sys Timer. You will see a screen similar to the following.

```
SYS TIMER
   GROUP  :  1
     Timer type      Total(h)   Lap(m)
   On Power time:       12.3      0.0 [OFF]
   Servo on time:       10.5      0.0 [OFF]
   Running time :        4.5      0.0 [OFF]
   Waiting time :        2.3      0.0 [OFF]
```

Note You will not be able to change the group number unless your system uses multiple motion groups.

4. To change the system timer display to a different motion group, press F2, GROUP#, and type the new motion group number.
5. Move the cursor to the timer type you want to turn on or off.
   - **To turn the lap counter on**, press F3, on/off until on is displayed.
   - **To turn the lap counter off**, press F3, on/off.
6. **To reset a lap counter**:
   a. Move the cursor to the timer type you want to reset.
   b. Press F3, ON/OFF until OFF is displayed to turn the timer type off.
   c. Press F4, RESET. You will see a screen similar to the following.

```
Reset are you sure[NO]?
                                YES     NO
```

   d. Press F4, YES, to reset the lap counter.

8.11 CLOCK STATUS

The clock menu displays the current date and time. Table 8–10 lists and describes each item and function key on the clock screen.

Table 8–10. Clock Screen Items

ITEM	DESCRIPTION
Date	This item displays the current date by year, month and then day.
Time	This item displays the current time using a 24 hour clock. The time is displayed by hour, minute, and then seconds.
[TYPE]	Press this key to access various application-specific options.
ADJUST	Press this key to modify the date or time on the system.
FINISH	Press this key when you are finished modifying the date or time on the system.

Use Procedure 8-11 to display the clock screen.

Procedure 8-11 Displaying the Clock Screen

Steps

1. Press MENUS.
2. Select SYSTEM.
3. Press F1, [TYPE].
4. Select Clock. You will see a screen similar to the following.

```
SYSTEM Display
    DATE               xx/01/01
    TIME               18:56:28
Please select function

To change the date or time display
```

5. **To change the date or time display**, press F4, ADJUST and enter the new information.
6. **When you are finished setting the clock**, press F4, FINISH.

8.12 VERSION IDENTIFICATION STATUS

The STATUS Version ID screen displays information specific to your controller. Use this information when you call the FANUC Robotics Hotline if a problem occurs with your controller. You cannot change the information displayed on this screen.

Table 8–11 through Table 8–14 list and describe the version identification status information.

Table 8–11. STATUS Version ID SOFTWARE Screen Items

ITEM	DESCRIPTION
application- Tool	This item lists the Application Tool software that is loaded. The "ID:" column for this item shows the most important version number for Robot Controllers located in Asia.
S/W Serial No.	This item is a unique number that identifies the software installed on your controller. This is similar to a Customer Order Number.
Controller ID	This item is a unique number that identifies your controller.
Robot Model	This item is the model name of the robot you are using. The "ID:" column for this item shows the Version Number of the Robot Library.

Table 8–11. STATUS Version ID SOFTWARE Screen Items (Cont'd)

Servo Code	This item is the version number of the robot servo software.
Cart. Mot. Parameter	This item is the version number of the Cartesian motion parameters.
Joint Mot. Parameter	This item is the version number of the joint motion parameters.
Software Edition No.	This item is the version number of the software. This item shows the most important version number for Robot Controllers located in North America.
Customization Ver.	This item is the version number of the customization, if applicable. It will show "None" if there is no Customization installed or the if Customization does not set this Version.
Root Version	This item is the version number of the Root software.
Boot MONITOR	This item is the version number of the BMON Flash ROM software.
Teach Pendant	This item is the version number of the teach pendant EPROM.
Browser Plugins	This item is the version number of the browser plugins.
TP Core Firmware	This item is the version number of the software item loaded.

Table 8–12. STATUS Version ID CONFIG Screen Items

ITEM	DESCRIPTION
FEATURE	This item is the name of the installed feature or option.
ORD. NO (order number)	This item uniquely identifies the feature or option.
application-Tool	This item is the software application tool that is installed.
All other items	These items list the features and options that are installed on your controller.

Table 8–13. STATUS Version ID MOTOR Screen Items

ITEM	DESCRIPTION
GROUP	This item is the number of the motion group associated with the displayed robot axis.
AXIS	This item is the number of the displayed robot axis.
MOTOR ID AND INFO	These items are the motor name (ID) and information that uniquely identify and describe the motor used for the displayed robot axis.

Table 8–14. STATUS Version ID SERVO Screen Items

ITEM	DESCRIPTION
GROUP	This item is the number of the motion group associated with the displayed robot axis.
AXIS	This item is the number of the displayed robot axis.
SERVO PARAM ID	This item is the name that identifies the group of servo parameters associated with the displayed robot axis.

Use Procedure 8-12 to display version identification status.

Procedure 8-12 Displaying the Version Identification Status

Steps

1. Press STATUS.
2. Press F1, [TYPE].
3. Select Version ID.
4. Press the key that corresponds to the version ID status screen you want to display:

   - **To display software version information**, press F2, SOFTWARE. You will see a screen similar to the following.

```
STATUS Version ID
     SOFTWARE:              ID:    1/15
  1  application—Tool       xxxx/xx
  2  S/W Serial No.         xxxxx
  3  Controller ID          F00000
  4  Robot Model, application—Tool
  5  Robot Model            Vx.xxxx
  6  Servo Code             12345678
  7  Cart. Mot. Parameter   V3.00
  8  Joint Mot. Parameter   V3.00
  9  Software Edition No.   Vx.xxxx
 10  Customization Ver.     None
 11  Root Version           Vx.xxxx
 12  Boot MONITOR           Vx.xxxx
 13  Teach Pendant          xxxx/xxx
 14  Browser Plugins        Vx.xxxx
 15  TP Core Firmware       Vx.xxxx
```

   - **To display a list of the software features and options that are loaded**, press F3, CONFIG.

```
STATUS Version ID
     SOFTWARE:              ORD NO:  1/128
1:
application
-Tool (TM)   XXXX
2:  English Dictionary      XXXX
3:  Kernel Software         XXXX
4:  Analog I/O              XXXX
5:  Arc EQ Program Select   XXXX
6:  Arc Softpart            XXXX
7:  Background Edition      XXXX
8:  Basic Software          XXXX
9:  Controller Backup       XXXX
10: Cycle Time Priority     XXXX
```

Note The information displayed here could be different at your site.

- **To display motor types for each axis**, press F4, MOTOR. You will see a screen similar to the following.

```
STATUS Version ID
   GR: AX: MOTOR ID AND INFO:       1/16
1   1  1  ACA22/2000 80A     H1  DSP1-L
2   1  2  ACAM30/3000HV 80A  H2  DSP1-M
3   1  3  ACA22/2000 80A     H3  DSP2-L
4   1  4  ACAM9/3000 40A     H4  DSP2-M
5   1  5  ACAM6/3000 40A     H5  DSP3-L
6   1  6  ACAM6/3000 40A     H6  DSP3-M
7 **  **   ***************************
8 **  **   ***************************
9 **  **   ***************************
10 **  **   ***************************
```

- **To display the servo parameters for each axis**, press F5, SERVO. You will see a screen similar to the following.

```
STATUS Version ID
    GROUP: AXIS:  SERVO PARAM ID:  1/16
1    1      1     P01.01
2    1      2     P01.01
3    1      3     P01.01
4    1      4     P01.01
5    1      5     P01.01
6    1      6     P01.01
7  **     **      ********
8  **     **      ********
9  **     **      ********
10 **     **     ********
```

8.13 MEMORY STATUS

Use Procedure 8-13 to display memory status.

8.13.1 Checking Memory

Table 8–15 lists and describes each memory status item. Use Procedure 8-13 to check system memory. You can also modify the allocation of SRAM memory for selected software products.

Table 8–15. Memory Status

MEMORY STATUS	DESCRIPTION
Pools	These items indicate the amount of memory for • **TPP** contains teach pendant programs. TPP memory is stored in permanent memory. Therefore, to determine the amount of memory available for teach pendant programs, you must subtract the total amount of TPP SRAM memory shown from the total amount of permanent memory shown. • **PERM** contains system variables and KAREL variables. • **SYSTEM** contains the operating system. • **SHADOW** contains variables designated to be in SHADOW memory. • **TEMP** contains temporary memory used for system operations, and KAREL programs. • **FR FROM** Flash ROM is file oriented and all of it is allocated to the Flash File system. The flash file system contains both system and user files.
Hardware	These items indicate the total amount of memory for • **FROM** Flash ROM • **DRAM** • **SRAM**

Procedure 8-13 Displaying Memory Status

Steps

1. Press MENUS.
2. Select STATUS.
3. Press F1, [TYPE].
4. Select Memory. You will see a screen similar to the following.

```
STATUS Memory
           Total      Free   Lrgst Free
 Pools    ----------------------------
   TPP      700.0KB   687.8KB   687.5KB
   PERM    2040.0KB   807.1KB   806.4KB
   SHADOW  1581.1KB  1421.7KB    32.0KB
   TEMP   32714.5KB 21329.7KB 21326.1KB
   FR     16334.0KB  7410.0KB
 Description: ----------------------------
 TPP:   Used by .TP, .MR, .JB, .PR
 PERM:  Used by .VR, RD:, Options
 TEMP:  Used by .PC, .VR, Options
```

5. To display the DETAIL screen press F2, DETAIL. You will see a screen similar to the following.

```
STATUS Memory
           Total      Free   Lrgst Free
 Pools    ----------------------------
   TPP      700.0KB   687.8KB   687.5KB
   PERM    2040.0KB   807.1KB   806.4KB
   SYSTEM  5120.0KB   678.1KB   678.1KB
   SHADOW  1581.1KB  1421.7KB    32.0KB
   TEMP   32714.5KB 21329.7KB 21326.1KB
   FR     16334.0KB  7410.0KB
 Hardware ----------------------------
   FROM     16.0 MB (T)
   DRAM     32.0 MB
   SRAM      2.0 MB (CMOS)
```

Note (T) indicates the FROM hardware manufacturer.

6. To display the first screen, press F2, BASIC.

Note Software options require TEMP and PERM memory.

8.14 POSITION STATUS

The POSITION screen displays positional information in joint angles or Cartesian coordinates. The positional information on this screen is updated continuously when the robot moves. The information displayed on these screens is for display only; you cannot change it.

Note E1, E2, and E3 indicate extended axis positional information if extended axes are installed in your system.

Refer to Section 6.4.3 for a description of positional information.

Joint

The joint screen displays positional information in degrees for each robot axis. Tool indicates the number of the active tool frame.

Table 8–16. POSITION Joint Screen Items

ITEM	DESCRIPTION
J1 / J2 / J3 / J4 / J5 / J6	These items display the current position of J1, J2, J3, J4, J5, and J6.

User

The user screen displays positional information in Cartesian coordinates based on the user frame. Tool indicates the number of the active tool frame. Frame indicates the number of the active user frame.

Table 8–17. POSITION User Screen Items

ITEM	DESCRIPTION
Configuration	This item displays the current configuration of the User frame.
x / y / z / w / p / r	These items display the x, y, z, w, p, and r coordinates of the current User frame.

World

The world screen displays positional information in Cartesian coordinates based on the world frame. Tool indicates the number of the active tool frame.

Table 8–18. POSITION World Screen Items

ITEM	DESCRIPTION
Configuration	This item displays the current configuration of the World frame.
x / y / z / w / p / r	These items display the x, y, z, w, p, and r coordinates of the current World frame.

Use Procedure 8-14 to display position status.

Procedure 8-14 Displaying Position Status

Steps

1. Press POSN.
2. Select the appropriate coordinate system.

   - **For joint**, press F2, JNT. You will see a screen similar to the following.

```
POSITION
  Joint                        Tool: 1
J1:     .001 J2:   10.028 J3:   -35.025
J4:    -.000 J5:   34.998 J6:     .001
E1:     .000 E2:     .001 E3:     .001
```

Note E1: , E2:, and E3: are displayed only if you have extended axes.

   - **For user**, press F3, USER. You will see a screen similar to the following.

```
POSITION
User                  Frame: 0     Tool: 1
Configuration: F, 0, 0, 0
  x: 1906.256  y:     .041  z:  361.121
  w:  178.752  p:  -89.963  r:    1.249
 E1:     .001  E2:    .001  E3:    .001
```

   - **For world**, press F4, WORLD. You will see a screen similar to the following.

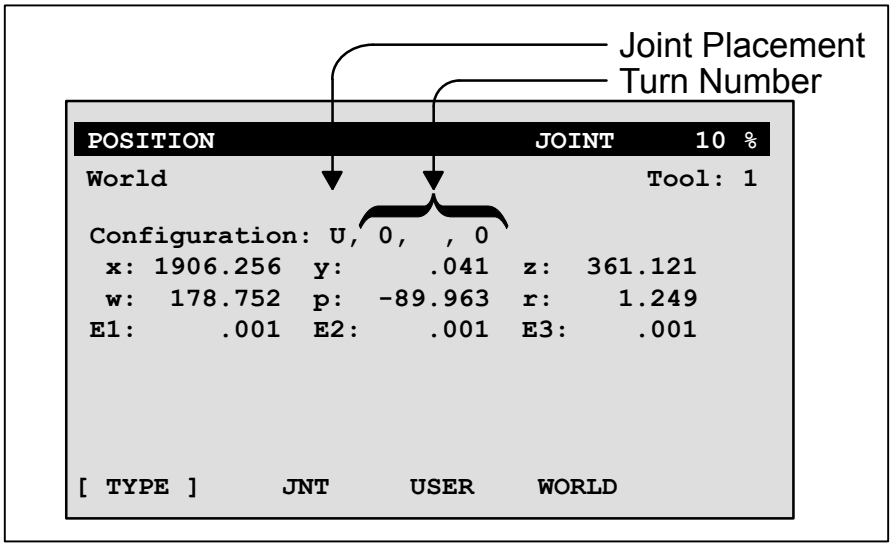
```
POSITION
World                              Tool: 1
Configuration: F, 0, 0, 0
  x: 1906.256  y:     .041  z:  361.121
  w:  178.752  p:  -89.963  r:    1.249
 E1:     .001  E2:    .001  E3:    .001
```

8.15 TURN NUMBER DISPLAY

8.15.1 Overview

Turn number display specifies the turn number displayed on the teach pendant screen. Figure 8–6 shows an example of where joint placement and turn number information is displayed on the POSITION screen. Refer to Section 8.14 for more information.

Figure 8–6. Turn Number and Joint Placement Display on Position Screen



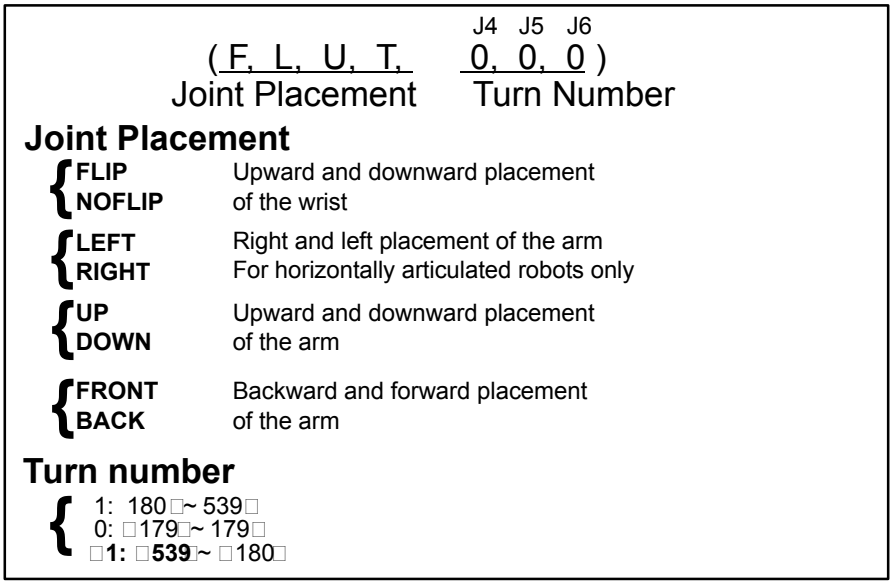
- For most robot models, the position data is usually represented in the format shown in Section 8.15.2 .
- For some robot models the system variable $SCR_GRP[group].$turn_axis[i] must be used to set position data. Refer to Section 8.15.3 .

8.15.2 Usual Configuration

The system variable $SCR_GRP[group].$TURN_AXIS[axis_number] defines the mapping of turn numbers to axes. This mapping varies with robot model and configuration. Refer to the *Software Reference Manual* for more information.

See Figure 8–7 .

Figure 8–7. Turn Number Display Configuration



In general, when the robot arm can reach the same TCP with the arm bent differently, a configuration identifier is needed to specify the joint placement. The joint placement varies for fully articulated robots, such as the M-710i, and horizontally articulated robots, such as the A-520i. See Figure 8–8 through Figure 8–10 .

Figure 8–8. Joint Placement Configuration Examples for Fully Articulated Robots

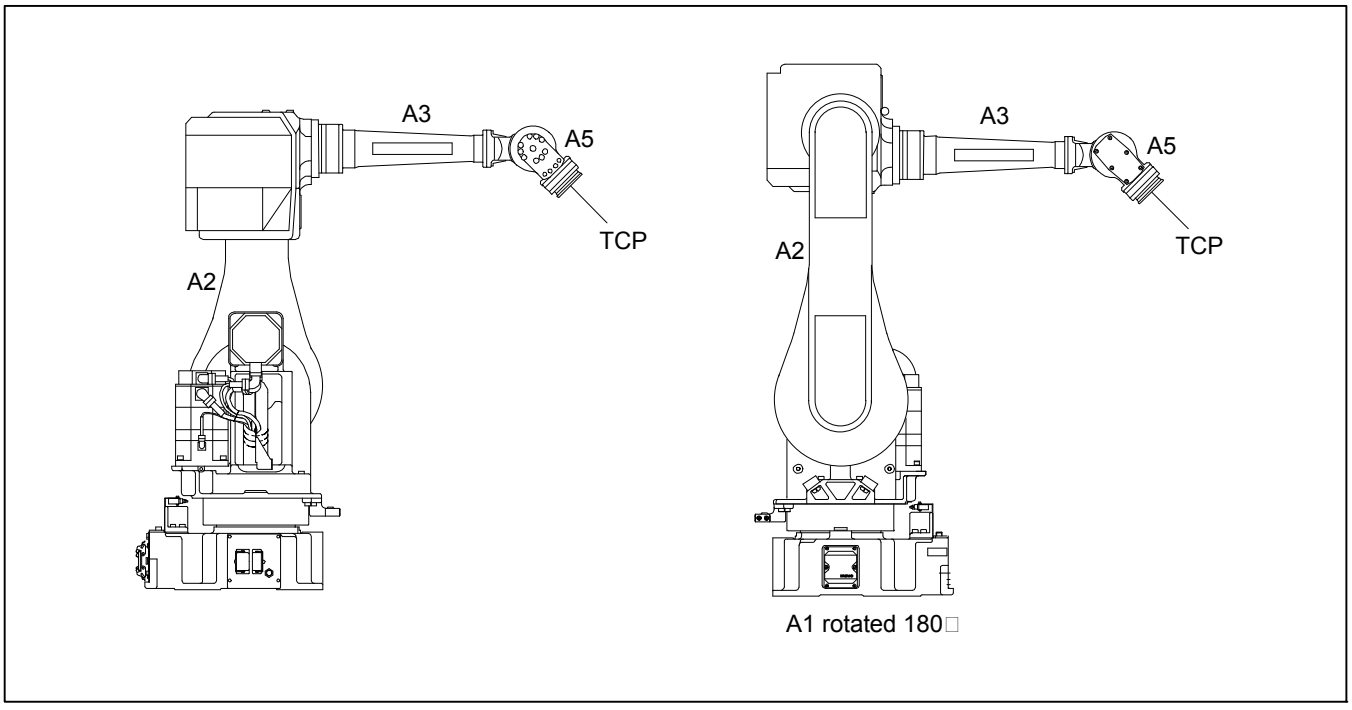


Figure 8–9. Joint Placement Configuration Examples for Fully Articulated Paint Robots

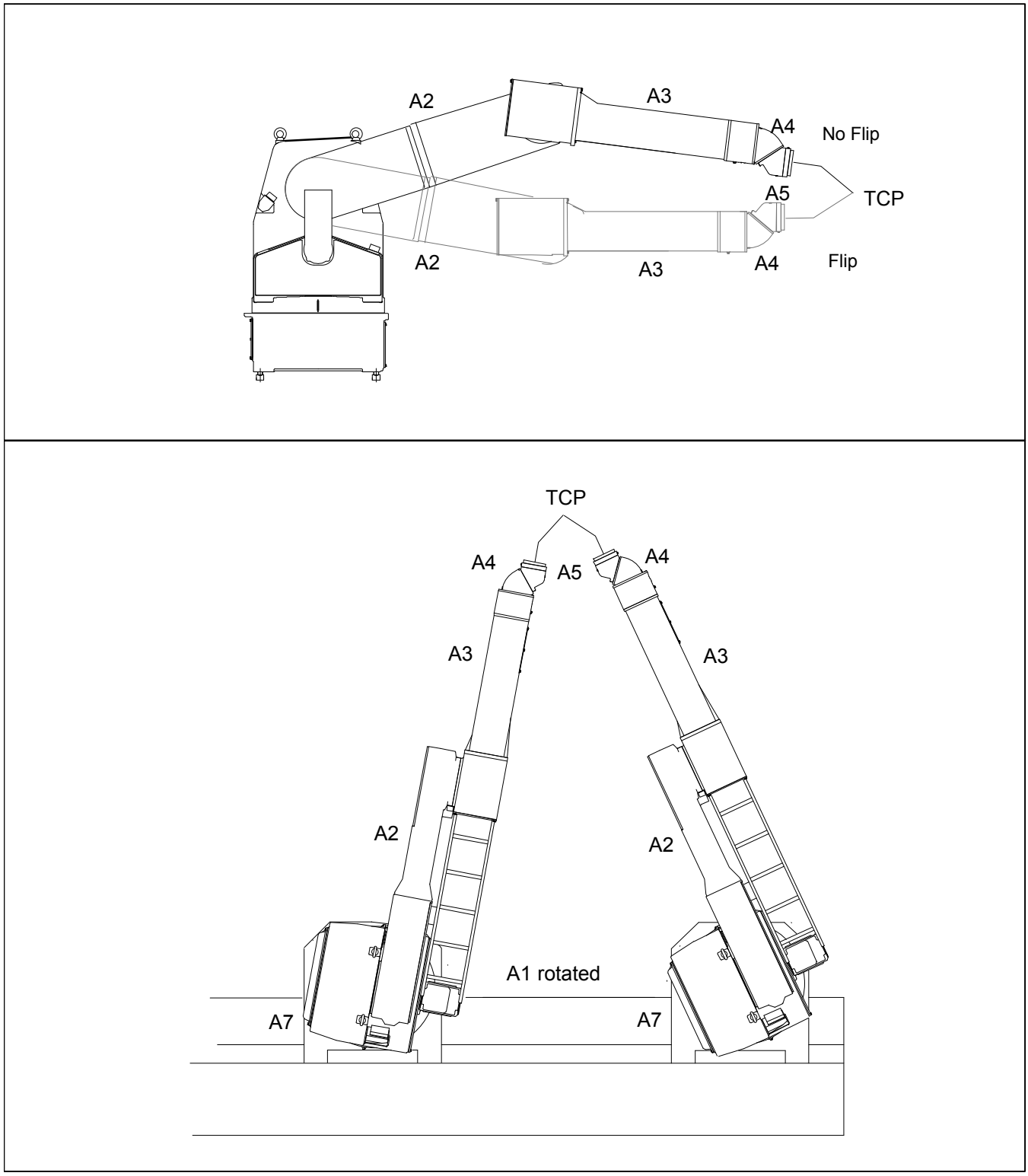
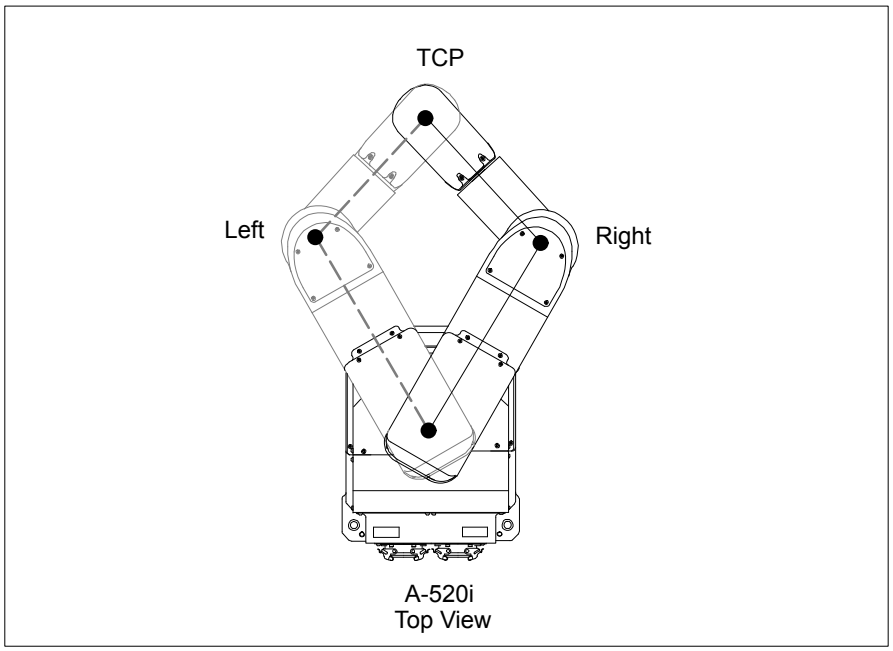


Figure 8–10. Joint Placement Configuration Examples for Horizontally Articulated Robots



8.15.3 $SCR_GRP[group].$turn_axis[i] System Variable

If a robot model has a turn number for the J1 axis, the usual configuration shown in Figure 8–7 will not represent the positional data. To represent the positional data in this case, the system will set the system variable $SCR_GRP[group].$tur n_axis[i] (where i = 1, 2, or 3) to the appropriate value as shown in Figure 8–11 .

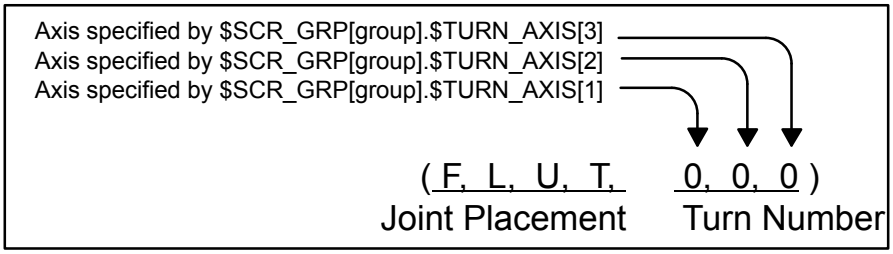
For models with a turn number for the J1 axis, such as the S-430i, the values of the system variable are as follows:

$SCR_GRP[group].$TURN_AXIS[1]=1

$SCR_GRP[group].$TURN_AXIS[2]=4

$SCR_GRP[group].$TURN_AXIS[3]=6

Figure 8–11. $SCR_GRP[group].$turn_axis[i] for Turn Number Display Configuration



In Figure 8–11 $SCR_GRP[group].$tu rn_axis[i] specifies the turn numbers for the robot axes. (These correspond to axes J4, J5, and J6 in the usual configuration.)

8.16 EXECUTION HISTORY

The program execution history function records the execution history of the most recently executed or halted program and enables checking of the execution history when the program terminates or halts.

Note Execution history of a program that is currently running cannot be checked.

The information displayed for each running program is listed and described in Table 8–19 .

Table 8–19. Execution History Screen

ITEM	DESCRIPTION
Program name	This item displays the name of the program being executed.
Line	This item displays the program line number.
Direction	This item displays the direction in which the specified line number is executed: forward (FWD) or backward (BWD).
Status	This item displays the execution status of the specified line number: • Not exec - The line was read but has not yet been executed. • Paused - While the line was being executed, the program halted (incomplete execution). • Done - The line was executed completely. • Aborted - The program was terminated.

When a program has been aborted, the information displayed in item 1 of the screen will contain a blank program name, a line number of 0, a blank direction, and a status of Aborted.

In standard execution history, up to 20 lines can be recorded. The number of lines to be recorded can be changed at Controlled start. Refer to Appendix C . Information that is older than the lines recorded in the execution history is erased automatically.

When Execution History is Not Recorded

Execution history is not recorded in the following cases:

- If a macro is executed using a method other than a program (such as MANUAL FCTNS or a user key), execution history is not recorded. If a program assigned to a macro is executed from the program editing screen, the execution history is recorded with the assigned program name, not with the macro name.
- The execution history of a KAREL program is not recorded, even if a KAREL program is executed.
- The execution history of a program that is executed automatically when power is turned on is not recorded.

Use Procedure 8-15 to display the execution history screen.

Procedure 8-15 Displaying Program Execution History

Steps

1. Press STATUS.
2. Press F1, [TYPE].
3. Select Exec-hist. You will see a screen similar to the following.

```
Execution History
   Program name           Line. Dirc.    Stat.
1                         0              Aborted
2  PROG987                9     FWD      Done
3  PROG987                8     FWD      Done
4  PROG987                7     FWD      Done
5  PROG987                6     FWD      Done
6  PROG987                5     FWD      Done
7  PROG987                4     FWD      Done
8  PROG987                3     FWD      Done
9  PROG987                2     FWD      Done
 Press NEXT to display other task
```

When a program has been aborted, the information displayed in item 1 of the screen will contain a blank program name, a line number of 0, a blank direction, and a status of Aborted.

Note To change the display from the first to the second to the third to the fourth and then back to the first program, press F2, NEXT. If you display the second program, for example, you will have to press F2, NEXT, three times to display the first program again.

4. **To display information for any of three other simultaneously running programs,** press F2, NEXT. To return to the first display, press F2, NEXT, until the program you want is displayed.

   Note You can clear status information only for programs that have been aborted.

5. **To clear the status information for the aborted program you are viewing,** press and hold SHIFT and press F5, CLEAR.
6. **To clear the status information for all aborted programs,** press and hold SHIFT and press F4, ALL_CLR.

8.17 PROCESS AXES STATUS (OPTION)

The Process Axes Status screen displays speed and torque data for all process axes defined on the controller. This screen is only available when the Basic Process Axes option is loaded on the controller. See figure Figure 10–9 for an example screen and table Table 10–8 for a description of the items on this screen.

Figure 8–12. Process Axes Status Screen

```
STATUS ISDT

             User    Motor     Motor
      Axis   Speed   RPM       Amps
        1      0.0     0.0      0.00
        2     51.2   301.0      1.41
        3   ******  ******    ******
        4   ******  ******    ******
        5   ******  ******    ******
        6   ******  ******    ******
        7   ******  ******    ******
        8   ******  ******    ******
```

Table 8–20. Process Axes Status Items

ITEM	DESCRIPTION
Axis	This item is the axis number whose data is displayed in this row.
User Speed	This item is the current speed of the axis in user units (i.e. units of the TP instruction). A "******" indicates that the axis is not defined.
Motor RPM	This item is the current speed of the axis in RPM at the motor shaft. A "******" indicates that the axis is not defined.
Motor Amps	This item is the commanded current in amps from the servo amplifier for this axis. A "******" indicates that the axis is not defined.

8.18 PROGRAM or PRODUCTION STATUS

The program status screens display information about the running teach pendant or KAREL program, and the running routine if a KAREL program is running. The ProductionStatus screen can be used to display the status of menus that you have created with the Menu Utility. Table 8–21 lists and describes each ProductionStatus item.

Use Procedure 8-16 to display the program status screens.

Table 8–21. Program Status or Production Status Items

ITEM	DESCRIPTION
Current Program	This item displays the name of the teach pendant or KAREL program that is currently being executed.
Current Routine	This item displays the name of the KAREL routine that is being executed, if there is a KAREL program currently being executed.
Current Line	This item indicates the line of the teach pendant or KAREL program that is currently being executed.

Table 8–21. Program Status or Production Status Items (Cont'd)

ITEM	DESCRIPTION
Status	This item displays the status of the teach pendant or KAREL program that is currently being executed. The status of a running program can be • RUNNING • PAUSED • ABORTED • MOVING • WAITING • WAITING FOR DI[n] • WAITING FOR RI[n]
Time	This item displays the current time according to the system clock.

Procedure 8-16 Displaying the Program or Production Status Screens

Steps

1. Press MENUS.
2. Select STATUS.
3. Press F1, [TYPE].
4. Select Prod. Status. The Prod. Status screen will be displayed. See the following screen for an example.

```
STATUS Program
            Prod.Status
 Current Program
 Current Routine
 Status                  ABORTED
 Time                    15-FEB-xx 17:52
```

5. **To display the Production Status screen,** press F4, [CHOICE], and select ProductionStatus. The Production Status screen will be displayed. See the following screen for an example.

```
STATUS Program
 1 ProductionStatus 5
 2                  6
 3                  7
 4                  8 -- NEXT --
STATUS Program
            ProductionStatus
 Current Program
 Current Routine
 Current Line           0
 Status                 ABORTED
 Time                   15-FEB-xx 17:52
```

8.19 ROBOT AXES STATUS

8.19.1 Overview

The robot axes status screen displays information for each robot axis. This information is continually updated. Use this information when you call the FANUC Robotics Hotline if a problem occurs with your robot.

This screen displays the following status information for robot axes:

- Status 1
- Status 2
- Pulse
- Torque monitor
- Tracking
- Disturbance torque
- Servo diagnosis

8.19.2 Status 1 Screen

The Status 1 screen displays the alarm status of the servo system. The status information consists of Flag Bits 1 (16 bits), shown on the upper row of bits for each axis, and Flag Bits 2 (16 bits), shown on the lower row of bits for each axis.

See Figure 8–13 for an example Status 1 screen and Table 8–22 through Table 8–26 for descriptions of the items on this screen.

Figure 8–13. Status 1 Screen

```
STATUS Axis

                                GRP[ 1 ]
      Flag Bits 1/2           History
 J1: 0000000000000000 (0000000000000000)
     0000000000000000 (0000000000000000)
 J2: 0000000000000000 (0000000000000000)
     0000000000000000 (0000000000000000)
 J3: 0000000000000000 (0000000000000000)
     0000000000000000 (0000000000000000)
 J4: 0000000000000000 (0000000000000000)
     0000000000000000 (0000000000000000
```

Table 8–22. Status 1 Screen Items

ITEM	DESCRIPTION
Flag Bits 1	This item provides servo alarm status 1 in 16 bits. The meanings of the bits are described in Table 8–23 .
Flag Bits 2	This item provides servo alarm status 2 in 16 bits. The meanings of the bits are described in Table 8–26 .
History	This item provides the most recent, non-current, values of the Flag Bits.

Table 8–23. Servo Alarm Status 1; Address: FC80h (L-axis), FCC0h (M-axis)

MSB	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	LSB
OHAL	LVAL	OVC	HCAL	HVAL	DCAL	FBAL	ALDF	MCAL	MOFAL	EROFL	CUER	SSTB	PAWT	SRDY	SCRDY

Table 8–24. Alarm Terminology

Alarm Type	Alarm Description
OHAL	Amplifier overheat alarm
LVAL	It indicates a low voltage alarm.
OVC	It indicates an overcurrent (OVC) alarm.
HCAL	It indicates a high current alarm.
HVAL	It indicates a high voltage alarm.

Table 8–24. Alarm Terminology (Cont'd)

Alarm Type	Alarm Description
DCAL	It indicates a regenerative discharge alarm.
FBAL	Disconnection alarm (ALDF indicates whether the disconnection is associated with the hardware or software.)
ALDF	Alarm distinction bit If an amplifier alarm (OHAL, LVAL, HCAL, FSAL, IPMAL, or DCLVAL) is raised while ALSF is set to 1, the alarm is detected by PSM. When both FBAL and ALDF are set to 1, the disconnection alarm is detected by the hardware.
MCAL	Amplifier MCC adhesion alarm
MOFAL	Move command overflow alarm When this bit is set to 1, it indicates that an overflow occurred when the move command was distributed.
EROFL	Error counter overflow alarm for line tracking When this bit is set to 1, it indicates that the error counter has overflowed.
CUER	Current offset error This bit is set to 1 when the current offset value of the A/D converter is higher than permitted.
SSTB	Servo standby signal After POWON, this signal is set to 1, and the system waits for ITP. When SSTB is set to 1, the host outputs ITPCON and generates ITP.
PAWT	Parameter change completion signal When the servo CPU finishes rewriting parameters, only 1 ITP is set to 1.
SRDY	Servo ready signal While this flag is held to 1, a move command is accepted.
SCRDY	Servo communication flag The servo CPU sets this flag to 1 once data writing to the shared RAM is completed. After reading the data, the host CPU resets the flag to 0.

Table 8–25. Description of Alarm Combinations

OVL	FBAL	ALDF	Alarm
1	0	1	Motor overload alarm (not used for a serial pulse coder)
1	0	0	Amplifier overload alarm
0	1	1	Pulse coder disconnection alarm (not used for a serial pulse coder)

Table 8–26. Servo Alarm Status 2; Address: FC81h (L-axis), FCC1h (M-axis)

MSB	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	LSB
SRCMF	CLALM	FSAL	DCLVAL	BRAKE	IPMAL	SFVEL	GUN SET	FSSB DC	SCU CAL	AMU CAL	CHG AL	NOA MP			

Table 8–27. Alarm Terminology

Alarm Type	Alarm Description
SRCMF	Compensation warning flag When part of the position data is missing because of noise or some other reason, data compensation is performed. This data, however, should not be used for mastering or other purposes. To inform the host of this state, the flag is set to 1.
CLALM	It indicates a collision detection alarm. When the servo CPU detects a collision, the flag is set to 1. The host CPU starts alarm handling after a lapse of a predetermined period from when the flag is set to 1.
FSAL	Fan stop alarm
DCLVAL	Low DC Link voltage alarm
BRAKE	Brake alarm of 6-axis amplifier
IPMAL	IPM alarm IPM is an abbreviation for intelligent power module, which is a power component to replace IGBT. The IPM detects overheating and HC by itself.
SFVEL	Soft float start permission signal When the velocity feedback falls below the velocity specified in a parameter, this flag is set to 1 to allow soft float to be started.
GUNSET	Servo gun switch completion signal Once the resetting (initialization) of the pulse coder has been completed after the servo gun is switched, the signal is set to 1 only for 1 ITP.
FSSBDC	FSSB disconnection alarm When a disconnection of FSSB is detected, this bit is set to 1. (Hardware detection by FSSBC)
SVUCAL	FSSB communication alarm When two consecutive alarms are detected in data communication between the slave and a servo module, this bit is set to 1. (Detected by the servo software)
AMUCAL	FSSB communication alarm When two consecutive alarms are detected in data communication between the servo module and a slave, this bit is set to 1. (Detected by the slave)
CHGAL	Amplifier charge alarm
NOAMP	No amplifier connection alarm This bit is set to 1 when an amplifier is not connected while the presence of the corresponding axis is specified (B3 of AXIS register set to 0).

8.19.3 Status 2 Screen

The Status 2 screen indicates the pulse coder alarm status in 12 bits.

See Figure 8–14 for an example Status 1 screen and Table 8–28 and Table 8–29 for descriptions of the items on this screen.

Figure 8–14. Status 2 Screen

```
STATUS Axis

                                GRP[ 1 ]
      Alarm Status     History
 J1 : 000000000000 ( 000000000000 )
 J2 : 000000000000 ( 000000000000 )
 J3 : 000000000000 ( 000000000000 )
 J4 : 000000000000 ( 000000000000 )
 J5 : 000000000000 ( 000000000000 )
 J6 : 000000000000 ( 000000000000 )
```

Table 8–28. Status 2 Screen Items

ITEM	DESCRIPTION
Alarm Status	This item provides the pulse coder alarm status in 12 bits. The meanings of the bits are described in Table 8–29 .
History	This item provides the most recent, non-current, values of the alarm status bits.

Table 8–29. Pulse Coder Alarm Status

MSB	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	LSB
SPHAL	STBERR	CRCERR	DTERR	OHAL	CSAL	BLAL	PHAL	RCAL	BZAL	CKAL	

Table 8–30. Alarm Terminology

Alarm Type	Alarm Description
SPHAL	When this bit is 1, it indicates a soft phase alarm (abnormal acceleration).
STBERR	When this bit is 1, it indicates a start/stop bit alarm.
CRCERR	When this bit is 1, it indicates a CRC alarm.
DTERR	When this bit is 1, it indicates a data alarm.
OHAL	When this bit is 1, it indicates a over heat alarm.

Table 8–30. Alarm Terminology (Cont'd)

Alarm Type	Alarm Description
CSAL	When this bit is 1, it indicates a check sum alarm.
BLAL	When this bit is 1, it indicates the low voltage alarm of the battery.
PHAL	When this bit is 1, it indicates a phase alarm.
RCAL	When this bit is 1, it indicates a rotating speed counter abnormal alarm
BZAL	When this bit is 1, it indicates an exhausted battery alarm.
CKAL	When this bit is 1, it indicates a clock alarm.

8.19.4 Pulse Screen

The Pulse screen displays the servo delay, machine position, and status of the motion command.

See Figure 8–15 for an example screen and Table 8–31 for descriptions of the items on this screen.

Figure 8–15. Pulse Screen

```
STATUS Axis

                                GRP[ 1 ]
      Position    Machine     Motion
        Error       Pulse       Command
 J1 :         0           0           0
 J2 :         0           0           0
 J3 :         0           0           0
 J4 :         0           0           0
 J5 :         0           0           0
 J6 :         0           0           0
```

Table 8–31. Pulse Screen Items

ITEM	DESCRIPTION
Position Error units: pulses	This item is the servo delay. This item displays the difference between the commanded APC count versus the actual APC count.

Table 8–31. Pulse Screen Items (Cont'd)

ITEM	DESCRIPTION
Machine Pulse units: pulses	This item is the machine position. These are actual absolute pulses. This item shows the actual APC count as read by the controller.
Motion Command units: pulses	This item indicates the relative command pulses from the host. This item displays the desired value of the Absolute Pulse Coder (APC) when the robot gets to the position commanded by the controller.

8.19.5 Torque Monitor Screen

The Torque Monitor screen displays the current values, and the status of the position, overtravel, and servo amplifier. Load to the motor and thermal loss can be estimated using the root-mean-square current values.

See Figure 8–16 for an example screen Table 8–32 for descriptions of the items on this screen.

Figure 8–16. Torque Monitor Screen

```
STATUS Axis

                                GRP[ 1 ]
     Torque Monitor
           Ave. / Max.     Inpos OT VRDY
 J1 :      0.000/     0.000  1   0  OFF
 J2 :      0.000/     0.000  1   0  OFF
 J3 :      0.000/     0.000  1   0  OFF
 J4 :      0.000/     0.000  1   0  OFF
 J5 :      0.000/     0.000  1   0  OFF
 J6 :      0.000/     0.000  1   0  OFF
```

Table 8–32. Torque Monitor Items

ITEM	DESCRIPTION
Average units: Amps	This item is the average of the root-mean-square current values.
Maximum units: Amps	This item is the maximum of the root-mean-square current values.
Inposition	This item is the position status, 0 or 1. 0 means that the robot is not in position. 1 means that the robot is in position.

Table 8–32. Torque Monitor Items (Cont'd)

ITEM	DESCRIPTION
OT	This item is the overtravel status, 0 or 1. 0 means the robot is not in overtravel. 1 means the robot is in overtravel.
VRDY	This item is the servo amplifier ready status, ON or OFF. ON means the servo amplifier is ready, OFF means the servo amplifier is not ready.

8.19.6 Tracking Screen

The Tracking screen displays the status of the tracking servo system.

See Figure 8–17 for an example screen and Table 8–33 for a description of the items on this screen.

Figure 8–17. Tracking Screen

```
AXIS Status
  Tracking Status

       Flag Bits 1      Flag Bits 2
 P1 : 0000000000000000 0000000000000000
 P2 : 1111111111111111 0111111111111111

     Alarm Status    Counter Value
 P1 : 000000000000              0
 P2 : 111111111111       ********
```

Table 8–33. Tracking Screen Items

ITEM	DESCRIPTION
Flag Bits 1	This item is the servo alarm status 1, in 16 bits. The meanings of the bits are described in Table 8–23 .
Flag Bits 2	This item is the servo alarm status 2, in 16 bits. The meanings of the bits are described in Table 8–26 .
Alarm Status	This item is the pulse coder alarm status, in 12 bits. The meanings of the bits are described in Table 8–29 .
Counter Value	This item is the line tracking counter.

8.19.7 Disturbance Torque Screen

The Disturbance Torque screen displays the disturbance torque to each motor (current torque and maximum and minimum torque for each ITP) The disturbance torque is indicated with the current values estimated from the difference between the scheduled and actual values of the pulse coder. If the maximum or minimum value set for the disturbance torque is exceeded, the collision detection function of the servo system regards a collision as occurring and turns off the servo power.

See Figure 8–18 for an example screen and Table 8–34 for a descriptions of the items on this screen.

Figure 8–18. Disturbance Torque Screen

```
STATUS Axis
                                GRP[ 1 ]
     Disturbance Torque (A)
     Curr./ Max.(Allowed)/Min.(Allowed)
J1 :   0.0    0.0(  28.9)    0.0( -28.9)
J2 :   0.0    0.0(  50.0)    0.0( -50.0)
J3 :   0.0    0.0(  49.4)    0.0( -49.4)
J4 :   0.0    0.0(  24.1)    0.0( -24.1)
J5 :   0.0    0.0(  15.3)    0.0( -15.3)
J6 :   0.0    0.0(  14.6)    0.0( -14.6)
```

Table 8–34. Disturbance Torque Screen Items

ITEM	DESCRIPTION
Current units: Amps	This item is the current disturbance torque to the servomotor.
Maximum units: Amps	This item is the maximum value of the disturbance torque.
Minimum units: Amps	This item is the minimum value of the disturbance torque.

8.19.8 Servo Diagnosis

The Servo Diagnosis screens provide diagnostic information you can use to troubleshoot problems with your robot and controller.

Main Screen

The Servo Diagnosis Main screen displays the values of each of the diagnostic items for the robot axis that has the most out of range values. From this main screen, you can display more detailed, individual screens for each of the diagnostic items: reducer, motor overheat, transformer overheat, current, disturbance, OVC, collision detection, and discharge.

Figure 8–19. Servo Diagnosis Main Screen

```
Diagnosis

 group [1]
 reducer             1185352.5 hours
 over heat(motor)        61.17 %
 over heat(trans)         0.00
 current                  0.05
 disturbance              0.00 %
 OVC                      0.00 %
 collision detection     ***** times
 discharge                   0 W
```

Table 8–35. Servo Diagnosis Main Screen Items

ITEM	DESCRIPTION
Group	This item displays the number of the motion group for the axes being displayed.
Reducer	This item displays the time remaining until the recommended overhaul of the reducers.
Over Heat (Motor)	This item displays the ratio of the root mean square current to the rated current for the motor.
Over Heat (Trans)	This item displays the ratio of the root mean square current to the rated current for the transformer.
Current	This item displays the ratio of the current torque to the maximum.
Disturbance	This item displays the ratio of the force observed by the servo software to the alarm threshold.
OVC	This item displays the ratio of the motor temperature simulated by the software to the alarm threshold.
Collision Detection	This item displays the number of collisions that have been detected and information from the last detected collision.
Discharge	This item displays the discharge value for the servo amplifier. Discharge is the power supplied from the motors to the amplifier. For example, when the robot swings downward, gravity supplies energy to the amplifier through the motors.

Reducer Screen

The Servo Diagnosis Reducer screen shows the percentage of time used and the estimated hours remaining until the recommended overhaul of the reducers for each axis in the selected motion group. If you want to increase the number of hours between overhauls, decrease the programmed speeds or override. Monitor the reducer time value after running for several days to determine the period between overhauls of the reducers.

This screen shows the estimated hours remaining until the next reducer overhaul. The overhaul time depends on the future motion of the reducer. The overhaul time is based on a prediction made by averaging the load on the reducer from robot motion during the time period displayed on the screen. As the motion load changes, the overhaul time will change.

The estimated result is calculated by using the average rotational speed and torque, and is defined as the time when failure probability of large quantities of reducers accumulates to 10%.

The life is a statistical value that distributes in big dispersion. The real life of each reducer is different. It is normal that the real life is much longer or shorter than the estimated result. Please treat this result as a reference.

If the following incidents occur, the real life will be shorter than the estimated result:

- Collision with Robot arm or EOAT
- Emergency stop is frequently used
- Grease is not appropriately exchanged

According to the status of the reducer, the following maintenance measures should be adopted:

- Increase frequency of checks
- Prepare to exchange reducer
- Exchange reducer

After the exchange or overhaul of the reducer, reset the parameters to start the new calculation.

To display the Servo Diagnosis Reducer screen, press F3, reducer. See Figure 8–20 .

Figure 8–20. Servo Diagnosis Reducer Screen

```
Diagnosis/reducer
 group [1]     used   until 100%(estmtd)
        J1     50%      76863.3 hours
        J2     17%      57686.3 hours
        J3     18%      85768.6 hours
        J4      5%      93217.6 hours
        J5     40%      76876.8 hours
        J6      3%      65337.8 hours
 100%(estimated) means L10 life achieved
Estimation is based on the duty of:
   2008/08/30 19:45 — 2008/09/10 21:45
```

Table 8–36. Diagnosis Reducer Screen Items

ITEM	DESCRIPTION
used	This is the percentage already used of the recommended time until the next reducer overhaul.
until 100% (estmtd)	This shows the recommended number of hours remaining until the next reducer overhaul. The overhaul time depends on the future motion of the reducer, based on a prediction made by averaging the load on the reducer from robot motion. As the motion load changes, the overhaul time will change. If you want to increase the number of hours between overhauls, decrease the programmed speeds or override. Monitor the reducer time value after running for several days to determine the period between overhauls of the reducers.

To reset the parameters **after a reducer overhaul**, do the following

1. After the robot is stopped, press the EMERGENCY STOP button. The reducer diagnosis calculation will pause.
2. From the System Variable screen, set $FMR_GRP[reducer exchanged group].$T_LIFE[reducer exchanged axis]=0

   This resets the reducer diagnosis parameters.



Caution

Do not set $FRM_GRP[other group].$T_LIFE[other axis].

3. Release the EMERGENCY STOP button. The reducer diagnosis calculation will restart.

Over Heat Screen

The Servo Diagnosis Over Heat screen shows the ratio of the root mean square current to the rated current. If you want to decrease this value, decrease the override value or reduce program speed.

To display the Servo Diagnosis Over Heat screen, press F4, ov.heat. See Figure 8–21 .

Figure 8–21. Servo Diagnosis Over Heat Screen

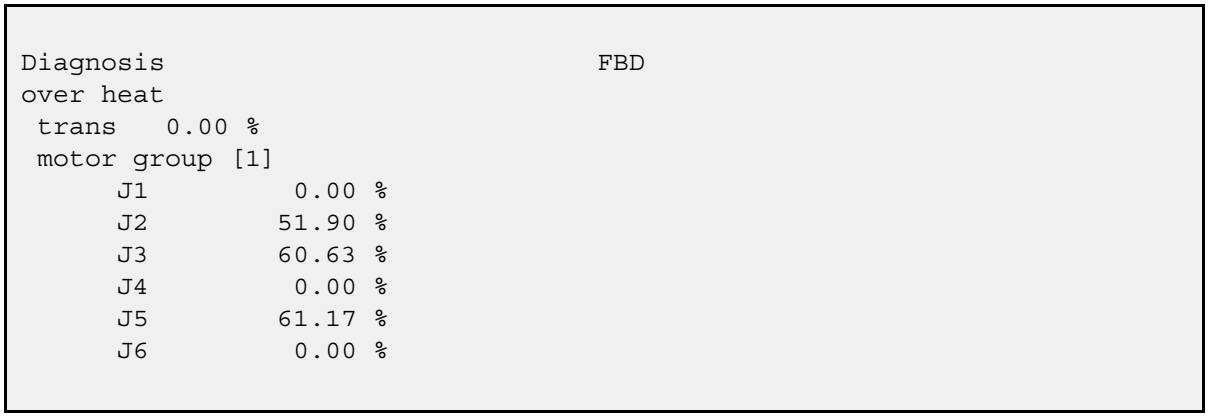


Table 8–37. Diagnosis Over Heat Screen Items

ITEM	DESCRIPTION
trans	This item is the ratio of the root mean square current to the rated current for the transformer.
Current ratio Units: %	This item is the ratio of the root mean square current to the rated current for the motor on the selected axis. If you want to decrease this value, decrease the override value or reduce program speed.

Torque Screen

The Servo Diagnosis Torque screen shows the ratio of the current torque to the maximum torque allowed. If this number is greater than 100%, confirm that the combined weight of the end-of-arm-tooling and workpiece does not exceed the maximum payload of the robot.

To display the Servo Diagnosis Torque screen, press NEXT, >, and then press F1, torque. See Figure 8–22 .

Figure 8–22. Servo Diagnosis Torque Screen

```
Diagnosis                          FBD
 torque
 group [1]
      J1    0.01 %
      J2    0.00 %
      J3    0.01 %
      J4    0.00 %
      J5    0.02 %
      J6    0.00 %
```

Table 8–38. Diagnosis Torque Screen Items

ITEM	DESCRIPTION
Torque ratio Units: %	This item is the ratio of the current torque to the maximum torque allowed. If this number is greater than 100%, confirm that the combined weight of the end-of-arm-tooling and workpiece does not exceed the maximum payload of the robot.

Disturbance Screen

The Servo Diagnosis Disturbance screen shows the ratio of the force observed by the servo software to the alarm threshold. If the maximum or minimum value is greater than 100%, confirm that the payload has been set correctly.

To display the Servo Diagnosis Disturbance screen, press NEXT, >, and then press F2, disturb. See Figure 8–23 .

Figure 8–23. Servo Diagnosis Disturbance Screen

```
Diagnosis                          FBD
 disturbance
 group [1] current   max(%)   min(%)
      J1    0.00 %    0.29 /    -2.41
      J2    0.00 %    2.92 /   -13.28
      J3    0.00 %   14.03 /    -6.36
      J4    0.00 %    1.57 /    -6.54
      J5    0.00 %    0.23 /   -46.59
      J6    0.00 %    6.80 /    -2.72
```

Table 8–39. Diagnosis Disturbance Screen Items

ITEM	DESCRIPTION
Current Units: %	This item is the current value of the ratio of the force observed by the servo software to the alarm threshold.
Max (%) Units: %	This item is the maximum value of the ratio of the force observed by the servo software to the alarm threshold. If the maximum value is greater than 100%, confirm that the payload has been set correctly and correct the setting if necessary.
Min (%) Units: %	This item is the minimum value of the ratio of the force observed by the servo software to the alarm threshold. If the minimum value is greater than 100%, confirm that the payload has been set correctly and correct the setting if necessary.

OVC Screen

The Servo Diagnosis OVC screen shows the ratio of the temperature simulated by the software to the alarm threshold. The temperatures of the motors are simulated by the software to protected them from overheating. If this value is 100 or greater, decrease the value by using the override command to decrease the override.

To display the Servo Diagnosis OVC screen, press NEXT, >, and then press F3, ovc. See Figure 8–24 .

Figure 8–24. Servo Diagnosis OVC Screen

```
Diagnosis
OVC
 group [1]
      J1    0.00 %
      J2    0.00 %
      J3    0.00 %
      J4    0.00 %
      J5    0.00 %
      J6    0.00 %
```

Table 8–40. Diagnosis OVC Screen Items

ITEM	DESCRIPTION
Ratio Units: %	This item is the ratio of the temperature simulated by the software to the alarm threshold. The temperatures of the motors are simulated by the software to protected them from overheating. If this value is 100 or greater, decrease the value by using the override command to decrease the override.

Last Detection Screen

The Servo Diagnosis Last Detection screen shows the number of collisions that have been detected and information from the last detected collision. If many collisions are detected, perform an overhaul to correct the problem.

To display the Servo Diagnosis Last Detection screen, press NEXT, >, and then press F4, cl.det.

Figure 8–25. Servo Diagnosis Collision Detection Screen

```
Diagnosis
 last detection
   2001/  7/ 18,  0: 52: 10
 group [1]
 count / position
 J1  ***** times    0.00 deg
 J2      1 times   66.13 deg
 J3  ***** times  -55.24 deg
 J4  ***** times   -0.00 deg
 J5      1 times   54.90 deg
 J6  ***** times    0.00 deg
```

Table 8–41. Diagnosis Last Detection Screen Items

ITEM	DESCRIPTION
Date and time year / month / day HH: MM: SS	This item is the date and time of the last detected collision.
Count	This item is the number of times a collision has occurred on the selected axis.
Position Units: degrees	This item is the position of the selected axis the last time a collision occurred.

8.19.9 Procedure

Use Procedure 8-17 to display the axis status screens.

Procedure 8-17 Displaying the Axis Status Screens

Steps

1. Press STATUS.
2. Press F1, [TYPE].

3. Select Axis.
4. Display the status screen you want:
   - **For Status 1**, press F2, STATUS1.
   - **For Status 2**, press F3, STATUS2.
   - **For Pulse**, press F4, PULSE.
   - **For Torque Monitor**, press NEXT, >, and then press F2, MONITOR.
   - **For Tracking**, press NEXT, >, and then press F3, TRACKING.
   - **For Disturbance Torque**, press NEXT, >, and then press F4, DISTURB.
   - **For Servo Diagnosis**, press NEXT, >, and then press DIAG.

   Note You cannot change any information on these screens except for the group number. Group number applies only if you have multiple groups; otherwise, it remains as 1.

5. **To change the group number**, press F5, GRP#, and type the number of the motion group you want.

8.20 PROGRAM MONITOR STATUS

The program monitor is started by a program monitor instruction and stops monitoring when the program executes a MONITOR END instruction or is aborted. You can display the Program monitor screen for the status of running or paused program conditions. Table 8–42 lists and describes each program monitor status item.

Note The Program monitor screen does not display a condition that has not been started.

Refer to Section 10.11 for more information on condition monitoring.

Table 8–42. Program Monitor Screen Items

ITEM	DESCRIPTION
CH Prog.	This item is the name of active condition handler program.
Status	This item is the status of condition. • Running : The monitoring of this condition is enabled. • Paused : The monitoring of this condition is disabled.

Table 8–42. Program Monitor Screen Items (Cont'd)

ITEM	DESCRIPTION
Program	This item is the name of the program that starts the condition. If the sub program starts the monitor, the main program name is displayed.
SYSTEM	This item displays the System Monitor screen. **Note** If $TPP_MON.$global_mt equal to 0, then this function key does not work and display the message "System monitor is not available."
RESTART	This item restarts the paused condition.
PAUSE	This item pauses the program condition.
END	This item ends this condition. The status is canceled and the condition is stopped.

Use Procedure 8-18 to display the program monitor status screen.

Procedure 8-18 Displaying Program Monitoring History

Steps

1. Press STATUS.
2. Press F1, [TYPE].
3. Select Condition. See the following screen for an example.

```
Program monitor
      CH Prog.  Status   Program
  1  WORK_DRP  Running  SAMPLE
  2  HAND_CHK  Paused   SAMPLE
  3  HAND_CHK  Paused   SAMPLE2
```

Note If you do not see this screen, press F2, PROGRAM.

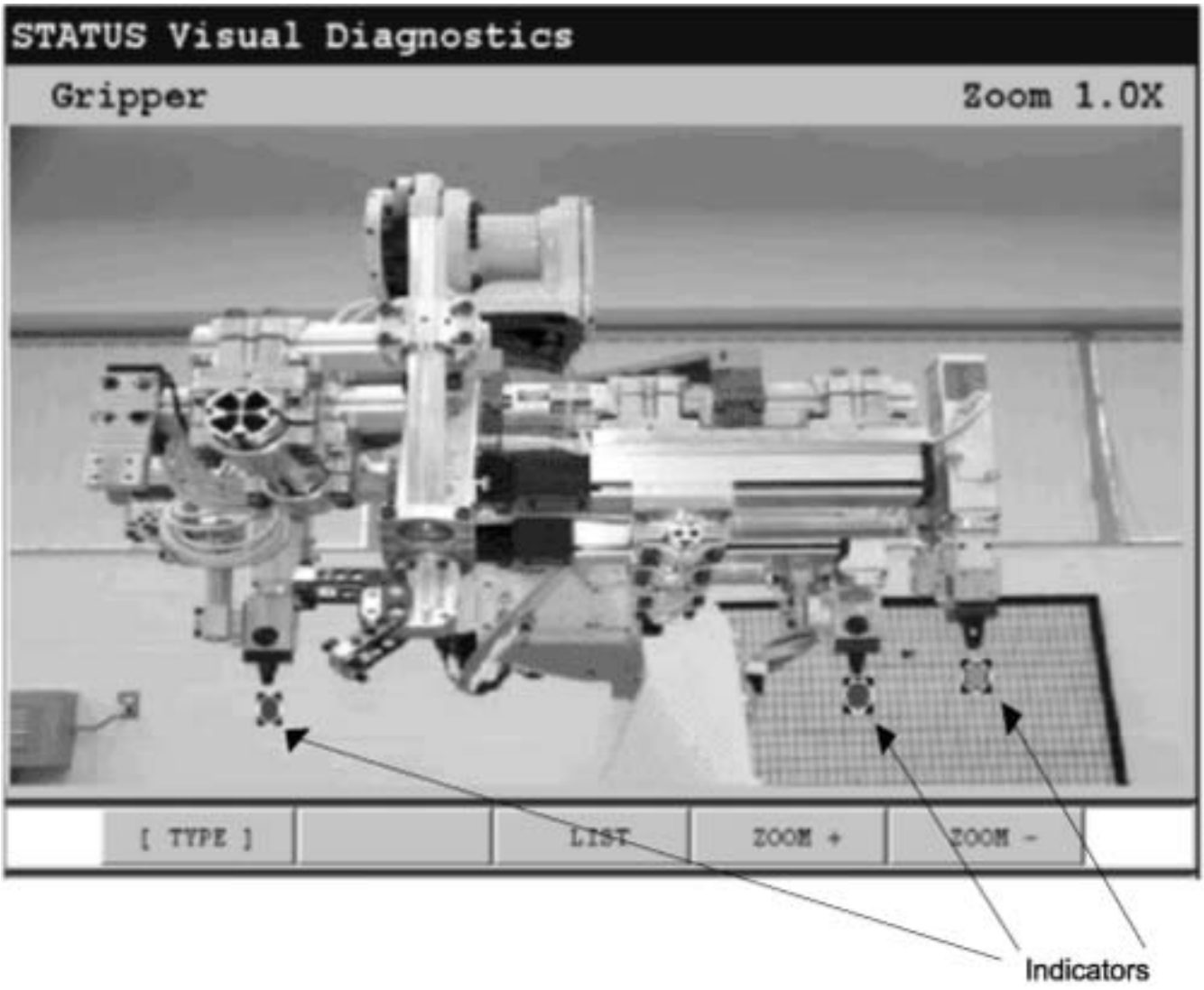
8.21 VISUAL DIAGNOSTICS

8.21.1 Overview

Visual Diagnostics uses the *i*Pendant animated picture display capability to provide a rich visual presentation of status information. A Visual Diagnostic (VD) screen is one that displays a reference

image as the background, and small indicator images strategically placed and activated in the foreground. The appearance of the indicator images is driven by the state of information within the controller. This can be useful, for example, in an application in which the robot is welding. Each weld position could be monitored and visually represented so that the success or failure of a weld is reported by green and red indicators. See Figure 8–26 for another example.

Figure 8–26. Simple Gripper Example



The background is a picture of a gripper with several actuators. In the foreground near many of the actuators are status indicators which are pictures of an LED to indicate the status of the actuator. The LED is green when the actuator is open and red when it is closed.

The foreground indicators are not limited to pictures of red and green LEDs. In fact, each indicator can have multiple states, each with a unique picture to display the state graphically. For example, a

clamp indicator could use images of the clamp in the opened, closed and intermediate positions. If properly scaled and oriented, the clamp on the VD screen would appear to represent the configuration of the clamp accurately.

An Indicator is displayed on or near an associated detail in the background image to specify its status. An indicator can have up to four states and each state can be represented by a different image. A Visual Diagnostic screen can have as many indicators as you want. However, be careful when adding indicators because each takes controller resources in terms of space and time, as well as screen clutter.

An indicator can have up to four states. An indicator's state is defined by the value of its one or two state control elements at run-time. If only one state control element (SCE) is defined (ie: number 2 is marked "**") then it can be a Boolean I/O (0=False, 1=True), Numeric I/O, Data Register or *i*RVision Register.

When the state control element is a vision register VR[x], the model ID from view 1 is used. If, at run-time, a numeric state control element evaluates to a number other than 0 through 3, the base 4 modulus of the number is used. If both state control elements are used, then they must both be Boolean I/O types. The indicator state is then determined as shown in Table 8–43 .

Table 8–43. Indicator States

SCE2	SCE1	State
0	0	0
0	1	1
1	0	2
1	1	3

Each of the four indicator states has an associated image that is displayed. If there is no state image associated with the number evaluated, then no image is shown and the background image shows through. The State Image: fields enable you to set up this state-to-image relationship.

8.21.2 Creating and Editing Visual Diagnostic Screens

You can create custom Visual Diagnostic screens on the robot controller. Table 8–44 and Table 8–45 list items you can set when setting up a Visual Diagnostic screen. Use Procedure 8-19 to create and edit a Visual Diagnostic screen.

Procedure 8-19 Creating and Editing Visual Diagnostic Screens

Conditions

- You have graphic files accessible on the controller that you want to use as your background and indicator images. These files should be stored as a .gif or .jpg and you must know where they are located on the controller.

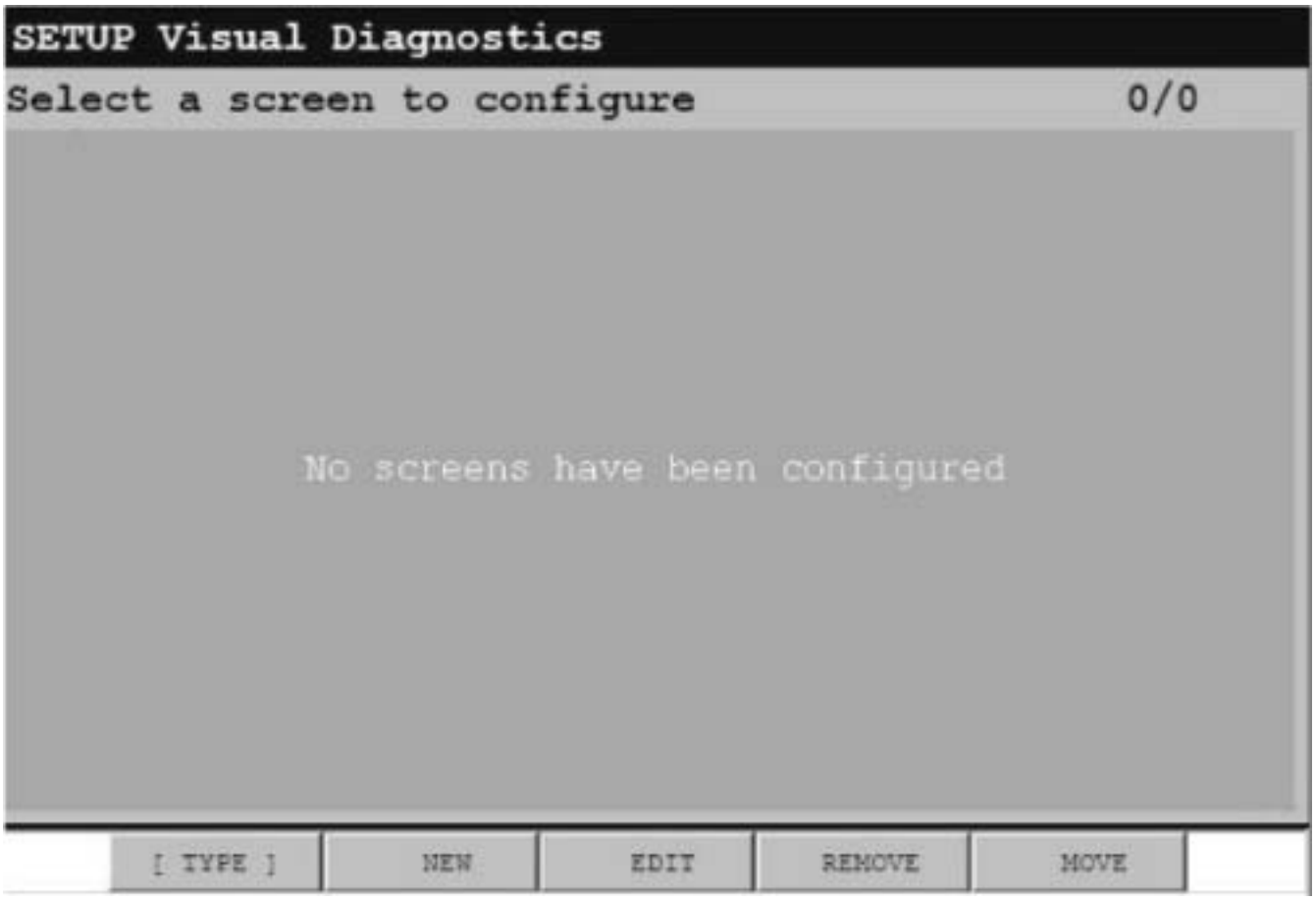
  The most common place to store these images is on FR:. If these are stored on MC:, then the memory card must be inserted to view the Visual Diagnostic screen properly.

  Note The screen area allocated for the background image is 630 pixels wide by 330 pixels high. For maximum performance, the background image should not be larger than this.

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Visual Diagnostics. If this is the first time you are creating a Visual Diagnostic screen, you will see a screen similar to the following.

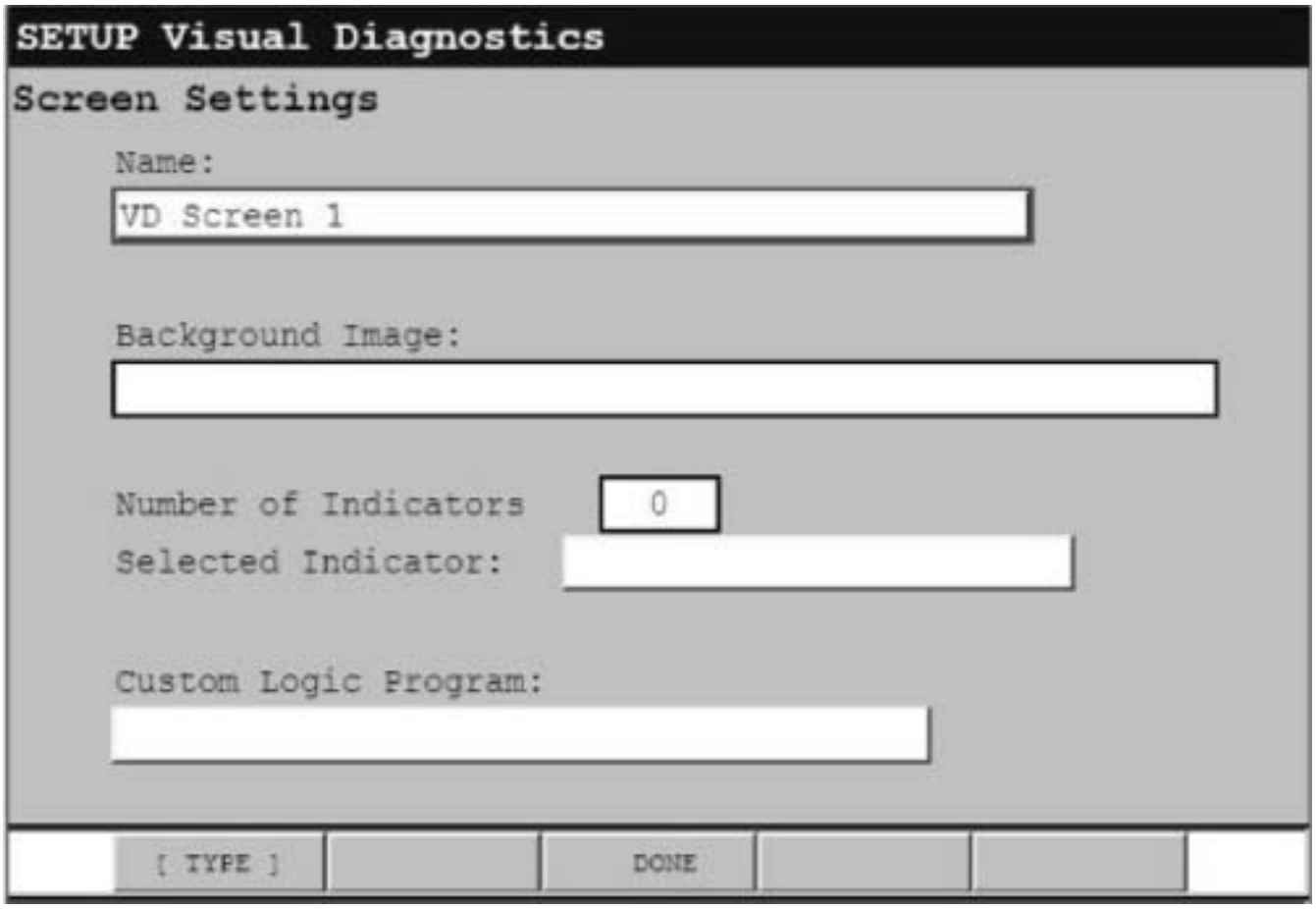
Figure 8–27. Select a Screen to Configure



Note If this is not the first time you have created a Visual Diagnostic screen, thumbnails of Visual Diagnostic screens currently configured on the controller are listed. To edit an existing screen, select the screen and press F3, EDIT, go to Step 6 .

5. **To create a new screen**, press F2, NEW. You will see a screen similar to the one shown in Figure 8–28 . Refer to Table 8–44 and Table 8–45 to set up the new screen.

Figure 8–28. Screen Settings Page



Note The red box on the screen indicates the selected field. You can select a field using the arrow keys. If you have a touch screen-enabled *i*Pendant or mouse, you can select a field by touching it. The function keys are adjusted to reflect operations that can be performed on the selected field. F3 is always DONE which, along with PREV, returns you to the previous SETUP screen.

Table 8–44. SETUP Visual Diagnostics Screen Settings Setup Items

ITEM	DESCRIPTION
Name	This item is the text that shows up on the SETUP and STATUS list screens so it must be unique. It can be from 1 to 40 characters and it can contain spaces. You can modify the name directly on the SETUP Visual Diagnostics Screen Settings screen.
Background Image	The large picture that covers most of the Visual Diagnostic screen is called the background image. It is stored as a file. When set, the file name is shown on the SETUP Visual Diagnostics Screen Settings screen.
Number of Indicators	This item specifies the number of indicators that have been set up. You cannot modify the number directly. Only the function keys are active when you are on this field.

Table 8–44. SETUP Visual Diagnostics Screen Settings Setup Items (Cont'd)

ITEM	DESCRIPTION
Selected Indicator	Only one indicator can be configured at a time. The Selected Indicator field specifies which one is currently selected.

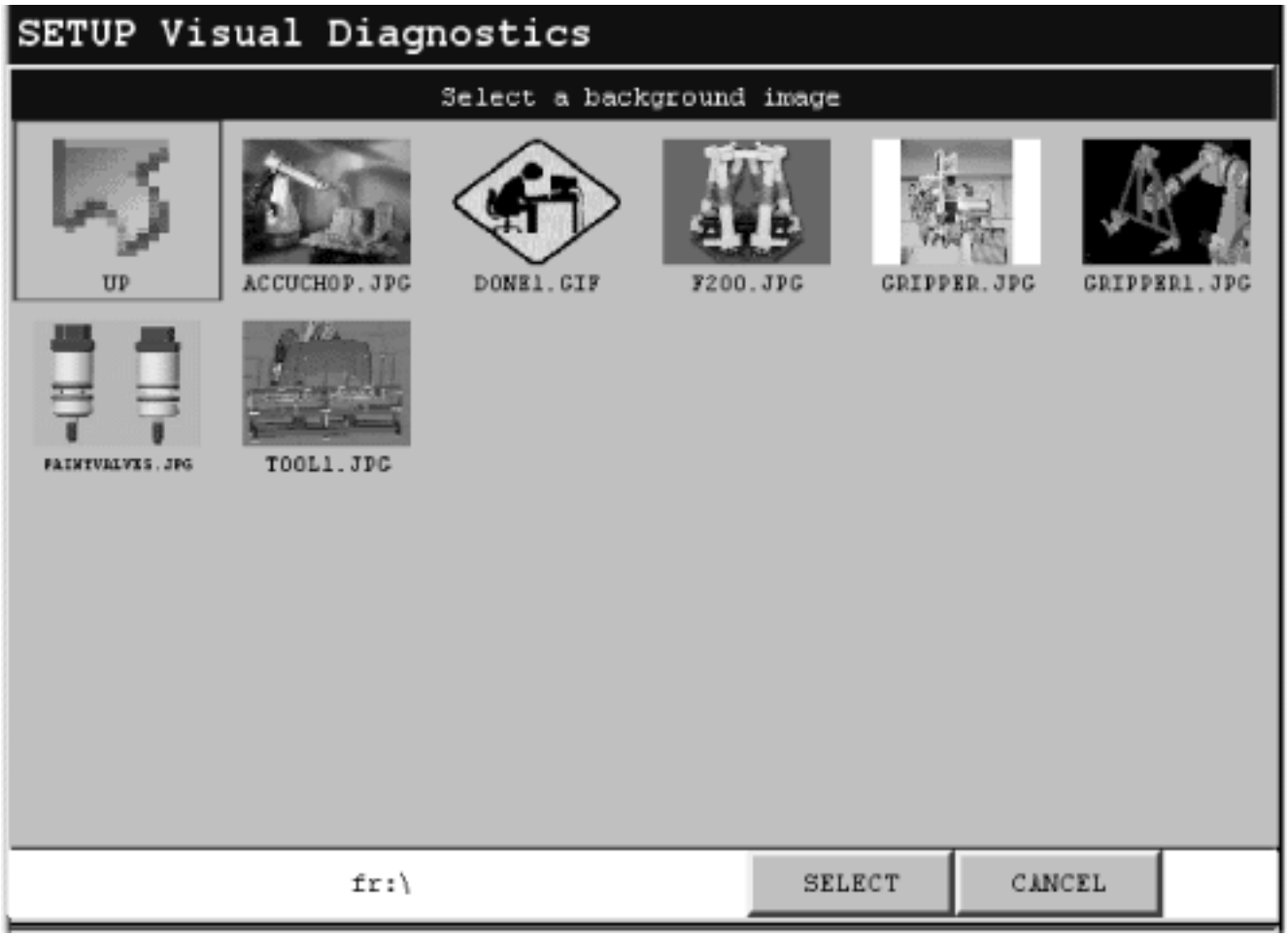
Custom Logic Program	You can identify a program that will run each time the Visual Diagnostic screen is refreshed. Use this feature if complex logic is required to determine the state of an indicator.
	
	A typical custom logic program will combine the status of various elements in the controller and set the indicators state control element accordingly. A custom logic program can be implemented in either TPP or KAREL. It must not contain any motion statements.
	
	Use the following guidelines to create a custom logic program:
	
	• The program must exist before setting this field but you can modify it later to change the logic. The most recent version of the program is run each time the screen is updated.
	
	• If it is a teach pendant program, the attributes must be set as follows:
	
	`BUSY_LAMP_OFF = 1,`
	`ABORT_REQUEST = 3,`
	`PAUSE_REQUEST = 7;`
	`DEFAULT_GROUP = *,*,*,*,*;`
	
	• The program must run quickly and terminate. If the program takes more than 1 second to complete an warning alarm will be posted but Visual Diagnostics will proceed. One second is very very long. A typical program should complete in less than a millisecond.

6. **To change the VD screen name**,
   a. Select the Name field, and press ENTER.
   b. Type the name of the screen, and press OK.
7. **To select the background image**,

a. Select the Background Image field and press F4, [CHOICE].

See the screen shown in Figure 8–29 for an example.

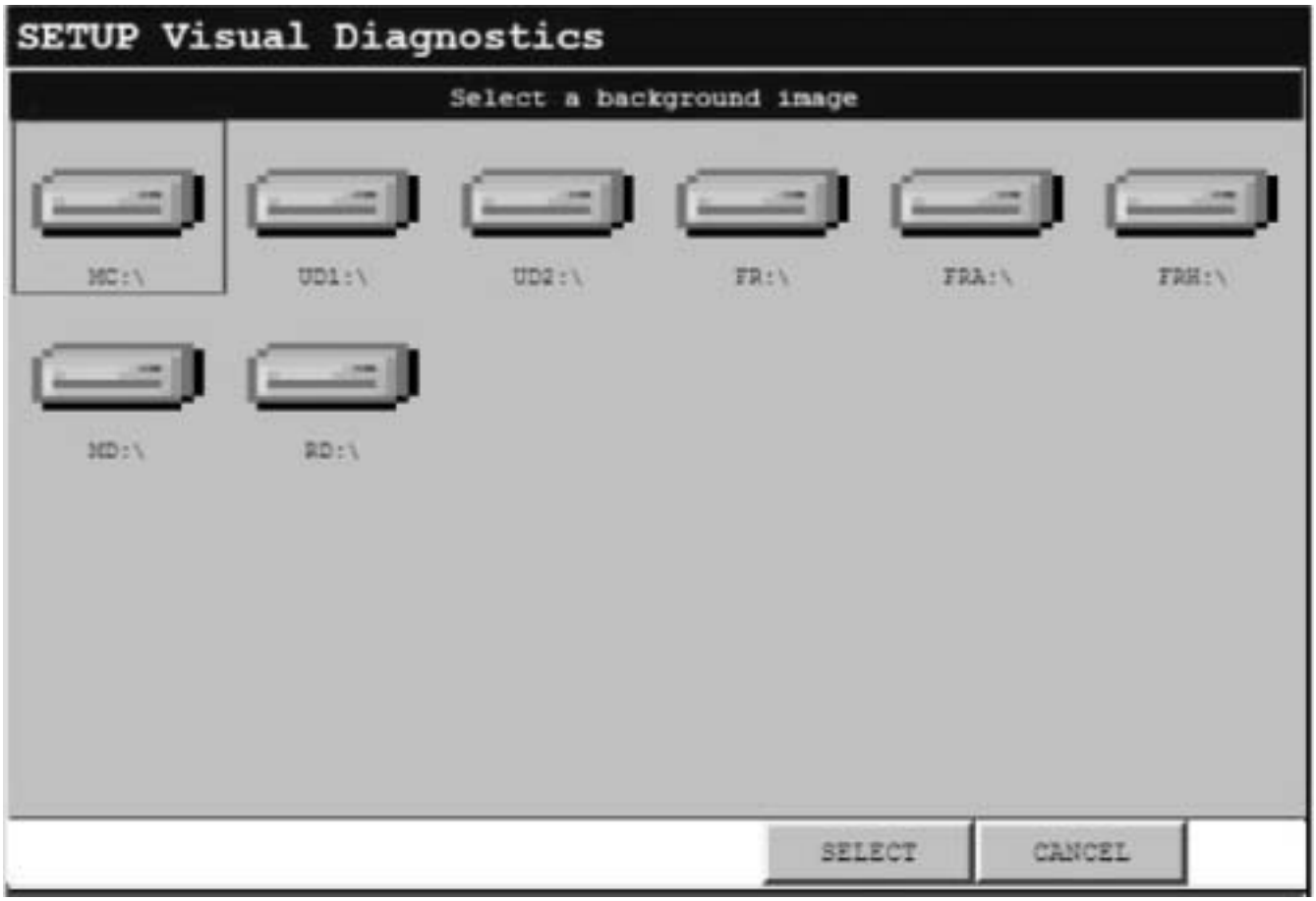
Figure 8–29. Selecting an Image File



b. The red box on the screen indicates the item with focus. Use the arrow keys to move the red box when more than one item is displayed. Press ENTER or F4 SELECT to select the item with focus.

If you have a touch screen *i*Pendant or mouse, clicking on a item gives it focus and selects it in one step. Selecting the green arrow displays the images and folders in the parent folder. The top most parent is a list of devices. See Figure 8–30 . Selecting a folder or a device displays the folders and image files within it. Selecting an image file returns that name to the field being edited in the Visual Diagnostics setup screen.

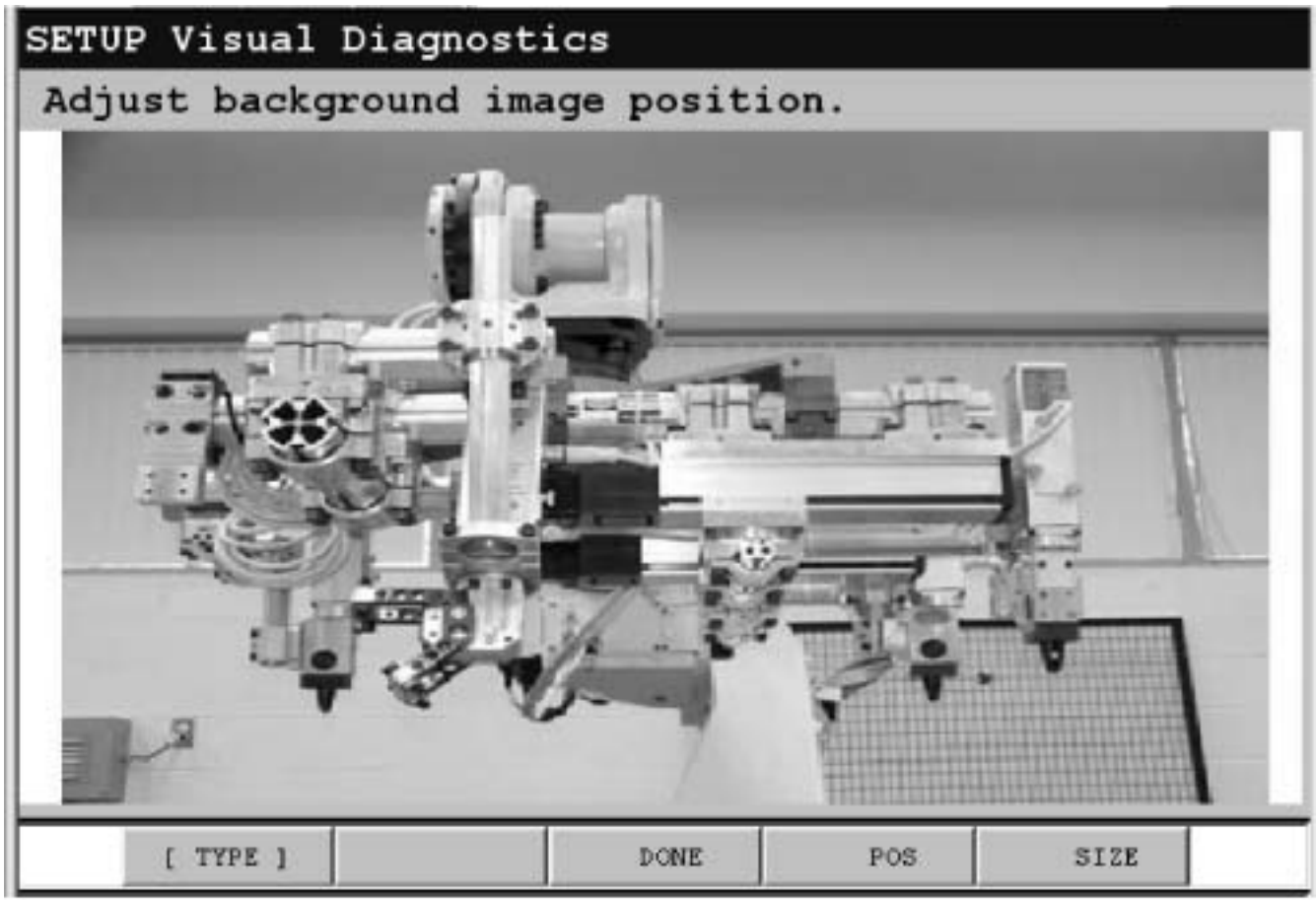
Figure 8–30. File Devices



8. **To adjust the background image**,

   a. Select the Background Image field and press F2, ADJUST. The selected graphic will appear in the center of the screen as shown in Figure 8–31 .

Figure 8–31. Adjust



b. F4, POS and F5, SIZE act like sticky radio buttons. Only one or the other is active at a time. When F4, POS is active, the arrow keys reposition the indicator image. When F5, SIZE is active, the up and down arrow keys grow and shrink the height of the image. The right and left arrow keys grow and shrink the width.

c. **To position or resize the graphic in larger increments**, press SHIFT and then use the arrow keys.

d. To complete the operation, press F3, DONE, PREV, or ENTER.

9. To add an indicator, select the Number of Indicators field and press F2, NEW. A new indicator with default settings will be created and you will immediately be placed into the edit mode as described below and shown in Figure 8–32

10. To edit an existing indicator,

a. Select the Selected Indicator field and press F2, EDIT.

b. Only one indicator can be edited at a time. If the name of the indicator you want to edit is not the one displayed, press F4, [CHOICE] and select it from the displayed list.

Indicator editing is done on a screen similar to the one shown in Figure 8–32 .

Refer to Table 8–45 for information on setting up items on the Indicator Settings screen.

Figure 8–32. Indicator Settings Page

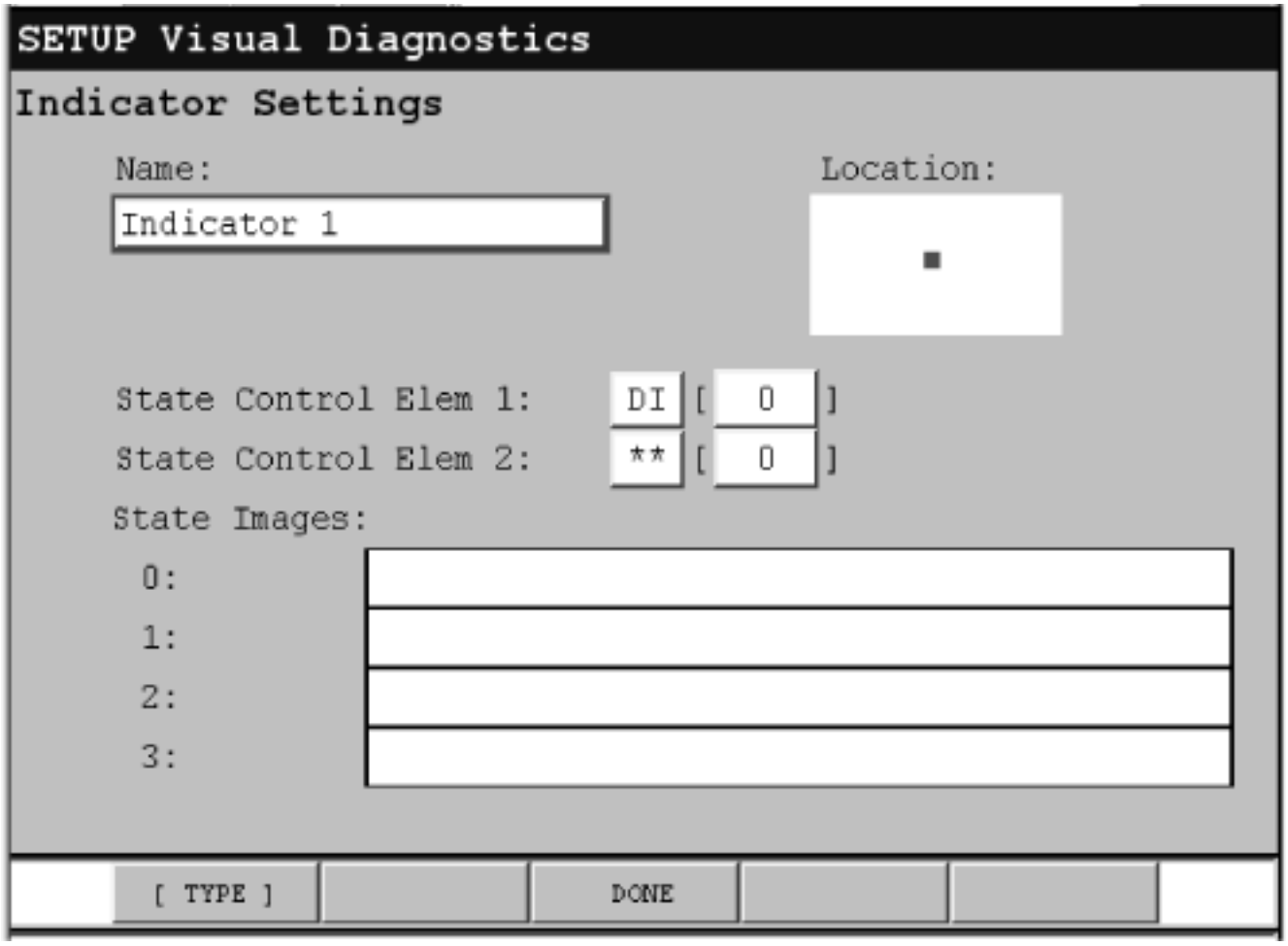


Table 8–45. SETUP Visual Diagnostics Indicator Setup Items

ITEM	DESCRIPTION
Name	This item specifies the name of the indicator. It provides you with a way to reference the indicator for setup. It is not displayed on the Visual Diagnostic STATUS screen.
Location	The Location field shows a graphic that provides a rough idea where the indicator is located.

Table 8–45. SETUP Visual Diagnostics Indicator Setup Items (Cont'd)

ITEM	DESCRIPTION
State Control Elem 1	This item specifies the first element that controls the indicator. It can be an I/O signal, numeric register, or vision register.
State Control Elem 2	This item specifies an optional second element to combine with the first one to control the indicator.
State Images	These are the images to display when the State Control Element(s) evaluate to the corresponding number.

11. To name the indicator,

a. Select the Name field, and press ENTER.

b. Type in the new indicator name and press OK.

12. To adjust the location of the indicator on the graphic,

a. Select the Location field and press F2, ADJUST.

The background image will be displayed with a small red box placed on it showing where the indicator is located. See the figure below.

b. Use the arrow keys to move the small red square to the location you want. If you have a touch screen or mouse pointer, you can click on a new location. Press F3, DONE, ENTER, or PREV to complete the operation.

13. To change the State Control Element type,

a. Select the left field, and press F4.

Note Each State Control Element line has two fields. The type field is displayed as one or two characters to the left of the left bracket. The index field is between the brackets.

Valid types for this field are:

- Boolean I/O - DI, DO, F, RI, RO, SI, SO, UI, UO, WI, WO
- Numeric I/O – AI, AO, GI, GO
- Data Register – R
- *i*RVision Register – VR
- Not used - ** (Only valid for SCE 2)

b. Select a type, and press ENTER.

14. To change the State Control Element index,

a. Select the right field and press ENTER.

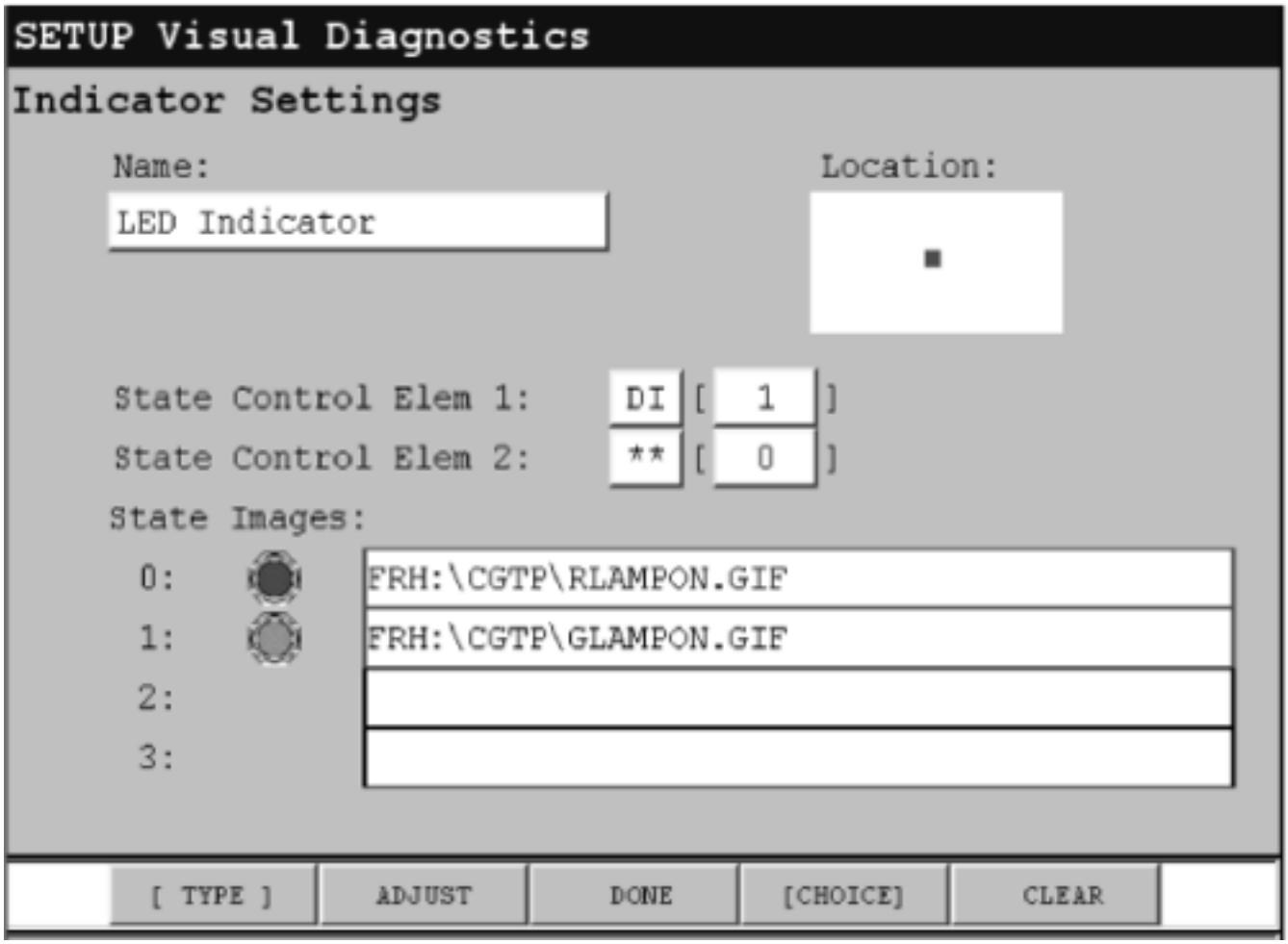
b. Enter the index number directly. No validation is done at this time. An invalid type or index is detected only when the Visual Diagnostic screen is viewed in the STATUS menu (refer to Section 8.21.4).

15. To associate an image to a state,

a. Select the image name field to the right of the desired state number and press F4, [CHOICE].

b. Use the file thumbnail browser to select the image. See Figure 8–33 for an example. Figure 8–33 shows the associated images to states 0 and 1.

Figure 8–33. Indicator Settings

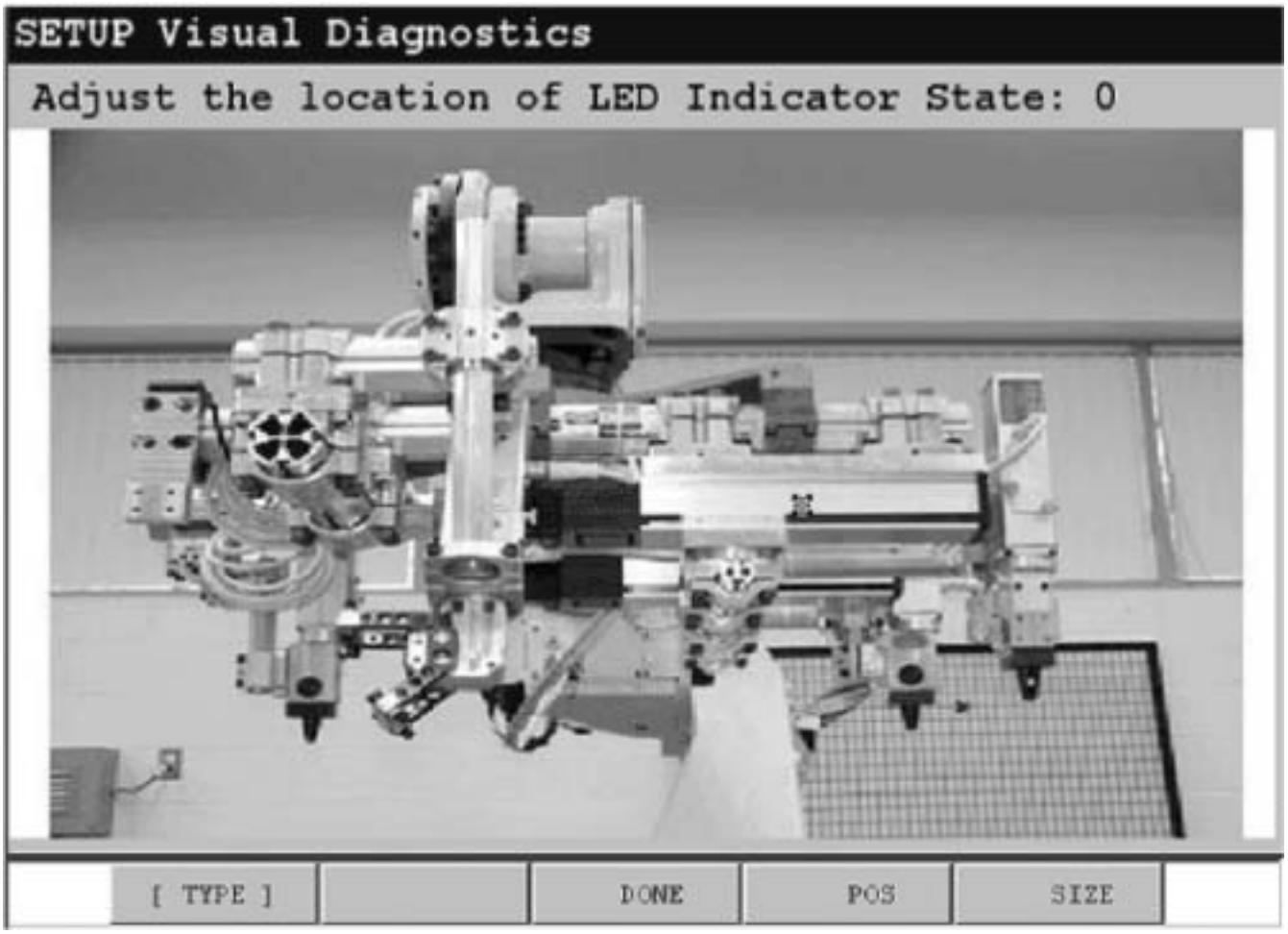


16. To make fine position and scale adjustments to the state image,

a. Select the State Image field and Press F2, ADJUST.

b. The background image with only the currently selected indicator state image overlaid on top will be displayed. See Figure 8–34 .

Figure 8–34. Adjust the Location of LED Indicator State: 0



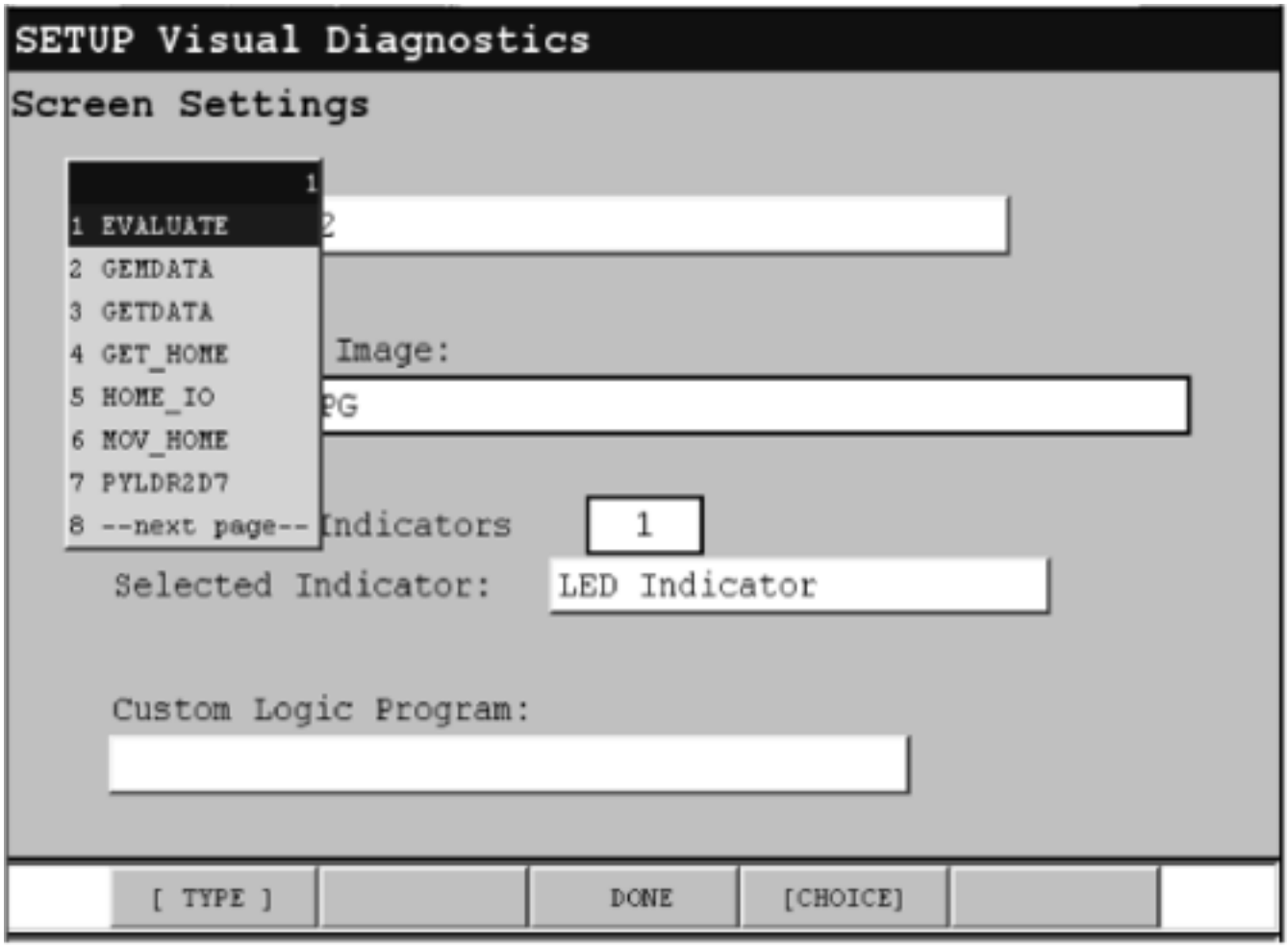
Note The position adjustment is relative to the grid location chosen for the indicator. These adjustments apply to the specific indicator state on the specific Visual Diagnostic screen being set up. They do not alter the indicator image file itself.

c. F4, POS and F5, SIZE act like sticky radio buttons. Only one or the other is active at a time. When F4, POS is active, the arrow keys reposition the indicator image. When F5, SIZE is active, the up and down arrow keys grow and shrink the height of the indicator image. The right and left arrow keys grow and shrink the width. To position or resize the graphic in larger increments, press SHIFT and then use the arrow keys.

d. Press F3, DONE, PREV, or ENTER to complete the operation.

17. When you are finished setting up the indicators on the screen, press F3, DONE or PREV. This will return you to the Screen Settings page on which the following procedures are performed.
18. **To remove the selected indicator from a screen**,
    a. Select the Selected Indicator field and press F4, REMOVE.
    b. A confirmation box will be displayed so that you are aware that the indicator will be permanently removed by this function.
19. **To assign a custom logic program**,
    a. Select the Custom Logic Program field and press F4, [CHOICE]. This will display the list of programs currently loaded on the controller.
    b. Select the program you want from this list, and press ENTER. See the Figure 8–35 for an example.

Figure 8–35. Custom Logic Program Field



20. **To clear the Custom Logic Program assignment,** select the Custom Logic Program field and press F5, CLEAR.

8.21.3 Managing Screens

You can perform functions such as remove, move, copy, and load a new screen.

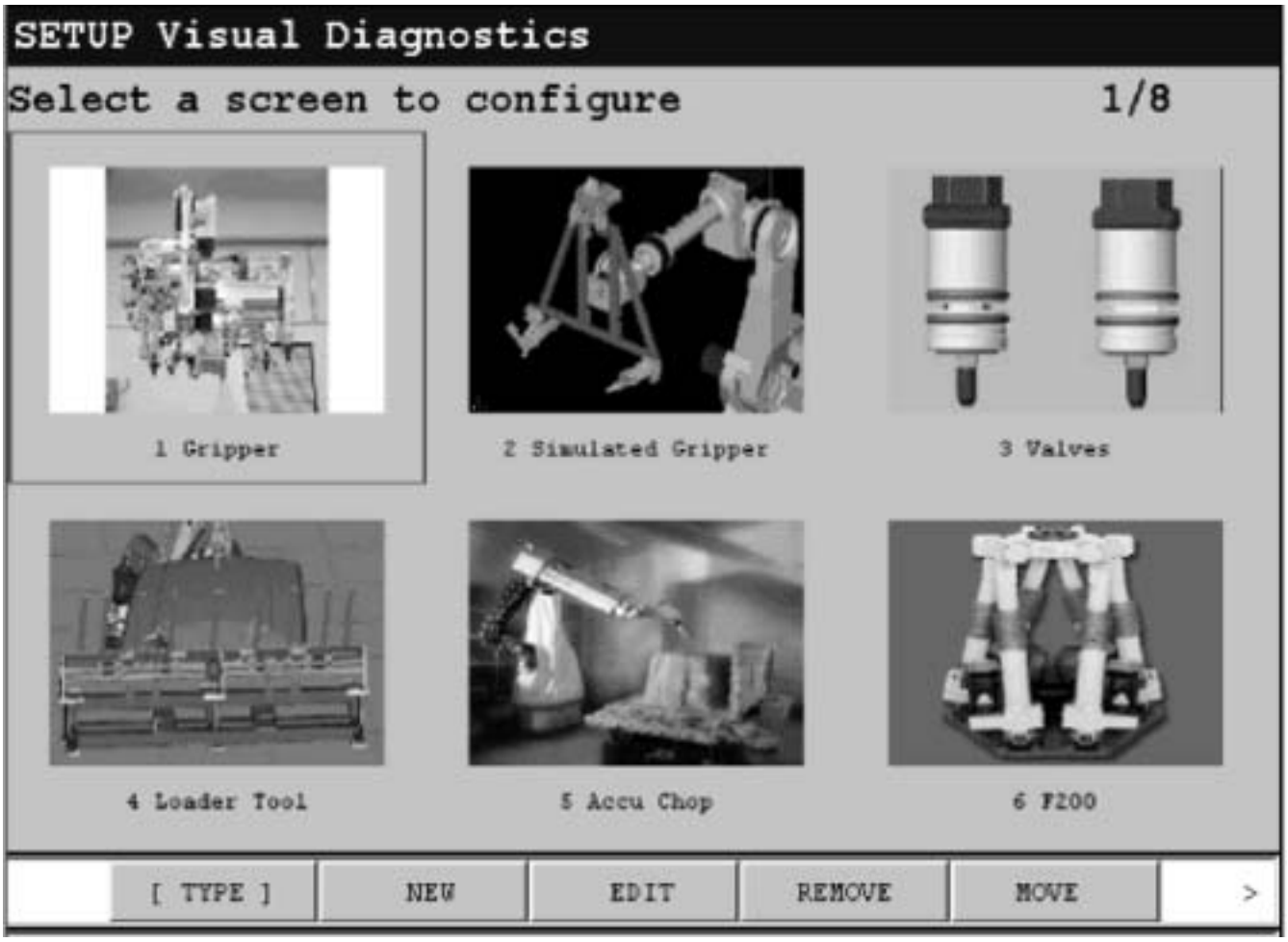
Note If the selected item is currently being displayed in any other *i*Pendant pane, the EDIT and REMOVE function keys will not let you operate on it.

Condition

- Visual Diagnostic screens have already been set up.

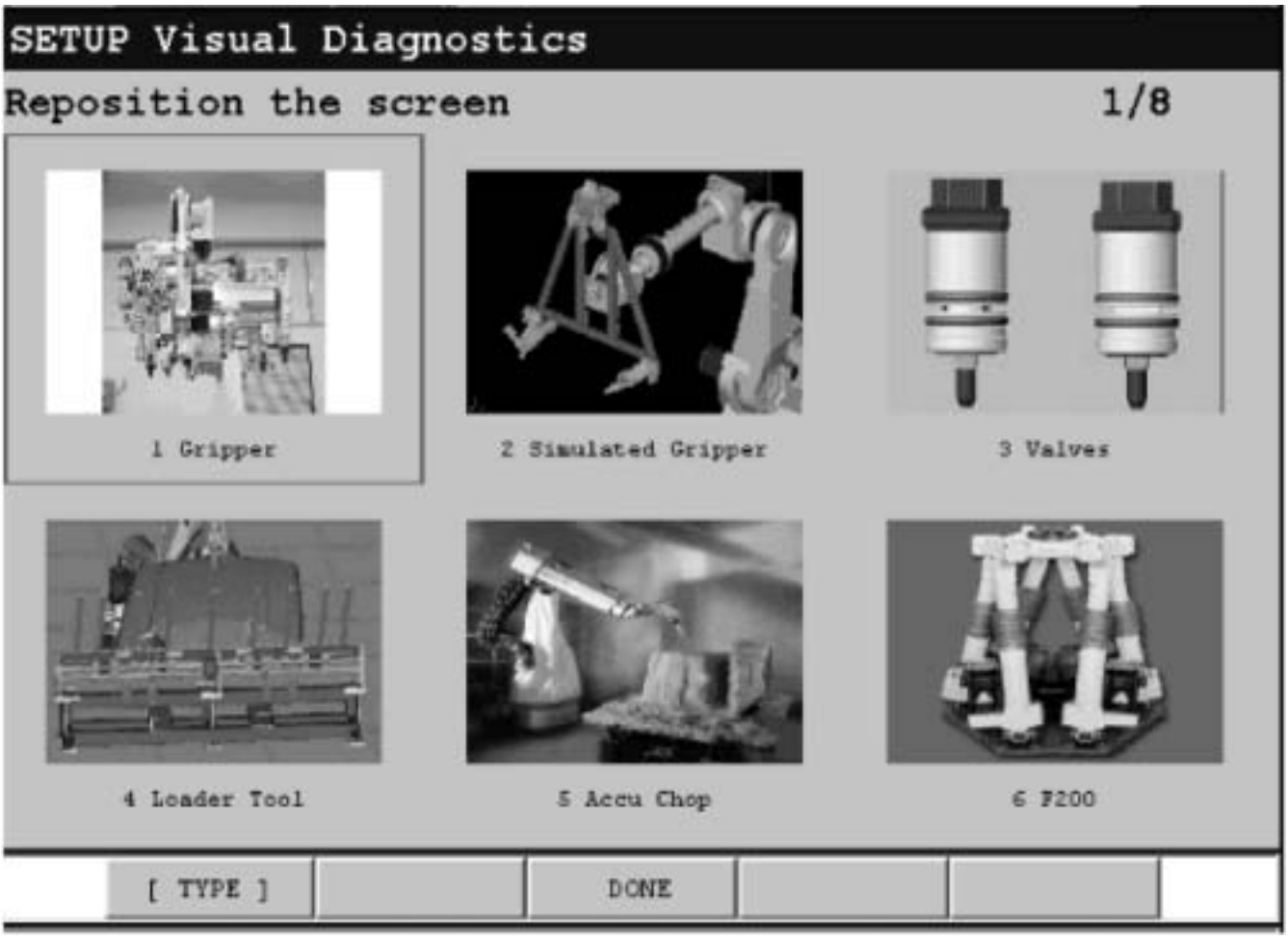
1. Press MENUS.
2. Select SETUP.
3. Select Visual Diagnostics. You will see a screen similar to the following.

Figure 8–36. SETUP Visual Diagnostics



4. **To remove a screen,** press F4, REMOVE. A confirmation box will be displayed so that you are aware that the screen will be permanently removed by this function.
5. **To reposition a screen,**
   a. Press F5, MOVE. You will see a screen similar to the following.

Figure 8–37. Reposition the Screen



b. Use the arrow keys to move the selected screen to its new position.

c. Press F3, DONE to finish the operation and return to the screen selection page.

6. **To make a new screen as an exact copy of the currently selected screen but with a different screen name**,

   a. Press >, NEXT, and then press F2, COPY. A new screen will be created and the Screen Settings page shown so that you can edit the new screen.

   b. Use Procedure 8-19 to edit the screen.

7. **To load Visual Diagnostic screens developed on another controller onto this controller**,

   a. Press >, NEXT and then press F3, LOAD .

   b. Using the file browser, find and select the file named “vdscreens.xml” on any file device. When selected, the system merges the screens defined in it with the screens that currently reside on the controller.

c. If a duplicate screen name is encountered, you are prompted with the option of replacing or keeping the one resident on the controller.

8.21.4 Viewing Visual Diagnostic Screens

Use Procedure 8-21 to view Visual Diagnostic Screens.

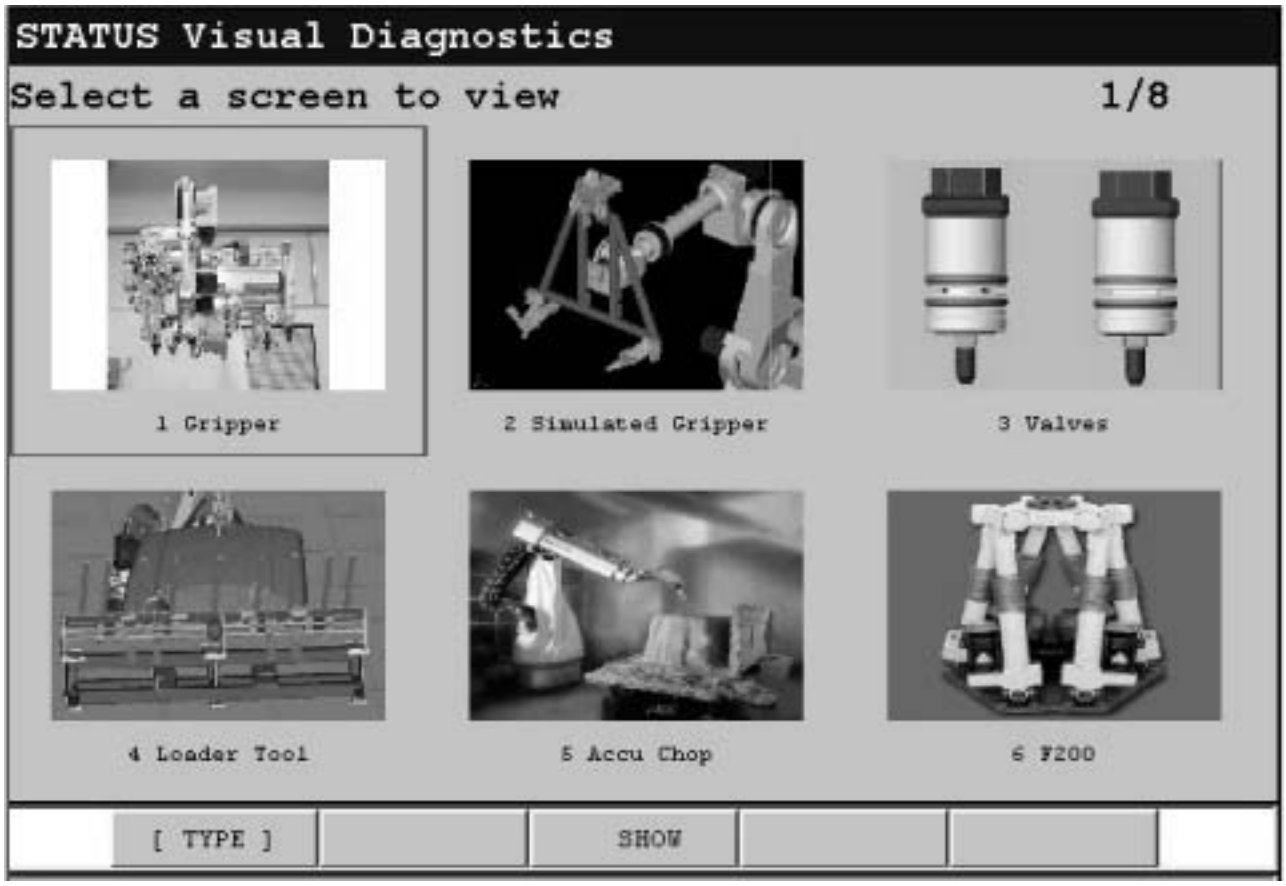
Procedure 8-21 Viewing Visual Diagnostic Screens

Condition

- The Visual Diagnostic screen you want has already been set up.

1. Press MENUS.
2. Select STATUS.
3. Press F1, [TYPE].
4. Select Visual Diagnostics. You will see a screen similar to the following.

Figure 8–38. Select a Screen to View



Note Use this menu item to display a Visual Diagnostic screen. Thumbnails of the set of screens configured on the controller are shown on the list selection screen.

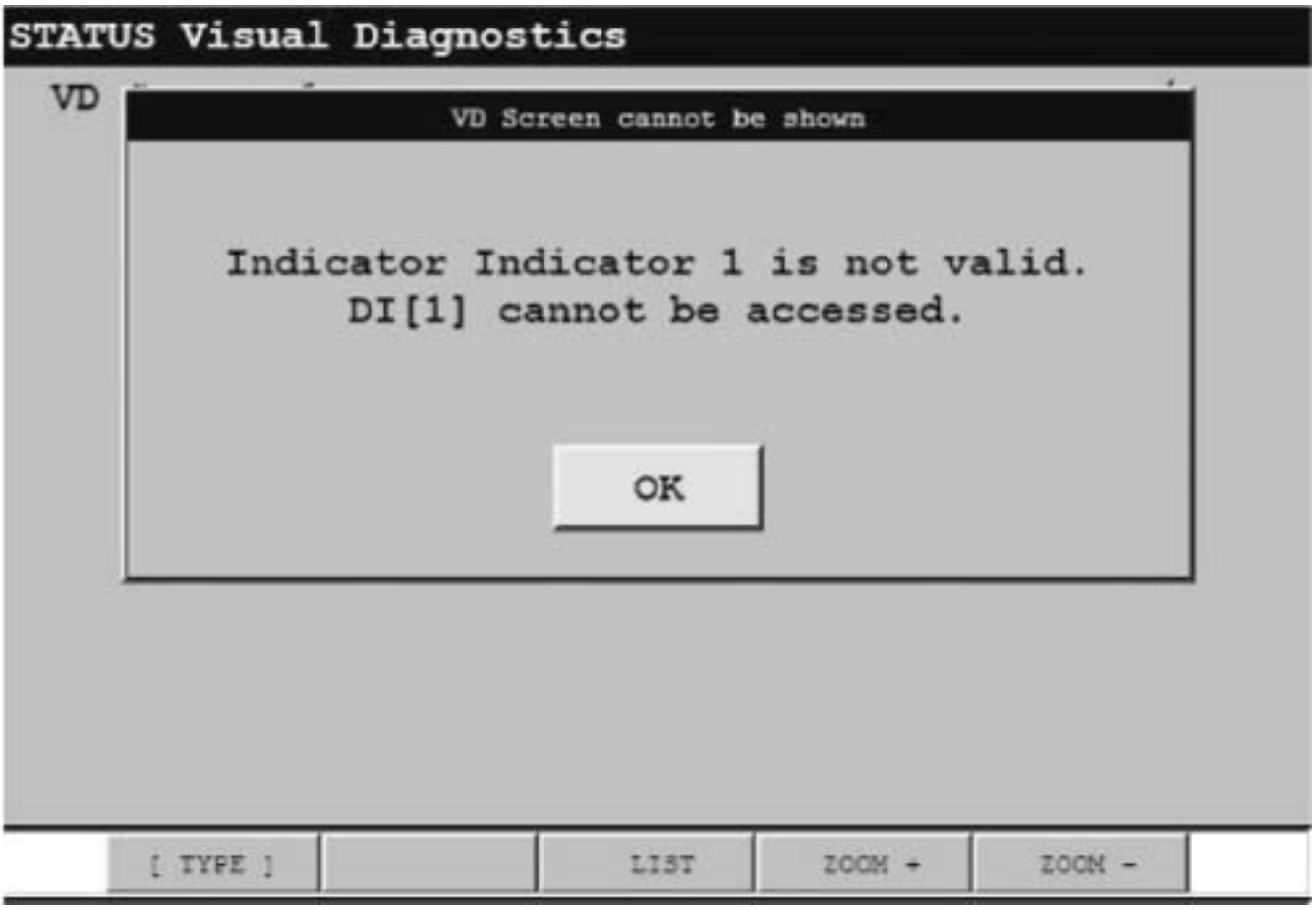
The red box on the screen indicates the selected screen. You can select a screen using the arrow keys. If you have a touch screen-enabled *i*Pendant or mouse, you can select a screen by touching it. The numbers in the upper right hand corner indicate How many screens are available and which is currently selected.

5. **To display a screen,** select it and press F3, SHOW.

Note If the selected item is currently being edited in any other *i*Pendant pane, F3, SHOW will not let you view it.

6. If the screen you have selected has an invalid signal set, you will see a screen similar to the following.

Figure 8–39. Invalid Indicator

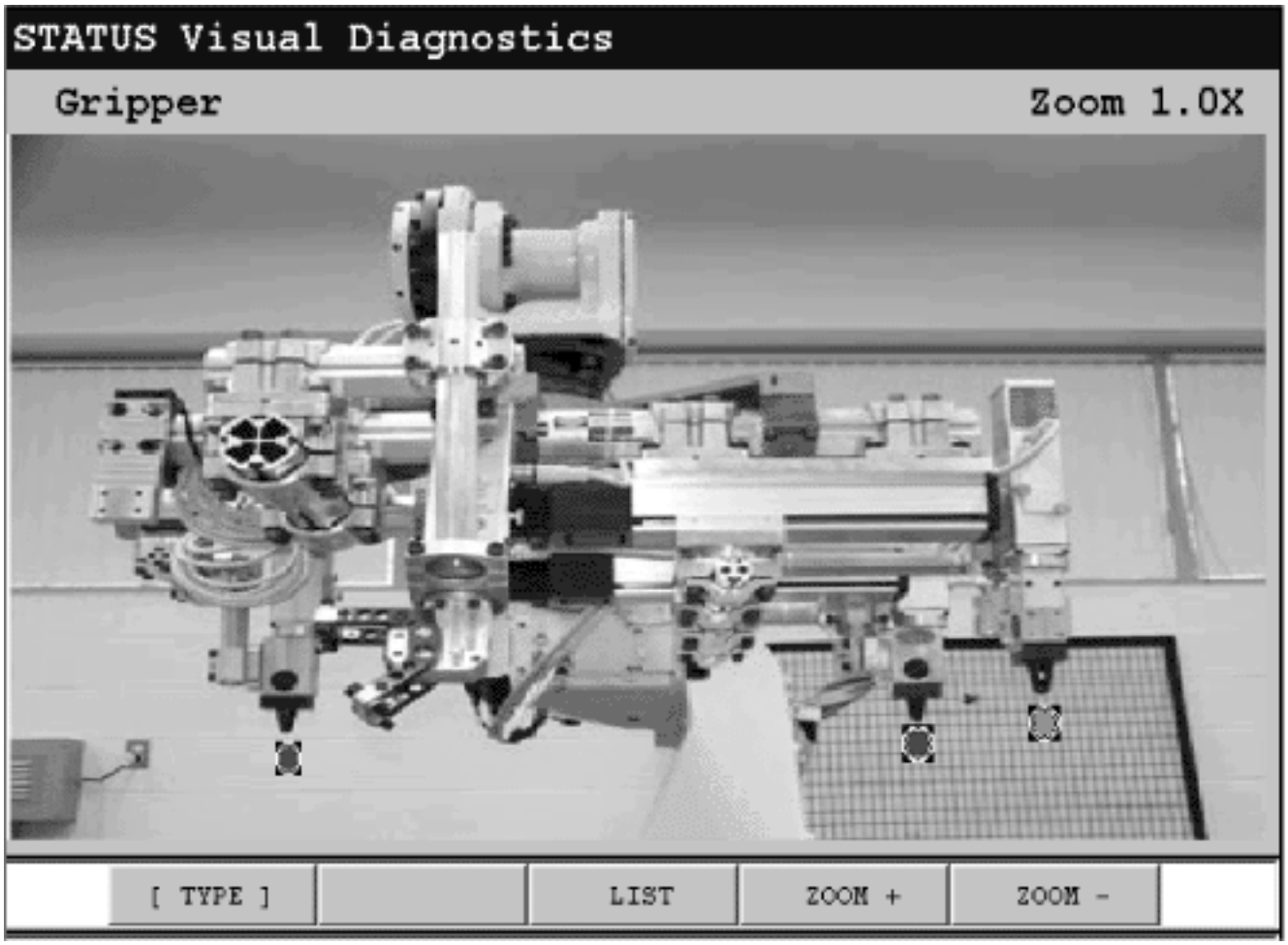


a. Press OK.

b. Go to the SETUP menu to redefine the signal for that indicator. See Procedure 8-19 .

7. After you select a Visual Diagnostic screen, the following screen is displayed.

Figure 8–40. Visual Diagnostics Screen Display



The top line displays the title of the screen. In this example, the title is Gripper.

Note Adjustments (since controller power up) to the magnification and picture center are remembered for each Visual Diagnostic screen. So if you leave the screen and then return, the view is restored.

8. **To increase magnification of the image,**
   a. Press F4, ZOOM+.
   b. The magnification is increased keeping the point of the image in the center of the display fixed in place.
9. **To decrease magnification of the image,**
   a. Press F5, ZOOM-. The point of the picture in the center of the display is kept in place until one of the edges of the image touches the edge of the display area. The image will then be shifted in that direction to keep the image edge touching the display edge.

b. Magnification will not be reduced below 100%.

c. F2, FULL, restores the image to full view.

10. **To adjust how the image is centered in the display when the magnification is greater than 100%,**

a. Press the right and left arrow keys to shift the view to the right and left.

b. Press the up and down arrow keys to shift the view up and down.

c. To move the view in larger increments, hold the shift key while pressing the arrow key.

11. **To display the Visual Diagnostic Screen list selection page,** press F3, LIST. This key has the same function as the PREV key.

Chapter 9

PROGRAM AND FILE MANIPULATION

Contents

Chapter 9	PROGRAM AND FILE MANIPULATION	9–1
9.1	STORAGE DEVICES	9–3
9.1.1	Overview	9–3
9.1.2	Storage Device Hints	9–9
9.1.3	Setting the Default Device	9–10
9.1.4	Setting Up a Port	9–12
9.1.5	Using a Memory Card or USB Memory Stick Interface	9–19
9.1.6	Formatting Devices	9–23
9.1.7	Subdirectories	9–26
9.2	MANIPULATING PROGRAMS	9–30
9.2.1	Overview	9–30
9.2.2	Selecting Programs On the SELECT Menu	9–31
9.2.3	Saving Programs	9–32
9.2.4	Loading Programs	9–34
9.2.5	Copying Programs Within the SELECT Menu	9–36
9.2.6	Deleting Programs from the SELECT Menu	9–37
9.2.7	Saving Files as ASCII	9–39
9.2.8	Filtering Programs in the Select List	9–41
9.3	MANIPULATING FILES	9–42
9.3.1	Overview	9–42
9.3.2	Altering the FILE Menu View	9–47
9.3.3	Generating a Directory of Files	9–50
9.3.4	Backing Up Files	9–52
9.3.5	Loading and Restoring Files to Controller Memory	9–58
9.3.6	Conversion of PC Files	9–64
9.3.7	Auto Software Update	9–65
9.3.8	Displaying Text (ASCII) Files	9–66
9.3.9	Copying Files	9–67
9.3.10	Deleting Files	9–70
9.3.11	Saving Files	9–71

9.3.12 Checking File Memory 9–74
9.4 AUTOMATIC BACKUP 9–75
9.4.1 Overview 9–75
9.4.2 Automatic Backup 9–75
9.5 TROUBLESHOOTING FILE BACKUP OR AUTOMATIC BACKUP ERRORS 9–80
9.6 IMAGE BACKUP 9–82
9.6.1 Overview 9–82
9.6.2 Performing an Image Backup 9–83
9.6.3 Restoring an Image Backup 9–86
9.7 ASCII UPLOAD 9–89
9.7.1 Overview 9–89
9.7.2 Loading an ASCII Teach Pendant Program From the Teach Pendant 9–90
9.7.3 Loading an ASCII Teach Pendant Program from KCL 9–93
9.7.4 Viewing ASCII Upload Errors 9–94
9.7.5 Example ASCII File 9–95

9.1 STORAGE DEVICES

9.1.1 Overview

The following kinds of storage devices can be used to store programs and files:

- Memory Card (MC:)
- Flash File Storage disk (FR:)
- RAM Disk (RD:) (Not for SpotTool+)
- Ethernet Device (optional)
- Memory Device (MD:)
- Memory Device Binary (MDB:)
- MF Device (MF:)
- Filtered Memory Device (FMD:)
- USB Memory Stick Device on the controller (UD1:)
- USB Memory Stick Device on the *i*Pendant (UT1:) (if you have an *i*Pendant with a USB port on it and you have the USB port on *i*Pendant option (J957) installed)

This section describes how to set up storage devices for use. Depending on the storage device, this can include

- Setting up a port on the controller
- Connecting the device to the controller
- Formatting a device

Memory Card (MC:)

The controller supports memory cards. Memory cards support various sizes 8MB or higher. Compact Flash PC cards are also supported if used with a suitable compact adapter. The memory card requires a memory card interface which is standard on Main CPU inside the controller.

Note The controller supports loading software from memory cards.

Warning

Lethal voltage is present in the controller WHENEVER IT IS CONNECTED to a power source. Be extremely careful to avoid electrical shock. HIGH VOLTAGE IS PRESENT at the input side whenever the controller is connected to a power source. Turning the disconnect or circuit breaker to the OFF position removes power from the output side of the device only.

Warning

The memory card interface is located on the Main CPU on the controller cabinet. When the power disconnect circuit breaker is OFF, power is still present inside the controller. Turn off the power disconnect circuit breaker before you insert a memory card into the memory card interface; otherwise, you could injure personnel.

Caution

Do not remove the memory card when the controller is reading or writing to it. Doing so could damage the card and lose all information stored on it.

The memory card can be formatted on the controller, and can be used as a load device to install software.

Note Data on all internal file devices such as FR:, RD:, and MD: should be backed up to external file device such as ATA Flash PC card.

Note The controller formats the card with a sector size of 512 bytes.

The memory card can be formatted and used as an MS-DOS file system. It can be read from and written to the controller and an IBM PC equipped with the proper hardware and software. If the memory card is used as an MS-DOS file system, it should be formatted on the controller.

The controller can read and write memory cards that are formatted with FAT (also referred to as FAT16) or FAT32 type of formatting (File Access Tables). When a memory card is formatted on the Controller it can be formatted as FAT or FAT32 type.

The FAT32 format (32 Bit FAT) removes a few limitations that are included with FAT. One of these is the limitation that only 512 files can be created in the Root directory. Another is that FAT format type only supports memory cards up to 2 GB in size. This feature is included in the controller to increase the compatibility of the robot controller with other computer systems.

USB Memory Stick Device on the controller (UD1:), and USB Memory Stick Device on the *i*Pendant (UT1:)

The controller supports USB Flash memory sticks up to 2GB in size if it is formatted as FAT16. FAT16 is much faster for access. USB Memory sticks with any security or encryption features are not supported. Memory sticks larger than 2GB can not be formatted FAT16, they must be formatted FAT32.

Note USB Memory sticks cannot be formatted when plugged into the USB connector on the iPendant (UT1:) They must be formatted on a PC or when plugged into the front panel USB connector on the controller.

Note USB Memory sticks with any security or encryption features are not supported.

Note Memory sticks larger than 2GB can not be formatted FAT16, they must be formatted FAT32.

Note Generally the larger the size of the device such as a USB memory stick, the slower the access speed and device performance. A USB memory stick should be formatted FAT16 if possible. Larger memory sticks formatted as FAT32 will work, but will be slower for file operations and startup of the memory stick when it is first inserted.

Note In order to use the UT: device on the *i*Pendant, you must have the USB Port on *i*Pendant option (J957) installed.

Note Since USB is a well established specification, FANUC Robotics America, Inc. does not qualify specific USB Memory sticks for use with the R-30*i*A controller. FANUC Robotics uses USB Flash drives manufactured by SanDisk® (CRUZER Mini and Micro) to qualify the operation of the USB interface. Therefore, FANUC Robotics recommends that you use SanDisk® USB Flash Drives, without security or encryption features. Other drives might work properly but are not specifically qualified by FANUC Robotics.

The USB Memory Stick Device requires a USB interface which is standard on the controller.

The USB memory stick device can be formatted on the controller.



Caution

Do not remove the memory stick when the controller is reading or writing to it. Doing so could damage the memory stick and lose all information stored on it.

Flash File Storage Disk (FR:)

Flash File Storage Disk is a portion of FROM memory that functions as a separate storage device. Flash file storage disk (FR:) does not require battery backup for information to be retained. You can store the following information on Flash file storage disk:

- Programs

- System variables
- Anything you can save as a file

You can format the Flash file storage disk. The size of the Flash file storage disk is set by the system at software installation. Due to the nature of FROM, each time you copy or save a file to the FR: there will be a drop in available FR: memory, even if you are working with the same file. Periodic purging will recover the lost space.

RAM Disk (Not for SpotTool+)

RAM Disk is a portion of Static RAM (SRAM) or DRAM memory that functions as a separate storage device. Any file can be stored on the RAM Disk. RAM Disk files should be copied to disks for permanent storage.

The location and size of the RAM disk (RD:) depends on the value of the system variable $FILE_MAXSEC. The default value of $FILE_MAXSEC depends on the options and tool packages that are installed.

The value in $FILE_MAXSEC represents the memory size allocated for RD: in 512 byte sectors. For example, a value of -128 means that 64K of memory is allocated in DRAM for RD:.

- If **$FILE_MAXSEC > 0** , then RAM disk is defined to be in the PERM pool of SRAM. Because RAM disk is a portion of SRAM, copy all RAM disk files to magnetic disks for permanent storage to prevent losing information due to loss of battery power or system software loading.

  SRAM is battery-backed volatile memory. This means that all information in SRAM, including programs, requires battery backup for information to be retained when the controller is turned off and then on again. Teach pendant programs are automatically stored in the TPP pool of SRAM when you write a program.

  Caution

  Data in SRAM can be lost if the battery is removed or loses its charge, or if new system software is loaded on the controller. To prevent loss of data, back up or copy all files to permanent storage devices such as FR: or ATA Flash PC memory cards.

- If **$FILE_MAXSEC < 0** , then RAM disk is defined to be in DRAM.

  DRAM is non-battery-backed volatile memory. This means that all information in DRAM disappears between power cycles. In effect, DRAM is a temporary device. Information stored in DRAM is lost when you turn off the controller.

Caution

Data in DRAM will be lost if you turn off the controller or if the controller loses power. Do not store anything you want to save beyond the next controller power cycle in DRAM, otherwise, you will lose it.

Note Volatile means the memory is lost when power is disconnected. Non-volatile memory does not require battery power to retain.

You can store anything that is a file on the RAM Disk. The RAM disk is already formatted for you.

Information stored on RAM disk can be stored as compressed or uncompressed. By default, information is compressed. If you want information to remain uncompressed, you must use the **RDU:** device designation to indicate that information will be saved to that device in an uncompressed file format.

FTP Ethernet Device

FTP Ethernet devices are used to copy files from the controller to the network PC or workstation if the FTP option is installed. The client devices displayed are the client devices that have been defined and started. Refer to the *Internet Options Setup and Operations Manual* for more information.

Memory Device (MD:)

The memory device (MD:) treats the controller's program memory as if it were a file device. You can access all teach pendant programs, KAREL programs, and KAREL variables loaded in the controller.

The Memory Device is a group of devices (MD:, MDB:, and optionally FMD:) that provide the following :

- MD: provides access to ASCII and binary versions of user setup and programs
- MDB: provides access to binary versions of user setup and programs (similar to "backup - all of the above" on the teach pendant file menu)
- FMD: provides access to ASCII versions of user setup and programs filtered to include only user settable information (eg. internal timers or time system variables changed by the system are not included) making these files useful for detecting user changes.

Memory Device Binary (MDB:)

The memory device binary device (MDB:) allows you to copy the same files as provided by the Backup function on the File Menu. This allows you to back up the controller remotely such as from

SMON, FTP, or KCL. For example, you could use the MDB: device to copy all teach pendant files (including invisible files) to the memory card (KCL>copy MDB:*.tp TO mc:).

MF Device (MF:)

MF: is a composite device that will search the RAM Disk (RD:) and flash file storage disk (FR:) devices, in that order, for a specified file. MF: eliminates your need to know the name of the device that contains the file you specify. For example, "DIR MF:file.ext" will search for the file first on RD:. If it is not found, it will search for the file on FR:. Also, "COPY MC:file.ext to MF" will place the file on RD:.

When files are copied to the MF: device, the RAM Disk is used by default if RD: is in SRAM($FILE_MAXSEC > 0). The Flash ROM disk is used by default if RD: is in DRAM ($FILE_MAXSEC < 0).

Note When you are backing up files, the MF: device will prompt you to select either FR: or the RD: device. The files will be copied to the device that you selected even if RD: is in DRAM.

Filtered Memory Device (FMD:)

The Filtered Memory Device option generates text versions of all backup files of user programs and variables that have been changed manually. Included are system and KAREL variables, position and data registers, teach pendant programs, and I/O configuration data.

When logging into the robot FTP server from a remote client you are defaulted into the MD: device. You can navigate to other robot file devices (FR:, RD:, MC:, MDB:, FMD:) using the change directory service in your remote FTP client. At a command line using the *cd* command where in this example fmd: is the device being used, this might look like :

```
D:\temp>ftp pderob029
Connected to pderob029.frc.com
220 R-30iA FTP server ready.  [PaintTool V7.50P/01]
User <pderob029.frc.com:<none>>:
230 User logged in [NORM].
ftp>cd fmd:
250 CWD command successful.
ftp>
```

You can compare these files with previous versions to determine what users or operators have changed. Variables and programs that change without user input are filtered out, and will appear in filter exclusion files.

After the option is installed, it will run automatically whenever you perform an Ethernet backup of the controller from the FMD: device. After you install the Filtered Memory Device option, any of the following filter exclusion files could appear on the FR: device.



Caution

Do not delete these files, or filter exclusion data will be lost.

- FR:SVAREEG.DT
- FR:KVAREEG.DT
- FR:POSREEG.DT
- FR:REGEEG.DT
- FR:TPLINEEG.DT

You can view program, variable, or filter exclusion files via KCL. For example:

```
KCL> DIR FMD:*.*.
```

Note Computer systems that perform periodic backups could be modified to use the FMD: device instead of the MD: device for some compare operations, for example. Contact FANUC Robotics for more information.

FRA:

There is a special area for Automatic Backup in the controller F-ROM (FRA:). You do not need an external device to use Automatic Backup, but a memory card can also be used. Refer to your *application-specific Setup and Operations Manual* for more information.

9.1.2 Storage Device Hints

The following are hints regarding the storage devices that can be used for a remote backup:

MDB: This is the only true source device for the files to be backed up. It produces an up to the minute set of backup files for the controller. The files provided are the closest match to the operation of the File Menu "Backup, All of the Above" or the backup made by the Automatic Backup function.

MD: This provides the same files as MDB: but in addition it provides a text file (ASCII) equivalent to most of the files. Backups using this device will take significantly longer than ones from the MDB: device, but the text files can be useful.

FRA: This device does not contain any files until the Automatic Backup function puts them there. If the Automatic Backup function is set up to run on a regular basis, once a day for example, then this could be a good source for a Full Backup. The files provided would very closely match those provided by the MDB: device.

Typically, the best device to use to back up files using a standard PC FTP client is the MDB: device. Otherwise, if the Automatic Backup function is set up and runs properly, (at least daily) then FRA could be an option to provide the backup more quickly. However, if a site evaluates the need and usefulness of the text file equivalents provided on the "MD:" device, then this device could also be used.

9.1.3 Setting the Default Device

Setting the default device specifies which device to use when manipulating programs and files. You must set the default device before you can perform any program or file manipulations, including formatting a memory card. You can set the default device to

- Memory card (MC:)
- USB Memory Stick (UD1:)
- Flash File Storage disk (FR:) - built-in storage for controller system files.
- Client tag (C1: - C8:) - used if the FTP option is installed. The client devices displayed are the client devices that have been defined and started.
- Memory device (MD:) - treats the controller's program memory as if it were a file device.
- Console device (CONS:) - used for debug purposes only. This displays the conslog.ls and constail.ls powerup log files.
- RAM disk - RAM disk, specified by RD: (Not for SpotTool+)
- MF device (MF:) - a composite device that searches the RAM Disk (RD:) and flash file storage disk (FR:) devices, in that order, for a specified file.

After you set the default device, the device will remain the default until you change it.

Use Procedure 9-1 to set the default device.

Procedure 9-1 Setting the Default Device



Caution

If a device such as a vision system are connected to the controller, always turn on the robot first, then turn on these devices; otherwise, equipment could be damaged.

Conditions

- **If you are setting the default device to memory card or USB memory stick,** the memory card or memory stick must be properly installed. Refer to Section 9.1.5 .

Steps

1. Press MENUS.
2. Select FILE.
3. Press F1, [TYPE].
4. Select File. You will see a screen similar to the following.

```
FILE
MC:\*.*
  1   *       *   (all files)
  2   *       KL  (all KAREL source)
  3   *       CF  (all command files)
  4   *       TX  (all text files)
  5   *       LS  (all KAREL listings)
  6   *       DT  (all KAREL data files)
  7   *       PC  (all KAREL p-code)
  8   *       TP  (all TP programs)
  9   *       MN  (all MN programs)
 10   *       VR  (all variable files)
 11   *       SV  (all system files)
 12   *       IO  (I/O config data)
 13   *       DF  (all DEFAULT files)
 14   *       ML  (all part model files)
 15   *       BMP  (all bit-map images)
 16   *       PMC  (all PMC files)
 17   [you enter]
Press DIR to generate directory
```

5. Press F5, [UTIL].
6. Select Set Device.
7. Move the cursor to the device you want to select and press ENTER. You will see a screen similar to the following.

```
FILE MC:\*.*
```

The default device is now set. The default device name is displayed on the FILE screen, under the word "FILE."

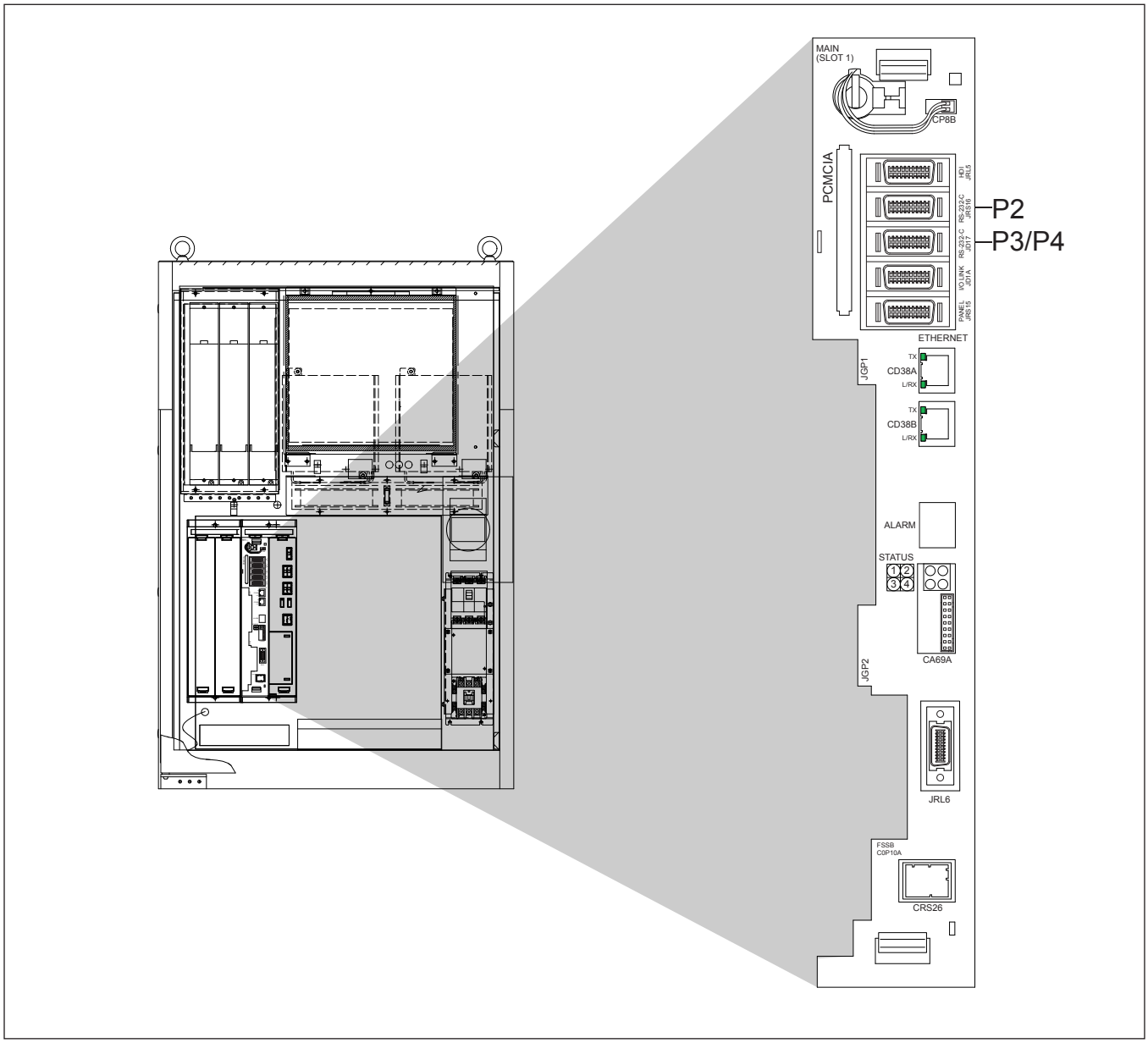
9.1.4 Setting Up a Port

Setting up a port means initializing controller serial ports to use specific devices, such as the CRT/KB. Initializing ports involves setting up specific information for a port based on the kind of device that will connect to the port. This is done on the teach pendant PORT INIT screen.

The controller supports up to four serial ports. Several different kinds of devices can be connected to these ports.

Figure 9–1 shows the location of the ports on the controller.

Figure 9–1. Location of Ports on the Controller



Ports

Up to four ports are available, P1-P4. Table 9–1 lists the ports. You can set up ports P2 through P4 if you have them, but you cannot set up the teach pendant port, P1.

Table 9–1. Ports P1 - P4

Port	Item Name on Screen	Kind of Port	Use	Default Device
P1	Teach Pendant **Note** This is a dedicated port and cannot be changed.	RS-422	Teach pendant	Teach pendant
P2	JRS16 RS-232–C	RS-232-C	Any device	Maintenance Console
P3	JD17 RS-232–C on Main CPU card	RS-232-C		KCL
P4	JD17 on Main CPU card. This port is displayed on the teach pendant if $RS232_NPORT=4.	RS-422		No use

Devices

You can modify the default communications settings for each port except port 1, which is dedicated to the teach pendant (TP). Table 9–2 lists the default settings for each kind of device you can connect to a port.

Table 9–2. Default Communications Settings for Devices

Device	Speed (baud)	Parity Bit	Stop Bit	Timeout Value (sec)
Sensor*	4800	Odd	1 bit	0
Host Comm.*	4800	Odd	1 bit	0
KCL/CRT	9600	None	1 bit	0
Maintenance Console	9600	None	1 bit	0
Factory Terminal	9600	None	1 bit	0
TP Demo Device	9600	None	1 bit	0
No Use	9600	None	1 bit	0
Current Position (for use with the Current Position option)	9600	None	1 bit	0
PMC Programmer	9600	None	2 bit	0
Modem/PPP	Refer to the *FANUC Robotics Internet Options Setup and Operations Manual* for information on the supported modems.			
HMI Device	19200	Odd	1 bit	0

*You can adjust these settings; however, if you do, they might not function as intended because they are connected to an external device.

RS-232-C Interface

On the SETUP Port screen, you can choose the RS-232-C interface on one of the following ports:

- Port 1 (P2:)
- Port 2 (P3:)
- Port 3 (P4:) (RS-222) available if $RS232_NPORT is set to 4

The maximum cable length is approximately 50 feet (15 meters). Consult the RS–232–C Industry Standard for more information.

Connector Pin Configuration for P2:, and P3:

The connector pin configuration for P2: is normal Data Terminal Equipment (DTE). For the pin configuration on P3: for the RS-232-C interface on JD17 (on the operator panel board), refer to the *Controller Maintenance Manual*.



Warning

***Pins 10 and 19 on JD17 are +24 volts. Be careful; otherwise, you could injure personnel or damage equipment.**

Use Procedure 9-2 to set up a port.

Procedure 9-2 Setting Up a Port

Conditions

- The default device is set.

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Port Init. You will see a screen similar to the following.

```
SETUP Port Init
     Connector      Port     Comment
1 JRS16 RS-232-C      P2:   [Maintenance Cons]
2 JD17  RS-232-C      P3:   [KCL/CRT         ]
3 JD17 RS-422         P4:   [                ]
```

5. Move the cursor to the port you want to set up and press F3, DETAIL. You will see a screen similar to the following.

```
SETUP Port Init
JRS16 RS-232-C         P2:
   1 Device             [Maintenance Cons]
   2 Speed(Baud rate)              [9600 ]
   3 Parity bit                    [None ]
   4 Stop bit                     [1bit  ]
   5 Time out value(sec)             [  0]
```

6. Select each item and press F4, [CHOICE], to select the appropriate value.

Note To indicate that you are not using a port, set the port to *No Use* . Set the port to no use if you are performing a READ/WRITE from a KAREL program.

7. A device cannot be assigned to two ports. To move a device to another port, set the existing port to **No Use** and then assign the device to another port.

8. Perform a Cold start of the controller to implement the changes to the Port Init screen.

RS-422 Interface

The RS-422 interface is available on port 3 (P4:).

- The maximum cable length is approximately 50 meters.
- RS-422 provides more noise rejection than RS-232C
- The electrical signal of RS-422 is different from the RS-232-C signal. If you need to connect between a robot controller and personal computer, you will need a converter, because normally a personal computer does not support the RS-422 interface.

  If you are using ArcTool , RS-422 is useful because the data transfer function or sensor interface fails sometimes due to electrical noise.

Connector Pin Configuration for P4:

Refer to the *Controller Maintenance Manual* for the pin configuration of the P4: port JD17 connector, located on the MAIN CPU Card or the operator panel board if you are using the optional cable. Table 9–3 lists the Port Init screen items you can set up. Use Procedure 9-3 to set up a port to use the RS-422 Interface.



Warning

***Pins 10 and 19 on JD17 are +24 volts. Be careful; otherwise, you could injure personnel or damage equipment.**

Table 9–3. SETUP Port Init Screen Items

ITEM	DESCRIPTION
Connector	This column shows the type of connector/protocol.
Port	This column shows the physical port on the robot.
Comment	This column shows the comments on port settings.

Procedure 9-3 Setting Up a Port to Use the RS-422 Interface

Conditions

- The default device is set. (Procedure 9-1)
- The RS-422 cable is connected to the JD17 connector on the Panel Board.

Steps

1. Perform a Controlled start:
   a. If the Controller is turned on, turn off the power disconnect circuit breaker.
   b. **On the teach pendant** , press and hold the PREV and NEXT keys.
   c. While you hold these keys, turn the power disconnect circuit breaker to ON. You will see a screen similar to the following.

```
---------- CONFIGURATION MENU ----------
1 Hot start
2 Cold start
3 Controlled start
4 Maintenance
Select >
```

d. Release all the keys.

e. Select Controlled start and press ENTER.

2. Press MENUS.
3. Select Variables.
4. Move the cursor to $RS232_NPORT.
5. Press ENTER.
6. Type 4 and press ENTER.
7. Press FCTN and select START (COLD).
8. Press MENUS.
9. Select Setup.
10. Press F1, [TYPE].
11. Select Port Init. You will see a screen similar to the following.

```
SETUP Port Init
     Connector       Port     Comment
1 JRS16 RS-232-C      P2:   [Maintenance Cons ]
2 JD17 RS-232-C       P3:   [KCL/CRT          ]
3 JD17 RS-422         P4:   [No Use           ]
```

12. Move the cursor to JD17-422 and press F3, DETAIL. You will see a screen similar to the following.

```
SETUP Port Init
JD17 RS-422-C          P4:
   1 Device              [Maintenance Cons]
   2 Speed(Baud rate)                [9600 ]
   3 Parity bit                      [None ]
   4 Stop bit                       [1bit  ]
   5 Time out value(sec)               [  0]
```

13. Select each item and set as desired.

9.1.5 Using a Memory Card or USB Memory Stick Interface

The memory card and USB memory stick interfaces are integral to the controller.

Use Procedure 9-4 to use the memory card interface or the USB memory stick device.

Procedure 9-4 Using the Memory Card or USB Memory Stick Interface

Conditions

- You are using a ATA Flash memory card that is based on one of the following standards:
  - — JEIDA "IC Memory Card Guideline Version 4.0"
  - — PCMCIA "PC Card Standard R. 2.0"

- You are using a memory stick that is compatible with a USB 1.1 interface. The USB Organization specifies standards for USB 1.1 and 2.0. Most memory stick devices conform to the USB 2.0 specification for operation and electrical standards. USB 2.0 devices as defined by the USB Specification be backward compatible with USB 1.1 devices.

  However, FANUC Robotics does not support any security or encryption features on USB memory sticks. The controller supports most widely-available USB Flash memory sticks from 32MB up to 1GB in size.

  Note Generally the larger the size of the device such as a USB memory stick, the slower the access speed and device performance.



Caution

The memory card and USB interfaces are fragile. Use care when inserting the memory card or memory stick to avoid damage to the connector. Do not remove a memory card or USB memory stick when the controller is reading or writing to it. Doing so could damage the media and lose all information stored on it.

Steps

1. If you are using a memory card, turn the power disconnect circuit breaker to OFF.

Warning

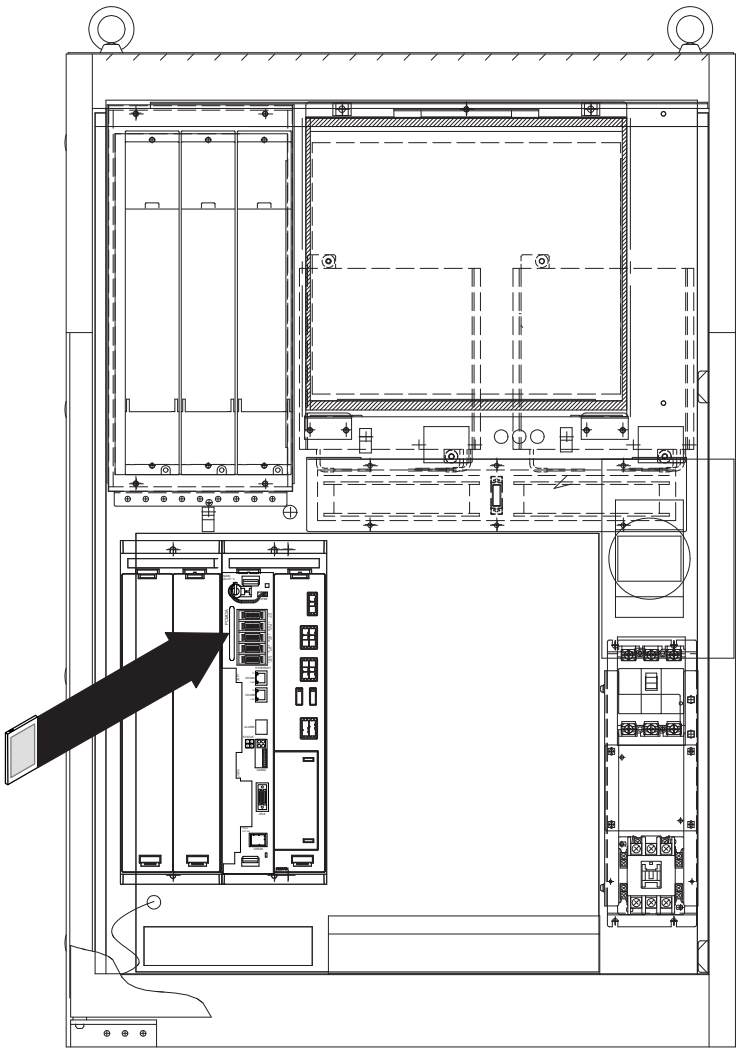
Lethal voltage is present in the controller WHENEVER IT IS CONNECTED to a power source. Be extremely careful to avoid electrical shock. HIGH VOLTAGE IS PRESENT at the input side whenever the controller is connected to a power source. Turning the disconnect or circuit breaker to the OFF position removes power from the output side of the device only.

Warning

When the power disconnect circuit breaker is OFF, power is still present inside the controller. Turn off the power disconnect circuit breaker before you insert a memory card into the memory card interface; otherwise, you could injure personnel.

a. Identify the memory card interface. See Figure 9–2 .

Figure 9–2. Memory Card Interface Location



b. Insert the memory card as shown in Figure 9–2 .

2. If you are using a USB memory stick device,

a. Identify the USB interface. See . Figure 9–3 or Figure 9–4 .

Figure 9–3. USB Memory Stick Device Location

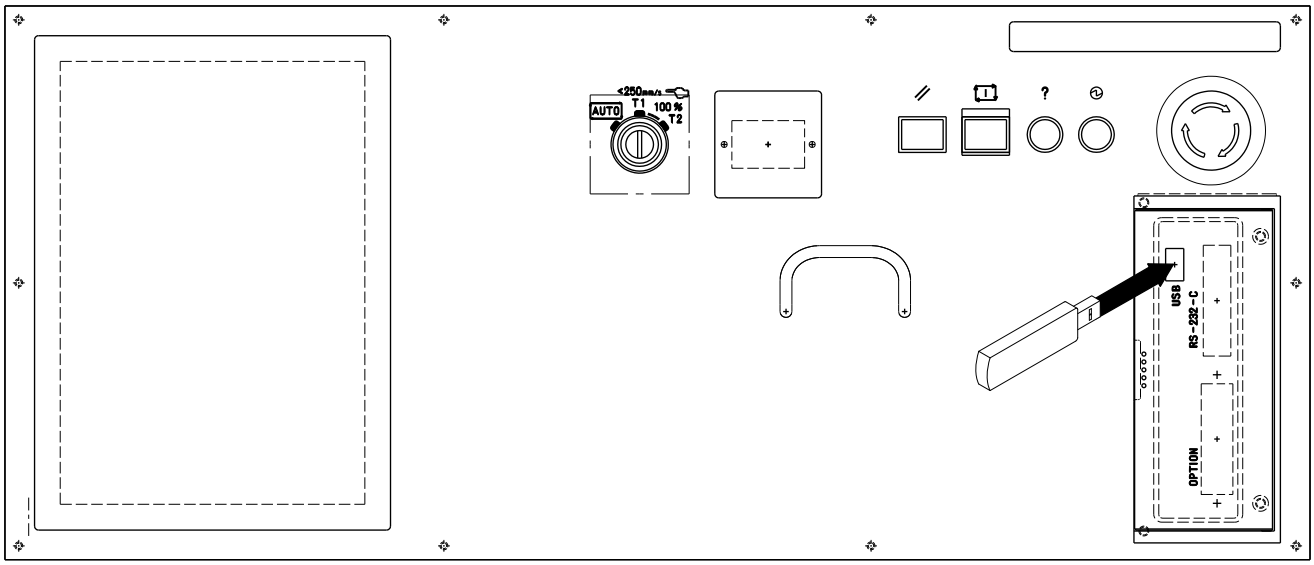
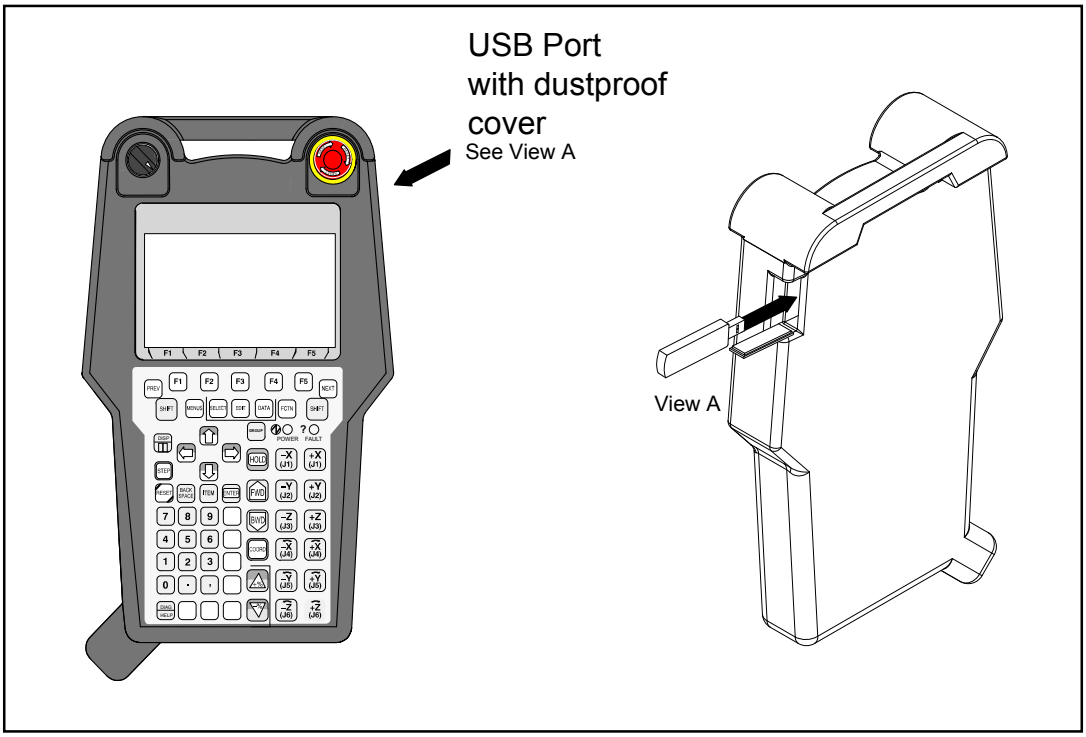


Figure 9–4. New *i*Pendant USB Port Interface Location



b. Insert the memory stick as shown in Figure 9–3 or in Figure 9–4 .

9.1.6 Formatting Devices

You must format the FLASH ATA memory card or USB memory stick only before you use them for the first time. Use Procedure 9-5 to format a memory card or memory stick.

Note USB Memory sticks cannot be formatted when plugged into the USB connector on the iPendant (UT1:) They must be formatted on a PC or when plugged into the front panel USB connector on the controller.

The RAM disk is formatted automatically for you each time you change the disk size. If you want to format the RAM Disk from KCL, use Procedure 9-6 .

Note SpotTool+ does not use the RAM disk.



Caution

Formatting deletes all the files on a memory card. Do not format a memory card or USB memory stick that contains files you want to keep.

Procedure 9-5 Formatting a Memory Card or USB Memory Stick from the File Menu

Conditions

- The default device is set to MC:. Refer to Procedure 9-1 .
- The FLASH ATA memory card is not write protected (if it has the write protect option).
- If you are formatting a memory card, the power disconnect circuit breaker is OFF and the memory card is inserted in the memory card interface properly. Refer to Procedure 9-4 .



Warning

When the disconnect or circuit breaker handle is OFF, power is still present inside the controller. Disconnect electrical power from the controller before you remove or replace components, or you could be injured seriously.

- If you are formatting a USB memory stick, hold the card to be formatted with the label toward you and insert it into the interface. Refer to Procedure 9-4 .

Steps

1. Turn on the power disconnect circuit breaker to turn on the controller.
2. Press MENUS.
3. Select FILE.

4. Press F5, [UTIL] and select Format (FAT16) or Format FAT32 to format the card or memory stick.

Note The device must be 2 Gbytes or smaller in size to be formatted as FAT16.

Select Format which is a FAT16 type of format if:

- More than 255 files do not need to be stored in the root directory of the media.
- It is necessary to write files to the media quickly.
- Larger quantities of files can still be stored in sub-directories.
- If the device is smaller than 64 Mbytes

Note The device must at least 64 Mbytes in size to be formatted as FAT32.

Select Format FAT32 which is a FAT32 type of format if:

- There is no limit to the number of files stored in the root directory of the media.
- If the device is larger than 2 Gbytes (2.0 GB).

If you are formatting a memory card, you will see a screen similar to the following.

```
File Format
MC:\*.*
                Formatting MC:
    *************  WARNING  *************
      ANY DATA ON THE DISK WILL BE LOST!
Insert the disk to be formatted into the
                interface
Format disk?
```

If you are formatting a USB memory stick, you will see a screen similar to the following.

```
File Format
UD1:\*.*
               Formatting UD1:
    *************  WARNING  *************
      ANY DATA ON THE DISK WILL BE LOST!
Insert the disk to be formatted into the
                interface
Format disk?
```

5. Format the card:

   - **If you do not want to format the memory card or memory stick,** press F5, NO.
   - **To format the memory card or memory stick,** press F4, YES. You will see a screen similar to the following.

```
1 Words
2 Upper Case
3 Lower Case
4 Options
                                    --Insert--
Enter volume label:
```

6. Use the appropriate function keys and numeric keys to type a volume label, such as usb1 or mcard1, and press ENTER.

   Formatting takes a few seconds. When the formatting is complete the teach pendant FILE menu will be displayed.

If your application includes KCL and you want to format the RAM disk , use Procedure 9-6 to format the RAM disk from KCL.

Procedure 9-6 Formatting the RAM Disk from KCL

Conditions

- The controller has a minimum of 2 MB of SRAM.
- The default device is set to RAM Disk. Refer to Procedure 9-1 .
- You have KAREL.
- You have enough memory in the Permanent Memory Pool.

Steps

1. Press MENUS.
2. Select KCL>.
3. Type the following and press ENTER:

```
SET VAR $FILE_MAXSEC = n
```

   where **n** is the size of the RAM disk divided by 512. For a 512 Kbyte RAM disk, **n** is 1000.

4. **If you are using PaintTool or HandlingTool,** perform a Cold start of the controller to implement the changes to the Port Init screen:
5. **If you are NOT using PaintTool or HandlingTool,** perform a Cold start of the controller to implement the changes to the Port Init screen:
6. Press MENUS.
7. Select KCL>.
8. Type the following and press ENTER:

```
FORMAT RD:
```

9. Type the following and press ENTER:

```
MOUNT RD:
```

10. Type the following and press ENTER:

```
SHOW DEVICE RD:
```

11. Type the following and press ENTER:

```
CD RD:
```

9.1.7 Subdirectories

Subdirectories can be used to organize programs or files on the controller. Subdirectories can be created on the FILE menu. Use to create a directory.

Procedure 9-7 Using Subdirectories

1. Press MENUS.
2. Select FILE.
3. Press F5, [UTIL].
4. Select Make DIR. You will see a screen similar to the following.

Figure 9–5. Making Subdirectories

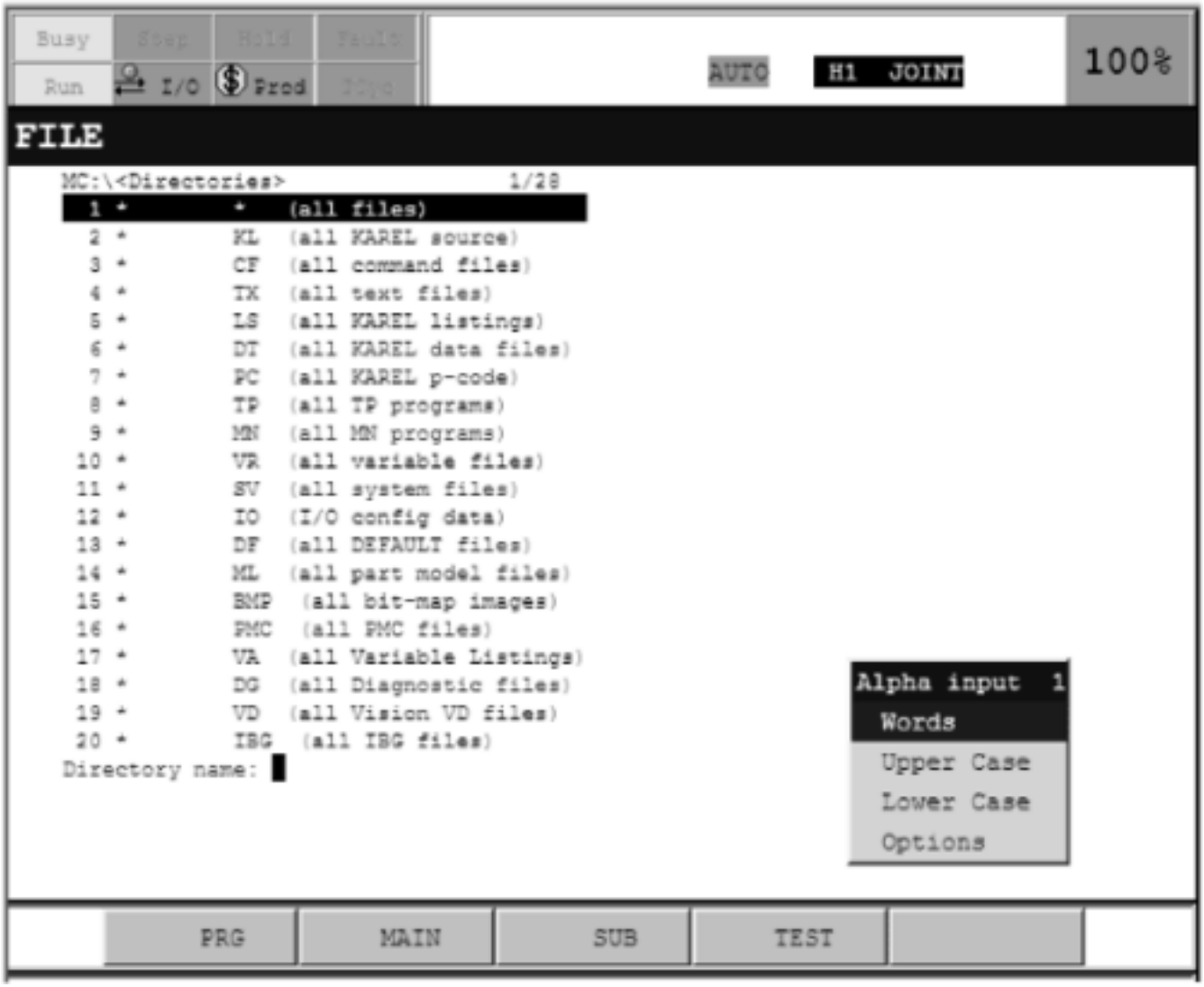


Figure 9–6. Displaying Subdirectories

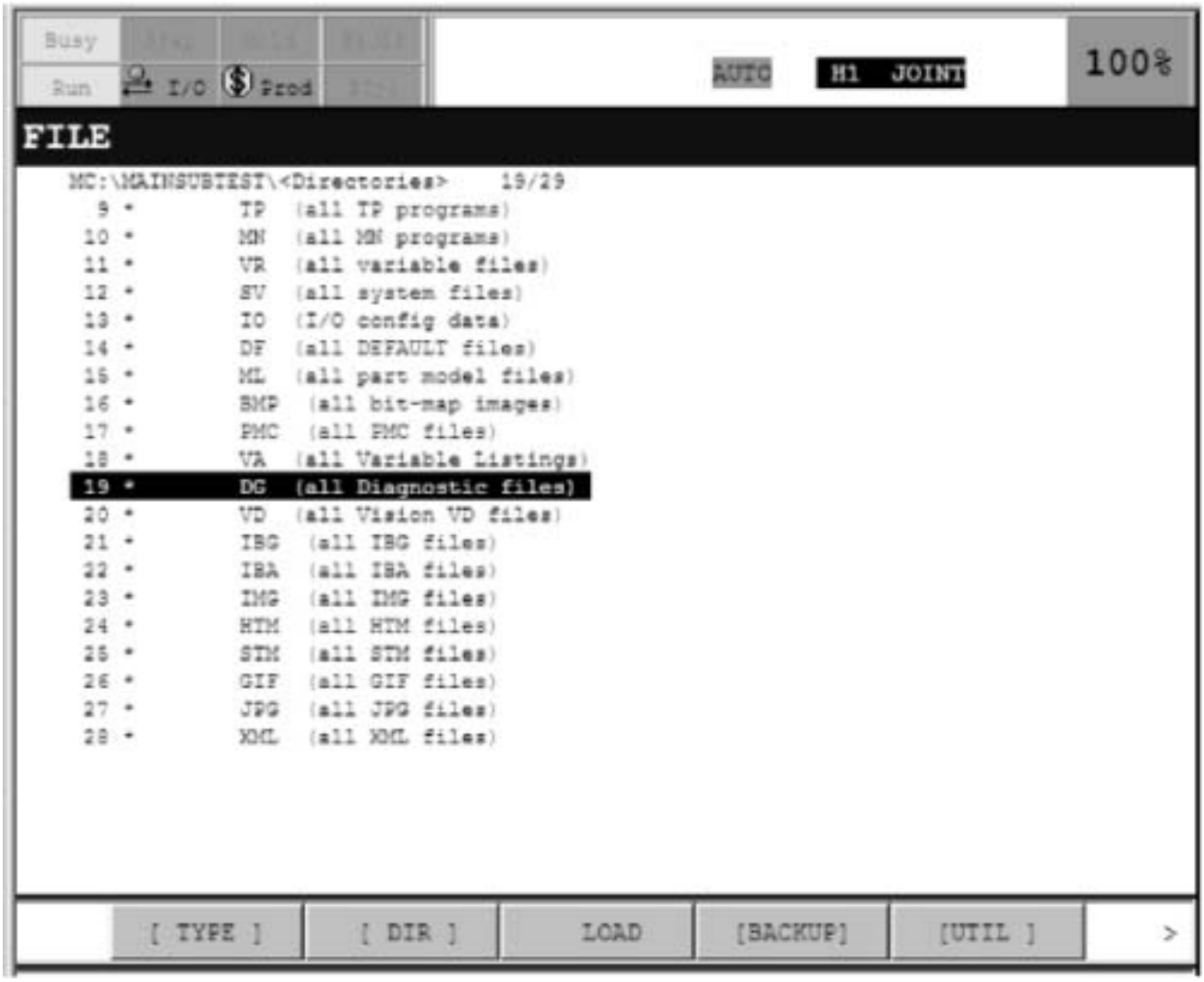
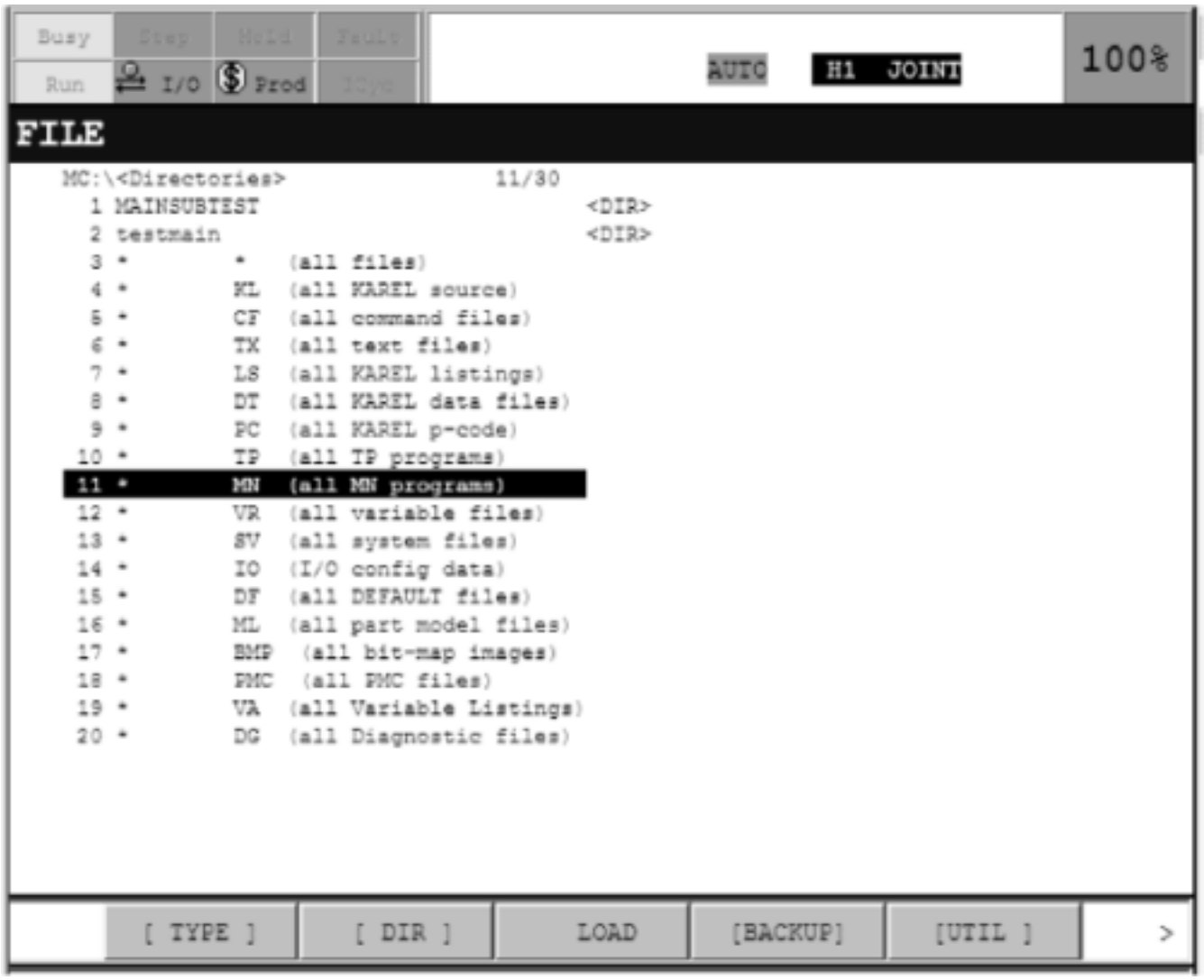


Figure 9–7. Subdirectories



5. Press F2, [DIR], and select Directories.
6. Move the cursor to “(Up one level), and press the right arrow key to display the available directories. You will see a screen similar to the following.
7. Use the left arrow key to back out of the directory display .

9.2 MANIPULATING PROGRAMS

9.2.1 Overview

A program is a series of robot commands that tells the robot and other equipment how to move and what to do to perform an application. As programs are created they are stored automatically on controller memory. A list of all programs stored on controller memory is displayed on the SELECT menu. See the following screen for an example.

```
Select
               702566 bytes free   7/7
No.  Program name           Comment
 1   -BCKEDT-         [                   ]
 2   HOME_IO          [SET AT HOME I-O    ]
 4   TEST1            [TEST1              ]
 5   TEST111_PLAN>    [                   ]
 6   UOP_ROBOT_12>    [                   ]




TEST111_PLANT1_ROBOT15467_A
```

Programs can be:

- Selected
- Saved
- Loaded
- Copied within the SELECT menu
- Deleted from the SELECT menu
- Monitored
- Printed to a file

A list of all KAREL programs loaded in controller memory is displayed on the SELECT menu as .PC types. KAREL programs that have variables but no .PC files that are loaded in controller memory are displayed on the SELECT menu as .VR types. These KAREL programs are only displayed if the system variable, $KAREL_ENB, is set to a value of 1.

Note Programs with invisible attributes cannot be viewed on the Select menu.

To select which program types will be displayed in the SELECT menu use F1, [TYPE].

9.2.2 Selecting Programs On the SELECT Menu

You can select programs on the SELECT menu. Selecting a program chooses the program as the current program, for modifying, testing, or executing. Use Procedure 9-8 to select a program on the SELECT menu.

Procedure 9-8 Selecting a Program on the Select Menu

Steps

1. Press SELECT. You will see a screen similar to the following.

```
Select
               702566 bytes free   7/7
No.  Program name           Comment
 1   -BCKEDT-         [                    ]
 2   HOME_IO          [SET AT HOME I-O     ]
 4   TEST1            [TEST1               ]
 5   TEST111_PLAN>    [                    ]
 6   UOP_ROBOT_12>    [                    ]




TEST111_PLANT1_ROBOT15467_A
```

Note Any program name that is longer than 12 characters is abbreviated on the SELECT screen and displayed with a > at the end. When the program name is selected, the entire program name is displayed at the bottom of the screen.

Note If you have common characters in long program names, you can set $UI_CONFIG.$PROG_COMMON[1–3] to contain the common characters. For instance, if your program name is TEST111_PLANT1_ROBOT15467_A and TEST111 is a common prefix, then set $UI_CONFIG.$PROG_COMMON[1] = TEST111. When the system needs to truncate the program name it will remove the common portion and replace it with a tilde (~) character. See the following screen for an example.

```
Select
               702566 bytes free   7/7
No.  Program name           Comment
 1   -BCKEDT-         [                   ]
 2   HOME_IO          [SET AT HOME I-O    ]
 4   TEST1            [TEST1              ]
 5   ~_PLANT1_ROBOT1>[                    ]
 6   UOP_ROBOT_12>    [                   ]




TEST111_PLANT1_ROBOT15467_A
```

2. Press F1, [TYPE].
3. Select from the following list:

   Note The list might vary depending on your software configuration.

   - **All** displays all programs.
   - **Jobs** displays all job programs.
   - **Processes** displays all process programs.
   - **TP Programs** displays all teach pendant programs.
   - **KAREL Progs** displays all KAREL programs.
   - **Macro** displays all macro programs.
   - **Cond** displays all condition monitor programs.

4. Select the name of the program you want and press ENTER.

9.2.3 Saving Programs

Saving programs allows you to save a program and its relevant data to the default device. Use Procedure 9-9 to save a program.

Procedure 9-9 Saving a Program

Conditions

- **If you are saving programs to a memory card or USB memory stick** , make sure that the memory card or memory stick is installed properly. Refer to Section 9.1.5 .

Steps

1. Press SELECT. You will see a screen similar to the following.

```
Select
               702566 bytes free   7/7
No.  Program name           Comment
 1   -BCKEDT-         [                   ]
 2   HOME_IO          [SET AT HOME I-O    ]
 4   TEST1            [TEST1              ]
 5   TEST111_PLAN>    [                   ]
 6   UOP_ROBOT_12>    [                   ]




TEST111_PLANT1_ROBOT15467_A
```

2. Move the cursor to the program you want to save.
3. Press NEXT, >, and then press F4, SAVE AS. If you selected the program name TEST.TP, you will see a screen similar to the following.

```
From:          TEST.TP
To Device:     MC:
To Directory:  \
To Filename:   TEST.TP
```

4. **To change the default device** , move the cursor to To Device: and press F4, [CHOICE].
5. Move the cursor to To Filename: and press F4, [CHANGE].
6. Select the method you want to use to name the new program, and press ENTER.
7. **To complete the operation,** press F1, DO_SAVE.

Note If the new program exists when you press F1, DO_SAVE, then you must confirm the overwrite operation before the save will be completed.

9.2.4 Loading Programs

Loading programs allows you to load programs from the default device onto controller memory. A program **must** be loaded into controller memory and listed on the SELECT menu before it can be modified or executed. Use Procedure 9-10 to load programs.

Procedure 9-10 Loading a Program

Conditions

- **If you are loading programs from a memory card or USB memory stick** , make sure that the memory card or memory stick is installed properly. Refer to Section 9.1.5 .

Steps

1. Set the default device:
   a. Press MENUS.
   b. Select FILE.
   c. Press F1, [TYPE].
   d. Select File.
   e. Press F5, [UTIL].
   f. Select Set Device.
   g. Move the cursor to the device you want and press ENTER.
2. Press SELECT. You will see a screen similar to the following.

```
Select
               702566 bytes free   7/7
No.  Program name           Comment
 1   -BCKEDT-         [                   ]
 2   HOME_IO          [SET AT HOME I-O    ]
 4   TEST1            [TEST1              ]
 5   TEST111_PLAN>    [                   ]
 6   UOP_ROBOT_12>    [                   ]




TEST111_PLANT1_ROBOT15467_A
```

3. Press NEXT, >, and then press F3, LOAD. You will see a screen similar to the following.

```
    ---  Load Teach Pendant Program ---
    
    


Enter program name
```

4. Type the program name to load and press ENTER.

   Note Do not include the file extension.

5. Load the selected program:

   - **If you do not want to load the selected program,** press F2, NO.
   - **If you want to load the selected program,** press F1, YES.

   Note If the teach pendant program does not load, you must perform a controlled start. Refer to Appendix C to perform a controlled start and then repeat this procedure.

   Note If you see the message "Protection error occurred," the same program already exists on the controller and it is write protected. To load that program, change its name and then load it.

   Note If you see the error message SCIO-016, you are trying to load a program that uses the remote TCP motion option but do not have the remote TCP motion option loaded on your controller. If this happens, load the remote TCP option at controlled start and then reload the program. Refer to the *Software Installation Manual* for information on how to load options.

   Note You might need to purge Flash ROM after repeated loading of KAREL programs. Each load uses some Flash ROM to store the loaded program. When the program is reloaded, a new portion of Flash ROM is used and the old portion is not released until a purge is performed. Refer to Section 9.3.12 for information on purging.

   The program you specified will be loaded from the default device onto controller memory. The SELECT menu will be displayed and the loaded program will appear on the menu.

   Note If the program you are loading is a line or rail tracking program and was made on an off-line PC, you must manually process all the line tracking information before you can run the program in production. Refer to the "Testing a Program and Running Production" chapter for more information.

9.2.5 Copying Programs Within the SELECT Menu

Programs can be copied within the SELECT menu. This means that both the original program and the copied program will be on controller memory. Use Procedure 9-11 to copy programs within the SELECT menu.



Warning

Before you copy a program with embedded macros from one controller to another, compare the SETUP menu macro lists of the two controllers. Be sure that the list on the first controller matches the list on the second controller. If they are not identical, DO NOT copy the program; otherwise, you could injure personnel or damage equipment.

Procedure 9-11 Copying a Program within the SELECT Menu

Steps

1. Press SELECT. You will see a screen similar to the following.

```
Select
               702566 bytes free   7/7
No.  Program name           Comment
 1   -BCKEDT-         [                   ]
 2   HOME_IO          [SET AT HOME I-O    ]
 4   TEST1            [TEST1              ]
 5   TEST111_PLAN>    [                   ]
 6   UOP_ROBOT_12>    [                   ]




TEST111_PLANT1_ROBOT15467_A
```

2. Move the cursor to the program you want to copy.
3. Press NEXT, > and then press F1, COPY. You will see a screen similar to the following.

```
---  Copy Teach Pendant Program ---
    From:
       TEST111_PLANT1_ROBOT15467_A
    To:
       ▌
```

4. Type the program name to which to copy the selected program. Press ENTER.
5. If necessary, type the sub type for the copied program and press ENTER.
6. Copy the selected program:

   - **If you do not want to copy the selected program,** press F5, NO.
   - **If you want to copy the selected program,** press F4, YES.

   The selected program will be copied. The SELECT menu will be displayed. If the copied program has a new name, it will be displayed in the SELECT menu.

9.2.6 Deleting Programs from the SELECT Menu

If you no longer want to have a program loaded on controller memory (displayed on the SELECT menu) you can delete it. If you want to keep a copy of the program, save it to a storage device before you delete it from the SELECT menu.

Note Deleting a program from controller memory does not delete it from the storage device, if a copy exists on one of those devices.

Use Procedure 9-12 to delete a program from the SELECT menu. For information on deleting a program from a storage device, refer to Procedure 9-20 .

Procedure 9-12 Deleting a Program from the SELECT Menu

Conditions

- The program you want to delete is listed on the SELECT menu.

Steps

1. Press SELECT. You will see a screen similar to the following.

```
Select
               702566 bytes free   7/7
No.  Program name           Comment
 1   -BCKEDT-         [                   ]
 2   HOME_IO          [SET AT HOME I-O    ]
 4   TEST1            [TEST1              ]
 5   TEST111_PLAN>    [                   ]
 6   UOP_ROBOT_12>    [                   ]



TEST111_PLANT1_ROBOT15467_A
```

2. Move the cursor to the name of the program you want to delete.
3. Continuously press the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.
4. Press NEXT, >, and then press F3, DELETE. See the following screen for an example.

```
Select
               702566 bytes free   7/7
No.  Program name           Comment
 1   -BCKEDT-         [                   ]
 2   HOME_IO          [SET AT HOME I-O    ]
 4   TEST1            [TEST1              ]
 5   TEST111_PLAN>    [                   ]
 6   UOP_ROBOT_12>    [                   ]



Delete OK ?
```

5. Delete the program:

   - **If you do not want to delete the selected program,** press F5, NO.
   - **If you want to delete the selected program,** press F4, YES. The program will be deleted from controller memory. The SELECT menu will be displayed and the deleted program will no longer be listed.

     Note You cannot delete a program that is paused, assigned to a macro, or write-protected. If the program is paused, you must abort it (press FCTN and select ABORT ALL). If the

program is assigned to a macro, it must be "unassigned" first on the SETUP Macro screen. Refer to Section 3.3.2 .

6. To delete an assigned macro:
   a. Press SELECT.
   b. Press NEXT, >.
   c. Press F2, DETAIL, and change the sub type to NONE. When you have finished, press F1, [END].
   d. Press MENUS.
   e. Select SETUP.
   f. Press F1, [TYPE], select MACRO, and press ENTER.
   g. To clear all assignments, press F2, CLEAR.
   h. Press SELECT and select the name of the process to delete.
   i. Repeat Step 4 and Step 5 to delete the process program.

9.2.7 Saving Files as ASCII

Use Procedure 9-13 to save a teach pendant screen.

Note If the controller is connected to a PC, print will generate a listing file called TPSCRN.LS on that device.

ASCII File Output

You can save the file program listing to an ASCII file. The output depends on the format of the device.

For example,

- For MC:, the output is a .LS file
- For RD: or FR:, the output is a .LS file
- For KCL, the output is displayed on the KCL screen

Refer to Table 9–4 for information on how a file will be output when you print it, under various conditions.

Table 9–4. File Output Using PRINT

Operation	Output data	File name
Press F5, PRINT, on the second page of the SELECT screen.	The current program selected by the cursor.	(program name).LS
Select PRINT SCREEN on the FCTN menu at the teach pendant.	The current teach pendant screen image. If this file already exists, the new information is appended to the TPSCRN.LS file.	TPSCRN.LS
Select PRINT SCREEN on the FCTN menu at the CRT.	The current CRT screen image.	CTSCRN.LS

ASCII Files (.LS)

You can print an ASCII file to a memory card. When you save an ASCII file to an MS-DOS formatted memory card, you can read the file with an editor on a personal computer. You can also print the ASCII file using a printer connected to a personal computer.

Procedure 9-13 Saving a Teach Pendant Screen as ASCII

Steps

1. Set the default device to the device you want:
   a. Press MENUS.
   b. Select FILE.
   c. Press F1, [TYPE].
   d. Select File.
   e. Press F5, [UTIL].
   f. Select Set Device.
   g. Move the cursor to the device you want. If you want to save the screen as an ASCII file, select the device on which you want to save the file and press ENTER.
2. Display the screen you want to print or save as ASCII.
3. **To save the current screen as ASCII to a file called TPSCRN.LS:**
   a. Press FCTN.
   b. Select PRINT SCREEN. The file will be written to TPSCRN.LS on the default device.

   Note If TPSCRN.LS does not exist, it will be created. If this file already exists, the new information will be appended to the previous information in the TPSCRN.LS file. If you repeat Step 3 , data will be continuously appended to the existing data in TPSCRN.LS.

9.2.8 Filtering Programs in the Select List

Sometimes the list of programs in the select menu can become too long to be manageable. System variables are provided to limit the programs displayed based on a text filters. This is similar to selecting a program type from the [TYPE] menu.

One easy example of this would be if you wanted only one letter of the alphabet. In this case you can set up a filter to only display programs beginning with a certain letter.

When you apply this filter the [TYPE] menu changes to reflect how the filter is set up. These filters are set up by system variables. The “default” filter is to provide a [TYPE] list corresponding to program type.

Table 9–5. System variables to control filtering

$PGINP_FLTR	Defines which filter to use
$PGINP_WORD	Defines 5 words to match for display
$PGINP_TYPE	Defines 30 words to match for display

Table 9–6. $PGINP_FLTR values

$PGINP_FLTR = 0	Default setting, display program types in [TYPE] menu
$PGINP_FLTR = 1	$PGINP_WORD values are displayed in [TYPE] menu along with program types.
$PGINP_FLTR = 2	$PGINP_WORD values are displayed in [TYPE] menu instead of program types.
$PGINP_FLTR = 5	$PGINP_TYPE values are displayed in [TYPE] menu along with program types.
$PGINP_FLTR = 6	$PGINP_TYPE values are displayed in [TYPE] menu instead of program types.

Filtering works on matching the name in the filter system variable ($PGINP_WORD or $PGINP_TYPE) with the first few characters of the program name. For example if $PGINP_WORD[1] = ’MAIN’ MAIN will show up as a choice in the [TYPE] menu. When the operator selects MAIN ALL programs beginning with the word MAIN are displayed. Only programs beginning with the work MAIN are displayed.

If $PGINP_WORD is set to each letter of the alphabet the user can display only programs beginning with that letter in the SELECT list. Typically the settings in $PGINP_WORD would be more logical than a simply putting the alphabet in the filter.

9.3 MANIPULATING FILES

9.3.1 Overview

A file is a unit in which the system stores information. Files can be stored on a variety of storage media.

You perform file manipulations using the FILE screen.

See Figure 9–8 . Refer to Section 9.1 for a listing and description of the various types of storage media.

Figure 9–8. File Screen

```
FILE
MC: *.*
  1   *       *   (all files)
  2   *       KL  (all KAREL source)
  3   *       CF  (all command files)
  4   *       TX  (all text files)
  5   *       LS  (all KAREL listings)
  6   *       DT  (all KAREL data files)
  7   *       PC  (all KAREL p-code)
  8   *       TP  (all TP programs)
  9   *       MN  (all MN programs)
 10   *       VR  (all variable files)
 11   *       SV  (all system files)
 12   *       IO  (I/O config data)
 13   *       DF  (all DEFAULT files)
 14   *       ML  (all part model files)
 15   *       BMP  (all bit-map images)
 16   *       PMC  (all PMC files)
 :
 XX   [you enter]
Press DIR to generate directory
```

From the FILE screen you can:

- Generate a directory of files
- Load or restore files from the default device onto controller memory
- Back up program and system files
- Update software automatically
- Display text (ASCII) files
- Copy files to and from various devices

- Delete files from the default device
- Save files to the default device
- Create error log files
- Move files between RAM disk and FROM disk (Not for SpotTool+)
- Check and purge file memory
- Automatically update a controller from a memory card

Types of Files

To manipulate a file you must know the *type* of file you are manipulating. Table 9–7 lists several types of files available. During your work on the controller, you might only use a few types of files. You can determine the *file type* by looking at the *file name* as it is displayed on the FILE menu. The file name consists of up to 36 characters for file name, followed by a period, followed by a two- or three-letter file type:

```
file.XX
```

where **file** is the file name and **XX** is the file type.

Note File types with three characters might be displayed on the FILE screen. These types are for various kinds of compressed files. The display of these file types is controlled by the system variable $FILE_MASK. Refer to the *System Software Reference Manual* for more information.

Table 9–7. Types of Files

File Type	Description
Bit map file (.BMP)	This file type contains bit map images used in robot vision systems.
Command file (.CF)	This file type contains text (ASCII) files that contain a sequence of KCL commands for a command procedure.
Condition handler file (.CH)	This file type contains files that are used as part of the condition monitor feature.
Default file (.DF)	This file type contains binary files that contain the default motion instructions for teach pendant programming.

Table 9–7. Types of Files (Cont'd)

File Type	Description
Diagnostic file (.DG)	This file type is an ASCII file that provides you with a snapshot of the special diagnostic files on the memory device. The names of the diagnostic files are stored in the system variable $FILE_DGBCK.
Data file (.DT)	This file type contains text (ASCII) or binary files that contain any data that is needed by the user.
I/O file (.IO)	This file type contains binary files that store configuration data.
KAREL file (.KL)	This file type contains text (ASCII) files that contain the KAREL language statements for a KAREL program.
Listing file (.LS)	This file type contains text (ASCII) files that contain the listing of a KAREL language program, and line numbers for each KAREL statement. Listing files are also generated when a teach pendant screen is saved. Listing files also include error log files and other special diagnostic files.
Part model file (.ML)	This file type contains part model information used in robot vision systems.
Mnemonic(.MN)	Files of this type are supported in previous versions of the application software.
Macro (.MR)	This file type contains programs with a macro sub type.
P-Code file (.PC)	This file type contains binary files that contain the translated version of a .KL KAREL program file. This is the file that is actually loaded into controller memory and executed.
PMC (.PMC)	This file type contains Programmable Machine Controller (PMC) information.
Process (.PR)	This file type contains programs with a process sub type.

Table 9–7. Types of Files (Cont'd)

File Type	Description
System file (.SV)	This file type contains binary files that store default values for system variable, servo parameter data, and mastering data.
Teach pendant program file (.MN)	This file type contains binary files that contain teach pendant instructions for teach pendant programs.
Text file (.TX)	This file type contains text (ASCII) files that contain system-defined or user-defined text.
Variable listing file (.VA)	This file type contains text (ASCII) files that contain the listing of KAREL variables or system variables.
Variable file (.VR)	This file type contains binary files that contain variable data for a KAREL program.
Vision Data file (.VD)	This file type contains binary files that contain Vision Data for Vision.
Vision Data ASCII file (.VDA)	This file type contains text (ASCII) files that contain Vision Data parameters for Vision.
Zip file (.ZIP)	This file type contains multiple types of files in one container file.

Table 9–8 describes how file types are grouped based on their function, and lists example file names for each group. This information can be used to determine files to load or download depending on a functional area of the controller.

Table 9–8. Robot Data File Groups

Robot Data File Group	Description	Example Files
Operating System Files	These files are related to the robot operating system's configuration.	• DIOCFGSV.IO— I/O information • NUMREG.VR— data register information • POSREG.VR— position register information • FRAMEVAR.VR—Frame setup information • SYSFRAME.SV—Tool frames and user frames • SYSVARS.SV— system variable information • SYSSERVO.SV— servo parameter data • SYSMAST.SV— robot mastering data • SYSPASS.SV— password information (if applicable) • SYSFSAC.SV— remote access control information (if applicable) • SYSHOST.SV— Ethernet host table (if applicable) • SYSMACRO.SV— macro table definitions
Supplier Application Files	These files are application files that are pertinent to the manufacturing process. They are provided by the supplier and are common to a number of robots.	• SWCUSTO.VR— FANUC Robotics customized variables • SYSSPOT.SV— Spot welding-specific variables • SYSSEAL.SV— Dispense application-specific variables • MAIN_TCP.VR— Torch Mate data (if applicable) • SET_ATCP.VR— Torch Mate data (if applicable) • OFFSETDT.VR— Torch Mate data (if applicable) • TMMOVETO.VR— Torch Mate data (if applicable) • User-specified variables listed in $FILE_APPBCK
User Application Files	These files are application files that are pertinent to the manufacturing process. However, they are customized by the user and are common to a number of robots.	• User-specified files listed in $FILE_AP2BCK

Table 9–8. Robot Data File Groups (Cont'd)

Robot Data File Group	Description	Example Files
Robot Path Files	These are files that are related to specific robot functions such as robot path and variable data customizable by the robot plant floor operator.	• All teach pendant programs (*.tp)
Robot Specific Data Files	These are robot specific files containing calibration and configuration data, and I/O mapping information.	• SWCUSTO.VR— FANUC Robotics customized variables • SYSSPOT.SV— spot welding specific variables • User-specified files listed in $FILE_AP2BCK (Calibration and configuration data and I/O mapping can be specified.
Error Log	These are robot system error log files.	• ERRALL.LS— Alarm history • ERRACT.LS— Active alarms
Vision Data Files	These are vision process specific data files.	• All vision data files (*.VD)

9.3.2 Altering the FILE Menu View

If you are using an iPendant and are in Single Display Mode, you can alter how files are displayed on the FILE menu. Press NEXT, > and then press F5, [VIEW] when the FILE menu is displayed. You can select between Normal, Name Only, and Wide. See Table 9–9 .

Table 9–9. Altering the FILE MENU [VIEW] Display

[VIEW] Choice	FILE MENU VIEW
Normal	FILE MC:\.* 25/42 20 ABORTIT TP 466981 21 APRG21 TP 686805 22 GETDATA TP 279932 23 HOME_IO TP 123809 24 LISTMENU TP 129862 25 MENUTEST TP 618 26 PRODUCT TX 71 27 VERSION TXT 10 28 * * (all files) 29 * KL (all KAREL source) 30 * CF (all command files) 31 * TX (all text files) 32 * LS (all KAREL listings) 33 * DT (all KAREL data files) 34 * PC (all KAREL p-code) 35 * TP (all TP programs) 36 * MN (all MN programs) 37 * VR (all variable files) 38 * SV (all system files) Press DIR to generate directory

Table 9–9. Altering the FILE MENU [VIEW] Display (Cont'd)

[VIEW] Choice	FILE MENU VIEW
Name Only	FILE MC:*.* 25/NN ABORTIT TP APRG21 TP GETDATA TP PRODUCT TX TESTPRG8901234567890123456789012 LS TESTPRG890123456789012345678901234 56 TXT TESTPRGTESTPRG LS TESTPRGTESTPRG TP VERSION TXT * * (all files) * KL (all KAREL source) * CF (all command files) * TX (all text files) * LS (all KAREL listings) * DT (all KAREL data files) * PC (all KAREL p-code) * TP (all TP programs) * MN (all MN programs) * VR (all variable files) * SV (all system files)

Table 9–9. Altering the FILE MENU [VIEW] Display (Cont'd)

[VIEW] Choice	FILE MENU VIEW
Wide	FILE MC:*.* 25/NN ABORTIT TP 279932 xx-NOV-XX 11:16 APRG21 TP 123809 xx-NOV-XX 11:16 GETDATA TP 129862 xx-NOV-XX 11:16 PRODUCT TX 71 xx-NOV-XX 11:12 TESTPRG8901234567890123456789012 LS 123809 xx-NOV-XX 11:16 TESTPRG8901234567890123456789012 TXT 118 xx-NOV-XX 10:17 TESTPRGTESTPRG LS 729 xx-NOV-XX 01:16 TESTPRGTESTPRG TP 137 xx-NOV-XX 01:16 VERSION TXT 10 xx-NOV-XX 01:14 * * (all files) * KL (all KAREL source) * CF (all command files) * TX (all text files) * LS (all KAREL listings) * DT (all KAREL data files) * PC (all KAREL p-code) * TP (all TP programs) * MN (all MN programs) * VR (all variable files) * SV (all system files)

9.3.3 Generating a Directory of Files

A *directory* is a list of files on a specific storage device. You can display a directory of files on the devices listed in Section 9.1 , "Storage Devices."

Directory Subsets

Some devices contain hundreds of files. You can display a directory of all files, or a *subset* of the files. When you generate a directory of files, you can choose from among the file types listed in Table 9–7 .

Use Procedure 9-14 to generate a directory of files.

Procedure 9-14 Generating a Directory of Files

Steps

1. Set the default device to the device you want:
   a. Press MENUS.
   b. Select FILE.
   c. Press F1, [TYPE].
   d. Select File.
   e. Press F5, [UTIL].
   f. Select Set Device.
   g. Move the cursor to the device you want and press ENTER.
2. Press MENUS.
3. Select FILE.
4. Press F1, [TYPE].
5. Select File.
6. Press F2, [DIR].
7. Select the subset of files you want to display and press ENTER. If you select *.TP to display all teach pendant files, you will see a screen similar to the following.

```
FILE
MC: *.TP
  1 -BCKEDT- TP        105
  2 ABORTIT  TP        106
  3 APRG21   TP         72
  4 GETDATA  TP        179
  5 HOME_IO  TP        241
  6 LISTMENU TP        177
  7 MENUTEST TP        368
  8 MOV_HOME TP        200
  9 MOV_REPR TP        245
 10 OPERMENU TP        177
```

To select another subset of files, press F2, [DIR], and repeat Step 7 .

9.3.4 Backing Up Files

When you back up a file, you save it from controller memory to the default device so that you have a second copy of the file. You can back up program, system, application, diagnostic, and error log files to the default device using the FILE screen.

Note To back up all memory on the controller, use the Controller Backup and Restore function. Refer to Section 9.4 .

Program Files

When you back up program files, all teach pendant program files currently loaded onto controller memory (listed on the SELECT menu) and default motion instruction files (DF_xxxx.DF) will be saved to the default device.

System Files

System files are binary files that store default values for system variables, servo parameter data, and mastering data. They contain information specific to the controller, robot, and software. When you backup system files, all system variable, servo parameter, and mastering data currently on controller memory is saved to the default device. Table 9–10 lists and describes the various kinds of system files.

Note Some system files are application-specific; therefore, the files displayed in Table 9–10 might vary according to your setup.

Table 9–10. System Files

NAME	DESCRIPTION
AWSCHED.SV*	This file contains arc welding schedules.
AWSETUP.SV*	This file contains arc welding setup information.
DIOCFGSV.IO	This file contains I/O configuration information.
NUMREG.VR	This file contains register information.
POSREG.VR	This file contains position register information.
SYSIPS.SV*	This file contains IPS dispenser setup and I/O information.
FRAMEVAR.VR	This file contains frame information.
SYSFRAME.SV	This file contains $MNUTOOL, $MNUFRAME, $MNUTOOLNUM, and $MNUFRAMENUM. These variables were stored in SYSVARS.SV in releases prior to V7.20.
SYSMACRO.SV	This file contains macro command setup information that is created when macro commands are set up. This file must have all supporting macro files loaded before it can be restored.
SYSMAST.SV	This file contains dynamic mastering data that is automatically created when the robot is mastered.

Table 9–10. System Files (Cont'd)

NAME	DESCRIPTION
SYSPASS.SV	This file contains password setup information.
SYSSERVO.SV	This file contains servo parameter data that the robot needs to function. The values in this file are loaded automatically when the controller is turned on.
SYSVARS.SV	This file contains system variable default values for your system. The $STYLE_NAME system variable, which contains style names, is included in SYSVARS.SV. When you back up system files, your style names will be backed up also.
TASCHED.SV *	This file contains TAST or AVC schedule information.
TASETUP.SV *	This file contains TAST or AVC setup information.
THSCHED.SV *	This file contains Touch Sensing schedule information.
THSETUP.SV *	This file contains Touch Sensing setup information.
WVSCHED.SV*	This file contains weave schedule information.
WVSETUP.SV*	This file contains weave setup information.

*** These files appear if you have the option loaded.**

Application Files

When you select "Application," all files listed in the $FILE_APPBCK system variable will be saved. **You should not modify this system variable.** Table 9–11 lists and describes the various kinds of application files.

Table 9–11. Application Files

Name	Description
*.VR	KAREL variable files
*.PC	KAREL program files
*.TX	Dictionary files
FR:*.HTM	HTML web pages on FR: device
FR:*.STM	HTML web pages using *i*Pendant Controls or Server Side Includes on FR: device
FR:*.GIF	GIF image files on FR: device
FR:*.JPG	JPEG image files on FR: device
SYSSEAL.SV*	This file contains sealing setup system variable values for your system.
SYSSLIO.SV*	This file contains sealing I/O setup system variable values for your system. (Note: does not include IPS I/O if equipped with IPS dispenser.)
SYSSLSCH.SV*	This file contains sealing schedule system variable values for your system.

*** These files appear if you are using DispenseTool or SpotTool+ with the Dispense Plug-in.**

Application Teach Pendant Program Files

Application teach pendant program files are teach pendant program files with file type .TP, .DF, or .MN. The names of the application TP files are stored in the system variable $FILE_AP2BCK.

Error Log Files

Error log files are ASCII files that give a snapshot of the errors in the system. They can be backed up to the default device, but cannot be restored or loaded into the controller. Refer Table 9–12 for descriptions of the error log files that are backed up.

Table 9–12. Error Log Files

File	Description	Menu
ERRACT.LS	Contains a snapshot of the active errors in the system	ALARM, [TYPE], Alarm Log, Active menu
ERRALL.LS	Contains a snapshot of the history of errors in the system	ALARM, [TYPE], Alarm Log, Hist menu
ERRAPP.LS	Contains a snapshot of the history of errors in the Application alarm log	ALARM, [TYPE], Appl Log
ERRCOMM.LS	Contains a snapshot of the history of errors in the Communication alarm log	ALARM, [TYPE], Comm Log
ERRCURR.LS	Contains a snapshot of the system configuration and current fault and incidental reporting.	N/A
ERREXT.LS	Contains a snapshot of up to 1000 of the most recent errors in the system.	N/A
ERRHIST.LS	Contains a snapshot of the system configuration and current fault and incidental reporting.	N/A
ERRMOT.LS	Contains a snapshot of the history of errors in the Motion alarm log	ALARM, [TYPE], Motion Log
ERRPWD.LS	Contains a snapshot of the history of errors in the Password alarm log	ALARM, [TYPE], Password Log
ERRSYS.LS	Contains a snapshot of the history of errors in the System alarm log	ALARM, [TYPE], System Log

The information in an error log file follows a specific format, which is shown as follows. The first line is the *error log header* , and subsequent lines are *error entries*.

```
Error Log Header:
```

```
S1:\ERRALL.LS Robot Name PALROB   Time: 17:21:26
08/28/xx
```

The **header** consists of the error log file name, robot host name, the name of the currently selected program or file, and the current system time and date.

```
Error Entry:
255" 10-SEP-xx 10:35 "   SRVO-154 HVAL(CNV-DC)
alarm
(G:1 A:4)" "   SERVO"  act  "
```

Each **error entry** consists of the following:

- Sequence number - internal system number that identifies a particular error in the error log
- Date and time
- Error facility name
- Error code number
- Error code message
- Cause code message, if applicable
- Severity text
- Active/inactive status of the alarm, for ERRALL.LS only - indicates whether the alarm is currently active. "act" indicates that the alarm is currently active. No text indicates that the alarm is not active.

Diagnostic Files

Diagnostic files are ASCII files that provide you with a snapshot of the special diagnostic files on the memory device. Diagnostic files are the files with extension .DG. The names of the diagnostic files are stored in the system variable $FILE_DGBCK. The first element of this system variable is set to summary.dg.

Remaining elements are, by default, uninitialized and you can set these elements to specific diagnostic file names to be backed up.

Vision Data Files

Vision Data files are binary files that store Vision setup parameters for Camera Setup, Camera Calibration, Vision Process and Application Data. When you backup Vision Data files, all Vision Data currently on the controller memory is saved to the default device. The names of the diagnostic files are stored in the system variable $FILE_VISBCK. The first element of this system variable is set to *.vd.

Remaining elements are, by default, uninitialized and you can set these elements to specific vision data file names to be backed up.

Backdate.dt

This file is created and saved on the default device when System files Backup, Application Backup, and All of above Backup is performed. This file contains information about the date and time of the file backup and information about the software version and software options loaded on the controller.

Use Procedure 9-15 to back up system and program files.

Procedure 9-15 Backing Up System and Program Files from the Teach Pendant or CRT/KB

Conditions

- **If you are backing up files to a memory card or USB memory stick device,** make sure that the memory card or memory stick is installed properly. Refer to Section 9.1.5 .

Steps

1. Set the default device to the device you want:
   a. **If you are using the teach pendant,** press MENUS. **If you are using the CRT/KB,** press F10>, MENUS.
   b. Select FILE.
   c. Press F1, [TYPE].
   d. Select File.
   e. Press F5, [UTIL].
   f. Select Set Device.
   g. Move the cursor to the device you want and press ENTER.
2. Press MENUS.
3. Select FILE.
4. Press F1, [TYPE].
5. Select File.
6. Press F4, [BACKUP]. If you do not see [BACKUP], press FCTN and select RESTORE/BACKUP.
7. Select an item that you want to back up from Table 9–13 and press ENTER.

Table 9–13. File Types Listings and Descriptions

FILE TYPE	DESCRIPTION
System Files	This item displays the first system file in the $FILE_SYSBCK system variable.
Teach Pendant Programs	This item displays the first teach pendant program loaded into controller memory.
Application Files	This item displays the first application file in the $FILE_APPBCK system variable.
Application Teach Pendant Programs	This item displays the first application teach pendant program in the $FILE_AP2BCK system variable.
Error Log Files	This item displays the first error log file in the $FILE_ERRBCK system variable.
Diagnostic Files	This item displays the first diagnostic file in the $FILE_DCBCK system variable.
Vision Data Files	This item displays the first vision data file in the $FILE_VISBCK system variable.
All of Above	This item selects all of the various file types for back up. When you select this option, all files on the default device are deleted before the backup is performed. If the destination device is networked, however, the files are not deleted. When the backup is complete, a file called BACKDATE.DT is created that contains the date and time of the backup. Also, at the completion of the backup operation, the FILE menu is displayed, which allows you to generate a directory of the default device by pressing DIR.

8. Choose the backup operation that you would like to perform from Table 9–14 and press the appropriate function key.

Table 9–14. Back Up Operations

IF YOU WANT TO	PRESS	RESULT
Back up the current file	F4, YES	This item backs up the current file then displays the next file in the queue.
Skip to the next file	F5, NO	This item skips the current file and displays the next file in the queue.
Back up all files	F3, ALL	This item backs up all files in the queue. If the file already exists, you will be prompted to either overwrite or skip the file, or cancel backing up that file.
Exit procedure	F2, EXIT	This item cancels the backup operation.

Note For information on backing up controller files, refer to the *Internet Options Setup and Operations Manual* .

Note If you are using the CRT/KB to back up a teach pendant program, and the teach pendant program is being edited by another device (such as the teach pendant), then you will have the option to continue, skip, or cancel as follows:

- F3, Continue, will retry backing up the teach pendant program.
- F4, Skip, will select the next file to be backed up.
- F5, Cancel will abort the rest of the backup.

9.3.5 Loading and Restoring Files to Controller Memory

Loading and restoring files allows you to load a file and all of its relevant data from memory card or USB memory stick into controller memory. You can load files into controller memory from one of the following file devices:

- Flash File Storage disk (FR:)
- RAM disk (RD:) - Not for SpotTool+
- Memory card/ATA Flash File storage
- USB Memory stick device

Loading Files

Typically, you **load** a file from the default device when

- You want to modify a program (teach pendant program file, .TP) that is not currently in controller memory
- You want to execute a file (teach pendant program file, .TP, or KAREL p-code file, .PC)
- You want to load variable information from a memory card or USB memory stick used to install the software initially or from media that has been previously backed up (system variable file, .SV)
- You want to load variable information that has been previously backed up (.VR file)
- You want to load saved I/O configuration (.IO file)
- You want to load saved default motion instructions (.DF file)
- You want to load saved vision data (.VD file)

Loadable Files

Loadable files are those files that can be loaded into controller memory. They are

- Teach pendant program files (.TP)
- KAREL p-code files (.PC)
- System variable files (.SV)
- Mnemonic files (.MN)
- Variable files (.VR)
- I/O configuration files (.IO)

- Default motion instruction files (.DF)
- Vision Data files (.VD)

Note Some system files can be loaded only at Controlled start. Refer to Appendix C for more information.

Note Purging is done automatically at power up when it is determined that a purge is required.

Note You might need to purge Flash ROM after repeated loading of KAREL programs. Each load uses some Flash ROM to store the loaded program. When the program is reloaded, a new portion of Flash ROM is used and the old portion is not released until a purge is performed. Refer to Section 9.3.12 for information on purging.

Only these types of files can be loaded into controller memory. You can load a single file or a group of files. Use Procedure 9-16 to load files using the FILE menu.

Restoring Files

You **restore** files from a memory card or USB memory stick when you have previously backed up the files using BACKUP on the FILE screen (Section 9.3.5). You can restore the following groups of files if you have previously backed them up using BACKUP:

- System files
- Teach pendant programs
- Application files
- Vision Data files

Use Procedure 9-17 to restore BACKUP files using the FILE menu. This procedure will restore all files on the default device that were backed up using the BACKUP command. Refer to Section 9.3.5 for more information on backing up files.



Caution

When you load or restore the file FRAMEVAR.VR, SYSFRAME.SV, SYSVARS.SV, or SYSMAST.SV, make sure the motion configuration (items such as the number of motion groups and extended axes) of your system is the same as the motion configuration of the system on which the FRAMEVAR.VR, SYSFRAME.SV SYSVARS.SV, or SYSMAST.SV files were created. Otherwise, your system might not function properly.

Note Variable files from an R-J3, or R-J3*i*B controller cannot be restored on an R-30*i*A controller.

Note When restoring Vision Data files make sure to have the same Vision options loaded or the Vision process may not work properly.

Procedure 9-16 Loading Files Using the FILE Menu

Conditions

- **If you are loading files from memory card or USB memory stick,** make sure that the memory card or memory stick is installed properly. Refer to Section 9.1.5 .

Steps

1. Set the default device to the device you want:
   a. Press MENUS.
   b. Select FILE.
   c. Press F1, [TYPE].
   d. Select File.
   e. Press F5, [UTIL].
   f. Select Set Device.
   g. Move the cursor to the device you want and press ENTER.
2. Press MENUS.
3. Select FILE.
4. Press F1, [TYPE].
5. Select File.
6. **To load a single file:**
   a. Generate a directory of the default device that contains the file you want to load. Refer to Procedure 9-14 .
   b. Move the cursor to the name of the file you want to load and press F3, LOAD. You will see a screen similar to the following.

```
FILE
MC: *.TP
  1 -BCKEDT- TP        105
  2 ABORTIT  TP        106
  3 APRG21   TP         72
  4 GETDATA  TP        179
  5 HOME_IO  TP        241
  6 LISTMENU TP        177
  7 MENUTEST TP        368
  8 MOV_HOME TP        200
  9 MOV_REPR TP        245
 10 OPERMENU TP        177
Load MC: GETDATA.TP?
```



Caution

When you load or restore the file FRAMEVAR.VR, SYSFRAME.SV, SYSVARS.SV, or SYSMAST.SV, make sure the motion configuration (items such as the number of motion groups and extended axes) of your system is the same as the motion configuration of the system on which the FRAMEVAR.VR, SYSFRAME.SV, SYSVARS.SV, or SYSMAST.SV files were created. Otherwise, your system might not function properly.

Note Variable files from an R-J3, or R-J3*i*B controller cannot be restored on an R-30*i*A controller.

7. Load the file(s):
   - **To load the file(s) you selected,** press F4, YES.
   - **If you do not want to load the file(s) you selected,** press F5, NO.
8. If the program already exists:
   - **To overwrite** , press F3, OVERWRITE.
   - **To skip** the file, press F4, SKIP.
   - **To cancel** , press F5, CANCEL.

Procedure 9-17 Restoring BACKUP Files Using the FILE Menu

Conditions

- **If you are restoring files from a memory card or USB memory stick,** make sure that the memory card or memory stick is installed properly. Refer to Section 9.1.5

Steps

1. Perform a Controlled start.
   a. Press FCTN, select CYCLE POWER, press ENTER, and select YES.
   b. When the BMON screen is displayed on the teach pendant, press and hold the PREV and NEXT keys. You will see the Configuration Menu similar to the following.

```
---------- Configuration Menu ---------
1 Hot start
2 Cold start
3 Controlled start
4 Maintenance
Select >
```

   c. Type 3 and press ENTER.
2. Set the default device to the device you want:
   a. Press MENUS.
   b. Select FILE.
   c. Press F1, [TYPE].
   d. Select File.
   e. Press F5, [UTIL].
   f. Select Set Device.
   g. Move the cursor to the device you want and press ENTER. You will see a screen similar to the following.

```
FILE
MC:\*.*
```

3. Press F4, [RESTOR]. **If F4, [BACKUP] is displayed**, press FCTN and then select RESTORE/BACKUP to display F4, [RESTOR].
4. Select the kind of file you want to restore according to Table 9–15 .

Table 9–15. File Types

FILE TYPE	DESCRIPTION
System Files	This item, when selected, allows you to restore all system files.
Teach Pendant Program Files	This item, when selected, allows you to restore .TP, .DF, and .MN program files.
Application Files	This item, when selected, allows you to restore application files listed in $FILE_APPBCK.
Application Teach Pendant Program Files	This item, when selected, allows you to restore .TP, .DF, and .MN program files, as well as non-program application files.
Vision Data Files	This item, when selected, allows you to restore .VD files.
All Files	This item, when selected, allows you to restore all of the files described above.

You will see a screen similar to the following.

```
Restore from **** (OVRWRT)?
```

Note **** can be either memory card or USB memory stick.

Caution

In the next step, backed up files will be loaded and will overwrite existing files of the same name. Be sure you want to overwrite existing files before you restore them; otherwise, you could lose important data.

Caution

When you load or restore the file FRAMEVAR.VR, SYSFRAME.SV, SYSVARS.SV, or SYSMAST.SV, make sure the motion configuration (items such as the number of motion groups and extended axes) of your system is the same as the motion configuration of the system on which the FRAMEVAR.VR, SYSFRAME.SV, SYSVARS.SV, or SYSMAST.SV files were created. Otherwise, your system might not function properly.

Note Variable files from an R-J3, or R-J3*i*B controller cannot be restored on an R-30*i*A controller.

5. Restore the files:

   - **To continue the restore**, press F4, YES.

- **To cancel the restore**, press F5, NO. You can also cancel the restore at any time by pressing the PREV key.

  Variable files that need to be converted will be done automatically.

6. **If an error occurs during the restore** , the restore will pause.

   - To **skip** the current file and continue restoring the remaining files, press F4, SKIP.
   - To **cancel** the restore from this file on, press F5, CANCEL.

   When the restore has completed, you will see a message reporting the number of files restored.

7. To operate the robot, perform a COLD START:
   a. Press FCTN.
   b. Select START (COLD).

9.3.6 Conversion of PC Files

KAREL programs that have been translated for an older version of the controller can be loaded and converted to a newer version of the controller. The loading of old versions only works on controllers with V7.10 and later. You can back up the KAREL application programs, upgrade the controller software, and then restore these application programs. System and user type mismatches will be converted to the new types if possible.

If a program requires some conversion, there will be a warning in the alarm log and on the status line. If a program fails to load because a type mismatch conversion failed, there will be a warning in the alarm log and on the status line. Conversion messages are written to the file, FR:KRLCONV.LS. Messages are appended to this file and the contents are kept until the next Controlled Start.

The system variable $KAREL_CFG.$CONV_ENABLE enables the conversion and saving of KAREL programs during the load of a KAREL program. If this variable is set to FALSE then KAREL programs are loaded but no conversion of variables or other version differences are completed and no saving of the KAREL program is performed. If this is set to TRUE then either the controller must be in Cold Start or $KAREL_CFG.$CONV_CTRL must also be set true for the conversion and saving to occur. This variable is initialized to TRUE.

The system variable $KAREL_CFG.$CONV_CTRL is only used at Controlled Start. If KAREL program conversion is enabled and the controller is at Controlled Start and this variable is set TRUE, then when a KAREL program is loaded it will be converted and saved as needed. This variable is set automatically by the FILE Menu, KCL and FTP to enable KAREL program conversion whenever a KAREL program is loaded. You can set this variable when it is required to load a user program at Controlled Start by another means. This variable is initialized to FALSE every power up.

9.3.7 Auto Software Update

Auto Update is used to install a revised version of the controller software. The revised version might include new functions or fixes for various issues. It can be used to update versions by a small or large amount. For example it can be used to update from a V7.40P/05 to V7.40P/06 or it can be used to update from V7.40P/04 to V7.40P/24.

The Auto Update must be used on a card that is at least 512 MB in size. This is because the Auto Update makes a Full Backup and an Image Backup of the controller to the memory card.



Caution

Do not attempt a full load from an Auto Update memory card. Otherwise, the following warning message will be displayed:

```
CAUTION: If software is version V7.4
         and Up, then software can't load
         from this PC Card or USB device.
            Quit ? [Y=1/N=else] :
```

If this message is ignored, (N=else), then other more serious alarms and failures will occur.

The INSTAL function key provides the operations to begin an Auto Update. In addition, functions to support the Auto Update operations are also included.

Table 9–16. Auto Update Operations Items

ITEM	DESCRIPTION
Start Auto Update	This function will start the complete Auto Update operation.
Display Backups	This function is used to display the Auto Update backups so that one can be selected for the other operations.

Table 9–16. Auto Update Operations Items (Cont'd)

ITEM	DESCRIPTION
Get Image Backup	This function is used to quickly get the right Image Backup files to the top-level directory. This will allow the user to quickly select and copy a specific Image Backup made by Auto Update.
Clear Backup	This function is used to make room on the Auto Update media. This will allow you to select and clear out a specific Backup made by Auto Update. This deletes the content of the directory and all sub-directories and removes the directory itself.

Refer to the *Software Installation Manual* for more information.

9.3.8 Displaying Text (ASCII) Files

Displaying a text file shows the contents of an ASCII file on the screen. You can display the contents of ASCII files only. Loadable files cannot be displayed.

Displayable (ASCII) Files

Displayable files are ASCII or text files. They are

- KAREL program files (.KL)
- Command files (.CF)
- Text files (.TX)
- Listing files (.LS)
- Data files (.DT)
- Part model files (.ML)

Use Procedure 9-18 to display a text (ASCII) file.

Procedure 9-18 Displaying the Contents of a Text (ASCII) File

Conditions

- **If you are displaying the contents of a file from memory card or USB memory stick,** make sure that the memory card or memory stick is installed properly. Refer to Section 9.1.5 .

Steps

1. Set the default device:
   a. Press MENUS.
   b. Select FILE.
   c. Press F1, [TYPE].
   d. Select File.
   e. Press F5, [UTIL].
   f. Select Set Device.
   g. Move the cursor to the device you want and press ENTER.
2. Press MENUS.
3. Select FILE.
4. Press F1, [TYPE].
5. Select File.
6. Generate a directory that displays the name of the file you want to display (refer to Procedure 9-14).
7. Move the cursor to the name of the ASCII or text file you want to display.
8. Press NEXT, >, and press F3, DISPLAY. The file will be displayed on the screen.
9. To continue displaying, press F4, YES, otherwise press F5, NO.
10. When the file is finished being displayed, press any key to continue.

9.3.9 Copying Files

You can copy a single file or a subset of files from one file device to another. These devices include memory card, USB memory stick, and Flash File Storage disk. Use Procedure 9-19 to copy files.

Procedure 9-19 Copying Files

Conditions

- **If you are copying files to a memory card, or USB memory stick** make sure that the memory card or memory stick is installed properly. Refer to Section 9.1.5 .
- The default device is set correctly. Refer to Procedure 9-1 .

Steps

1. Set the default device:

a. Press MENUS.
b. Select FILE.
c. Press F1, [TYPE].
d. Select File.
e. Press F5, [UTIL].
f. Select Set Device.
g. Move the cursor to the device you want and press ENTER.

2. Press MENUS.
3. Select FILE.
4. Press F1, [TYPE].
5. Select File.
6. **To copy a group of files,** move the cursor to the subset of files you want to copy and press NEXT, >, and then press F2, COPY.

   To copy a single file, generate a directory that displays the file name, move the cursor to the name of the file you want to load, and press NEXT, >, and then press F2, COPY.

   You will see a screen similar to the following.

```
FILE Copy
MC: *.LS
From:          MC: SYSVARS.SV
To Device:     ***
To Directory:
To Filename:   SYSVARS.SV
```

7. Press F4, [CHOICE], to select the device to which the file will be copied.
8. Move the cursor to the device name you want and press ENTER. You will see a screen similar to the following.

```
FILE Copy
From Path:
  MC:\
From Filename:
SYSVARS.SV
To Device:                    [MC:          ]
To Directory:
 \
To Filename:
  SYSVARS.SV
```

9. To change the name of the filename to which the selected file will be copied, press F4, CHANGE. You will see a screen similar to the following.

```
Select
MC:\
From:             MC:\
  SYSVARS.SV
To Device:        FR:
To Directory:     \
To Filename:
  SYSSERVO.SP
```

10. Type the new file name and press ENTER.

11. Change information if necessary:

- **If you want to change any information on the screen,** select the desired field and enter the new information.
- **If all the information on the screen is correct**, press F1, DO_COPY.

12. If file exists,

- To overwrite, press F4, YES, otherwise press F5, NO.

  When the copy is complete, the FILE menu is displayed.

9.3.10 Deleting Files

Deleting a file means permanently removing the file from an external storage device. After you delete a file, you cannot recover it.

Deleting a file from the FILE menu deletes the file from the default device. However, it does not delete it from controller memory. To delete a file from controller memory (the SELECT menu), refer to Procedure 9-12 . Use Procedure 9-20 to delete files.

Procedure 9-20 Deleting Files

Conditions

- **If you are deleting files from a memory card,** make sure that the memory card, or USB memory stick is installed properly. Refer to Section 9.1.5 .
- The default device is set correctly. Refer to Procedure 9-1 .

Steps

1. Set the default device:
   a. Press MENUS.
   b. Select FILE.
   c. Press F1, [TYPE].
   d. Select File.
   e. Press F5, [UTIL].
   f. Select Set Device.
   g. Move the cursor to the device you want and press ENTER.
2. Press MENUS.
3. Select FILE.
4. Press F1, [TYPE].
5. Select File.



Caution

Make sure the default device is set to the device from which you want to delete the file(s); otherwise, you could delete the wrong files.

6. **Generate a directory** of the device from which you want to delete the file. Refer to Procedure 9-14 .

7. **To delete a group of files,** move the cursor to the subset of files you want to delete and press NEXT, >, and then press F1, DELETE.

   To delete a single file, move the cursor to the name of the file you want to delete, and press NEXT, >, and then press F1, DELETE.

8. Delete the file(s):

   - **To delete the specified file(s),** press F4, YES.
   - **If you do not want to delete the specified file(s),** press F5, NO.

9.3.11 Saving Files

Saving files allows you to save individual variable files and other data files to the default device. The following information can be saved using the SAVE function:

Note To save servo parameters and other system files, use the BACKUP function. Refer to Section 9.4 .

- **KAREL variables** for the default program will be saved to a .vr file of the same name as the program. For example, if the default program name is “test” then the KAREL variables are saved to **test.vr** . Either the DATA KAREL Vars or DATA KAREL Posns screens must be displayed to save data to these files.
- **I/O configuration** information will be saved to **diocfgsv.io** . One of the I/O screens must be displayed to save data to this file.
- **Frame setup** information will be saved to **framevar.vr** . One of the SETUP Frame screens must be displayed to save data to this file.
- **Frame data** information will be saved to **sysframe.sv** . One of the SETUP Frame screens must be displayed to save data to this file.
- **Position register** information will be saved to **posreg.vr** . The DATA Position Reg screen must be displayed to save data to this file.
- **Register** information will be saved to **numreg.vr** . The DATA Registers screen must be displayed to save data to this file.
- **Macro setup** information will be saved to **sysmacro.sv** . The SETUP Macro screen must be displayed to save data to this file.
- **System variables** will be saved to **sysvars.sv** . The SYSTEM Variables screen must be displayed to save data to this file.
- **SERVO parameters** information will be saved to **sysservo.sv** . The SYSTEM Servo Param screen must be displayed to save data to the SYSTEM Servo Param file.
- **Mastering information** will be saved to **sysmast.sv** . The SYSTEM Master/Cal screen must be displayed to save data to this file.

- **Teach pendant programs** and **system variables** can be saved as ASCII files. Teach pendant programs are saved to a *.LS file where (*) represents the name of the original teach pendant program. System variables can be saved in ASCII format to a file called **sysvar.ls** .
- **Password information** will be saved to **syspass.sv** . The SETUP Passwords screen must be displayed to save data to this file.
- **SpotTool+ information** will be saved to **sysspot.sv** . Any SpotTool+ screen can be displayed to save data.

> **Caution**
>
> If the file you are saving already exists on the default device, saving the data using the SAVE function will **not** update the file. If you want to save the new file, first delete it from the default device then try saving it again.

Use Procedure 9-21 to save files.

Procedure 9-21 Saving Files to the Default Device

Conditions

- The default device is set correctly. Refer to Procedure 9-1 .
- If you are saving program data, the program you want is the default program.

Steps

1. Display the screen that contains the information you want to save. Refer to Table 9–17 .

Table 9–17. Valid SAVE Function Screens

To Save This Data	Display This Screen	Saved To
KAREL Variables	DATA KAREL VarsorDATA KAREL Posns	*.VR
Most System Variables	SYSTEM Variables	SYSVARS.SV
Mastering Data	SYSTEM Master/Cal	SYSMAST.SV
Macro setup information	SETUP Macro	SYSMACRO.SV
Jog frame setup comment and setup information **Note** Tool frame and user frame information, and the frame transforms are saved only when system variables or sysframe.sv are saved.	SETUP Frame	FRAMEVAR.VR

Table 9–17. Valid SAVE Function Screens (Cont'd)

To Save This Data	Display This Screen	Saved To
Frame variables $MNUTOOL, $MNUFRAME $MNUTOOLNUM, $MNUFRAMNUM	File menu	SYSFRAME.SV
Input/Output current port assignment, mode, and port comment information	I/O (any digital screen)	DIOCFGSV.IO
Register values	DATA Registers	NUMREG.VR
Position register values	DATA Position Reg	POSREG.VR
Servo parameter data	SYSTEM Servo Param Variables	SYSSERVO.SV
Password data	SETUP Passwords	SYSPASS.SV
All SpotTool+ data	Any SpotTool+ screen	SYSSPOT.SV
KAREL Variables	DATA KAREL Vars orDATA KAREL Posns	a file with the same name as the program, followed by .VR
Teach pendant programs and system variables as ASCII	FILE Press >, NEXT. Press F4, [ASCII]	Teach pendant programs: *.LS System Variables: SYSVAR.VA

For example,

To save system variable information:

a. Press MENUS.

b. Select SYSTEM.

c. Press F1, [TYPE].

d. Select Variables. You will see a screen similar to the following.

```
SYSTEM Variables
 1 $ANGTOL               [9] of REAL
 2 $APPLICATION          [3] of STRING [21]
 3 $AP_MAXAX             0
 4 $AP_PLUGGED           2
 5 $AP_TOTALAX           16777216
 6 $AP_USENUM            [32]of BYTE
 7 $ASCII_SAVE           FALSE
 8 $AUTOINIT             2
 9 $BLT                  0
10 $CHECKCONFIG          FALSE
```

2. OR, to save KAREL variable information:
   a. Press SELECT, move the cursor to the program name, and press ENTER.
   b. Press DATA.
   c. Press F1, [TYPE].
   d. Select KAREL Variables.
3. **To save the system variables in ASCII format to the default device:**
   a. Press FCTN.
   b. Select Print. Depending on how large the file is, there may be a wait during the save.
4. **To save teach pendant programs in ASCII format to the default device:**
   a. Press SELECT.
   b. Select a teach pendant program.
   c. Press FCTN.
   d. Select Print. Depending on how large the file is, there may be a wait during the save.
5. **To save all other files in binary format to the default device:**
   a. Repeat Step 1 , and choose the type of file that you want to display.
   b. Press FCTN.
   c. Select SAVE.

9.3.12 Checking File Memory

You can check the amount of memory you are using in the file system using the File Memory screen. You can check the amount of memory you are using in the file system using the File Memory screen. This will show the available memory in Kbytes (KB) for FR: and RD:. Note that the space available on the device FRA: is the same as on FR:.

Use Procedure 9-22 to check file memory.

Table 9–18. FILE Memory Screen Items

ITEM	DESCRIPTION
Total	This item displays the amount of total memory available for both the RD: and FR: devices.
Free	This item displays the amount of free memory available for both the RD: and FR: devices.

Procedure 9-22 Checking File Memory

Steps

1. Press MENUS.
2. Select FILE.
3. Press F1, [TYPE].
4. Select File Memory. You will see a screen similar to the following.

```
FILE Memory
                Total          Free
---------------------------------------
     RD:         64.0 KB        64.0 KB
     FR:        888.0 KB       357.0 KB
```

9.4 AUTOMATIC BACKUP

9.4.1 Overview

An Automatic Backup can be set to run at a predetermined time. Refer to Section 9.4.2 .

9.4.2 Automatic Backup

9.4.2.1 Overview

The Automatic Backup function performs the function of **All backup** in the FILE menu automatically at the following times:

- The specified time (Up to 5 settings).
- When the specified DI is turned on.
- At controller startup. (An interval can be specified.)

There is a special area for Automatic Backup in the controller F-ROM (FRA:). You do not need an external device to use Automatic Backup, but a memory card can also be used.

The Automatic Backup function can manage many backup versions on one device. Even if you back up the wrong programs or settings, you can load the previously backed up version. The number of versions to keep can be set from 1 up to 99. (The default is 2.)

If a memory card is used as the backup device, it must be initialized for Automatic Backup. If the Memory Card is not initialized for Automatic Backup, the Automatic Backup function will not write to the Memory Card. There is no concern about accidentally writing to the wrong memory card.

If the controller power is down during Automatic Backup, backup is performed at the next startup. If you cancel the backup, or if an error occurs in backup, the latest version of backup is recovered automatically. The backup device never has incomplete backup files.

Table 9–19. Automatic Backup Setup Items

ITEM	DESCRIPTION
Automatic Backup	This item indicates whether Automatic Backup is enabled or disabled.
Device	This item indicates what device will be used to store the backups. If it is set to Backup (FRA:), an area in F-ROM will be used. If it is set to Mem Card (MC:), backups will be saved to a memory card.
Sub directory	This item indicates the sub-directory (or path) that will be used on the device. It is a one to eight character name that you choose. **Note** If you are using the Memory Card for backups, it is strongly recommended that a sub-directory is set and used. There is a potential for getting a "Directory Full" alarm if a sub-directory is not used.
Status	This item indicates the current status of the device to be used for Automatic Backup. This will be either Ready for Automatic Backup or Device is not Ready.
Backup time n	This item indicates the times at which the Automatic Backup is to be performed. Up to five settings can be specified. To clear a setting, press F4, CLEAR.
Backup at DI rising	This item allows the Automatic Backup to be performed when the specified digital input is turned on. If the setting is 0, this item is disabled.
Backup at Powerup	This item, when enabled, starts the Automatic Backup when the controller is turned on.
Interval	This item indicates the interval at which backup at powerup will be performed.
Backup in progress	This digital output turns on when the backup is being performed. f the setting is 0, this item is disabled.
Error occurs at backup	This digital output turns on when an error occurs during Automatic Backup. If this setting is 0, this item is disabled.

Table 9–19. Automatic Backup Setup Items (Cont'd)

ITEM	DESCRIPTION
Maximum number of versions	This item indicates the maximum number of Automatic Backup versions that are stored.
Loadable version	This item indicates the date of the currently selected backup version on the device.

Use Section 9.4.2.2 to set up Automatic Backup optional devices and intervals.

9.4.2.2 Setting Up Automatic Backup

Procedure 9-23 Setting up Automatic Backup

1. Press MENUS.
2. Select FILE.
3. Press F1, [TYPE].
4. Select Auto Backup. You will see a screen similar to the following.

```
AUTO BACKUP JOINT 100%                  1/ 13
 1 Automatic Backup:             ENABLE
 2 Device:              Backup (FRA:)
 3   Sub directory             [          ]
     Status: Ready for auto backup
Backup Schedule-------------------------
 4 Backup time 1:                 12:00
 5 Backup time 2:                 23:30
 6 Backup time 3:                 **:**
 7 Backup time 4:                 **:**
 8 Backup time 5:                 **:**
 9 Backup at DI rising:      DI[    0]
10 Backup at Power up:          DISABLE
11   Interval:                7 Day
Status Output---------------------------
12 Backup in progress:        DO[    0]
13 Error occurs at backup:    DO[    0]
Version Management----------------------
14 Maximum number of versions:        1
15 Loadable version:     xx/xx/xx 12:00
```

Note To use a memory card for Automatic Backup, the memory card must be initialized for Automatic Backup. This is to prevent writing to other memory cards. The status of the device is displayed in the Status: line on the teach pendant screen.

5. Initialize the device by doing the following:
   a. If the device is a memory card, it must be formatted. Format the device in the FILE menu. Refer to Section 9.1.6 .
   b. Press F2, INITDEV. The message, “Initialize the device for auto backup?” is displayed.
   c. Press F4, YES. The message, Enter number of versions to keep:” is displayed.
   d. Type the number of incremental backup versions to be stored.

   Note INITDEV deletes all files in the device and creates special files and directories. INITDEV does not format the device. Use the FILE menu to format the device.

6. Set each item as desired. Refer to Table 9–19 .

9.4.2.3 Performing an Automatic Backup

When the specified condition is satisfied, Automatic Backup will be performed. When this occurs, you will see a screen similar to the following.

```
AUTO BACKUP            JOINT 100% 1/9
  Automatic Backup in progress
            Please wait
            Device:  FRA:
To stop backup, press [PREV] key
Saving MC:SYSVARS.SV (3/48)
```

When Automatic Backup is complete, the previously displayed menu will be displayed again.

To cancel the Automatic Backup, press the PREV key. Automatic Backup is cancelled and the previous menu is displayed. No other key besides PREV will be accepted during the Automatic Backup. Even if you are using the teach pendant when Automatic Backup is performed, this menu is displayed and no other key besides PREV will be accepted. Wait for Automatic Backup to complete.

Automatic Backup can work even if a program is running. If the status DO for in progress is set, the specified DO is turned on while this menu is displayed.

If an error occurs, you will see a screen similar to the following.

```
AUTO BACKUP              JOINT 100% 1/9
  Error occurred at Automatic Backup!
Check device (FRA:)
To stop backup, press [PREV] key.
```

When this screen is displayed, the system is not in an Alarm Status and program execution is not stopped. Press F5, RETRY, to perform the backup again. Press PREV to cancel the backup and exit this menu. If the status DO for error occurring is set, the specified DO is turned on while this menu is displayed.

9.4.2.4 Version Management

The Automatic Backup function can keep many backups on one device. The number of versions to keep is set when the device is initialized. You can change the number of versions to keep by setting the item "Maximum number of versions." During backup, if the number of versions exceeds the specified number, the oldest version is deleted automatically.

If the device is FRA:, and the free area of F-ROM is not enough, the oldest version is deleted automatically even if the version number is smaller than the specified number. If the device is MC:, and the free area is not enough to backup, an error occurs at backup. If an error occurs because the device is full, reduce the setting of "Maximum number of versions". The old versions are deleted by this operation and you can continue to back up.

Note The versions that are deleted by decreasing the number of versions are not recovered even if you increase the number of versions later.

Backups are stored in individual sub directories. When Automatic Backup is performed, backup files are saved to the root directory, then these files are copied to the appropriate sub-directory. The FILE menu can access the files only in the root directory, so the latest version of backup can be loaded using the FILE menu. You can also load the older versions. (Refer to Section 9.4.2.5 .)

When "all backup" is performed in the FILE menu to the device that is initialized for Automatic Backup, the files are copied to the appropriate sub directory the same as Automatic Backup.

If Automatic Backup is cancelled, the files of the current backup are deleted and the latest version of backup is copied to the root directory automatically.

9.4.2.5 Restoring the Automatic Backup

Backup files saved by Automatic Backup can be loaded using the FILE menu. Usually the latest version of backup is in the root directory and that version can be loaded. You can load the previous version by the following operation.

Procedure 9-24 Restoring the Automatic Backup

1. Press F4, [CHOICE] on the "Loadable version" item. See the following screen for an example. The menu that contains the backup time of all versions on the device is displayed.

```
1 99/06/1612:00        5 99/06/1412:00
2 99/06/1523:30        6 99/06/1323:30
3 99/06/1512:00        7 99/06/1312:00
4 99/06/1423:30        8 --Next Page--
AUTO BACKUP
Version Management----------------------
13 Maximum number of versions:1
14 Loadable version: xx/06/1612:00
```

2. Select the version to load, then the item “Loadable version” shows the time of the selected version. At this time, the files of the selected version of backup are copied to the root directory.
3. You can load the files of the selected version using the FILE menu.

9.5 TROUBLESHOOTING FILE BACKUP OR AUTOMATIC BACKUP ERRORS

The file system on an R-30*i*A controller only allows a total of 255 files to be saved in the root directory of any file device like a memory card or USB memory stick. The actual number of user files that can be saved is slightly less since the file system actually stores a few files in the root directory. This can inhibit an Automatic backup or File Backup from completing successfully if the number of files in the backup set is very large. This usually will result in errors when doing the Automatic Backup or file backup. Typically you will see the following message displayed on the teach pendant during the Automatic Backup or File backup operations, when this problem occurs:

```
Cannot backup file MC:\filename.xx
```

(where filename.xx is a valid file on the controller), or

```
Cannot backup file UD1:\filename.xx
```

SKIP and CANCEL function keys will also be displayed on the teach pendant, but will not resolve the problem.

If you have a PC available when this occurs, format the memory card or USB memory stick as FAT32 format type, then try the operation again. FAT32 allows more than 255files in the root

directory of a device. You can also format a memory card or memory stick as FAT32 using the controller. Refer to Section 9.1.6 .

If you do not have a PC available when this error occurs, perform Procedure 9-25 .

Procedure 9-25 Resolving Backup or Automatic Backup Errors without using a PC

Condition

- You are performing a FILE backup or Automatic Backup procedure but more than 255 files exist on the backup set.

  Note Be sure that the errors are not occurring because the media to which you are performing the backup is full. If the device is full, delete files from it or obtain a larger device before trying the backup operation again.

- **If you are backing up files to a memory card or USB memory stick,** the memory card or memory stick is installed properly. Refer to Procedure 9-4 .
- The default device is set correctly. Refer to Section 9.1.3 .

1. Create a sub-directory on the memory device with an appropriate name (such as BACKUP). This can be done from the FILE Menu, on the teach pendant as follows:
   a. Press MENUS.
   b. Select FILE.
   c. Press [UTIL].
   d. Choose Set Device.
   e. Select either Memory Card (MC:) or USB Memory Stick (UD1:).
   f. Create a sub-directory, called BACKUP for example, by pressing [UTIL] again and then by choosing Make Dir from the menu.
   g. Type the directory name.
2. **If you are using Automatic Backup,** set the system variable $AUTOBACKUP.$DEVICE to MC:\BACKUP\ or UD1:\BACKUP from the system variable screen by doing the following:
   a. Press MENUS.
   b. Select SYSTEM.
   c. Select Variables.
   d. Move the cursor to $AUTOBACKUP.
   e. Press DETAIL.
   f. Enter MC:\BACKUP\ if you are using an memory card or UD1:\BACKUP\ if you are using a USB Memory Stick.
   g. Press MENUS.

h. Select FILE.

i. Press F1, [TYPE].

j. Select Auto Backup.

k. Select [INIT_DEV]. The memory card or USB Memory stick is now prepared for Automatic Backup of more than 255 files.

3. **If you are using Backup, All of the Above,** and if you have just completed Step 1 then the directory that you created will already be selected and you can skip this step. If not, then perform the following operations to select the directory:

   a. Press MENUS.

   b. Select FILE.

   c. Press [DIR].

   d. Select Directories or *.*.

   e. Move the cursor to the desired directory and then press ENTER.

4. Complete the backup by selecting [BACKUP] on the FILE screen.

5. Select All of the above from the menu. The backup of all of the files should complete successfully.

9.6 IMAGE BACKUP

9.6.1 Overview

The Image Backup feature allows you to make an image of the F-ROM and S-RAM controller memory modules. The image is stored in a number of files (*.img) on a selected destination device.

Image Backup is available from the FILE menu when the system is in Cold start or Controlled start mode. After Image Backup is selected from the menu, the actual backup occurs the next time the controller is powered up.

The image created can later be used to restore the controller memory. The restore is also done during power up. If F1 and F2 are pressed while the controller is powering up, a menu is displayed that allows you to restore a previously saved image.

The following details must be considered when you use Image Backup:

- When TFTP: is selected, files are stored in the root directory of the device. The relationship between the Mac address and the directory should be configured on the server side.
- The FILE menu might not be automatically displayed after the image backup if another function forces a different menu on powerup.

- The automatic software update function precedes this function. Automatic software update may start software installation automatically at next startup even if the start up was caused by the image backup function. Software installation will occur and image backup will not be performed.
- If the F1 and F5 keys are pressed when the controller is turned on, image backup does not begin and the BOOT MONITOR menu is displayed. Image backup can be started by cycling power (FCTN:CYCLE POWER).
- If the F1 and F2 keys are pressed when the controller is turned on, the image backup does not begin and the restore menu (Section 9.6.3) is displayed instead. Image backup can be started by cycling power (FCTN:CYCLE POWER).



Caution

Do not turn off power while the image backup is occurring. Otherwise, the backup might not be performed correctly and you could lose data.



Caution

Do not use a USB memory stick at BMON for image backup/restore. Image backup/restore can only be performed using a memory card or over Ethernet.

9.6.2 Performing an Image Backup

Procedure 9-26 Performing an Image Backup

Conditions

- The controller is powered up in Controlled start or Cold start mode.
- A memory card is installed, a USB memory stick is installed, or TFTP is set up.

1. Press MENUS.
2. Select FILE.
3. Press F4, [BACKUP].

```
FILE                          JOINT  10 %
MC:\*.*                              1/25
  1
  2 *        KL  (a------------------
  3 *        CF  (a|1 System files  |
  4 *        TX  (a|2 TP program    |
  5 *        LS  (a|3 Application   |
  6 *        DT  (a|4 Applic.-TP    |
  7 *        PC  (a|5 Error log     |
  8 *        TP  (a|6 Diagnostic    |
  9 *        MN  (a|7 All of above  |
 10 *        VR  (a|8 Image backup  |
Press DIR to generate ----------+     +-
```

4. If the teach pendant is disabled when Image backup is selected, the following prompt is displayed.

```
Enable TP for image backup.
```

If not, the destination device selection menu is displayed as shown below.

```
                              JOINT 10%
 1 Mem Card(MC: )
 2 Ethernet (TFTP: )
 3 USB (UD1:)FILE
MC:\*.*                               1/25
  1
  2 *        KL  (a------------------
  3 *        CF  (a|1 System files  |
  4 *        TX  (a|2 TP program    |
  5 *        LS  (a|3 Application   |
  6 *        DT  (a|4 Applic.-TP    |
```

5. Image backup files will be stored in the root directory of the selected device. If the device is MC: or UD1: and it already includes image files (*.IMG), the following confirmation message is displayed:

```
Remove existing IMG files?
                           Yes    No
```

If the destination is TFTP:, this confirmation is always displayed.

Note If destination is TFTP:, *.IMG files are always overwritten.

6. If you select No, the image backup will be aborted. Otherwise, you must cycle power on the controller:

- If the system is in Cold start mode, power can be automatically cycled. The following prompt will be displayed:

```
Cycle power?
                          OK    Cancel
```

- If the system is in Controlled start mode, power must be manually cycled. The following prompt will be displayed.

```
Cycle power is needed. OK?
                          OK    Cancel
```

7. Press OK. If the system was in Cold start mode, the controller will be automatically turned off and then on again. If the system is in Controlled start mode, the following message is displayed:

```
 Please cycle power
```

a. Press FCTN.

b. Select CYCLE POWER.

8. On power up, the image backup starts automatically. A screen similar to the following is displayed:

```
WRITING MC:\FROM00.IMG
WRITING MC:\FROM01.IMG
……
WRITING MC:\SRAM02.IMG
DONE!
```

Caution

If the destination device is MC:, do not remove the memory card during the image backup.

Note If the destination device is TFTP, preparation for TFTP image backup must be completed before cycling power. Make sure communication between the robot controller and server is established during image backup.

9. After image backup is completed, or if an error occurs, the controller completes its usual startup sequence. When the startup is complete, one of the following screens is displayed:

```
FILE                          JOINT  10 %
MC:\
  *.*                             1/25
Image backup completed successfully.
                           OK
```

```
FILE                          JOINT  10 %
MC:\
  *.*                              1/25
Image backup failed.
                           OK
```

10. When OK is pressed, the file screen is automatically displayed.

11. If the backup fails, the alarm SYST-223 “Image backup failed(0x%x)” is posted in the alarm log. (0x%x) shows the cause of the failure.

9.6.3 Restoring an Image Backup

Procedure 9-27 Restoring an Image Backup

Conditions

- The controller is powered up in Controlled start or Cold start mode.
- A memory card is installed, or TFTP is set up. Refer to Procedure 9-4 for more information.

1. Press F1 and F2 while turning on the power disconnect circuit breaker. The following screen is displayed.

```
Base System Version V7.xxxx (FRNA)
Initializing file devices ... done.
******* BMON MENU *******
1.    Configuration Menu
2.    All software installation (MC:)
3.    INIT start
4.    Controller backup/restore
5.    Hardware diagnosis
6.    Maintenance
7.    All software installation (ETHERNET)
8.    All software installation (USB)
Select:
```

2. Release all of the keys.
3. Select Controller backup/restore and press ENTER. You will see a screen similar to the following.

```
**** BACKUP/RESTORE MENU ****
0.   Return To Main Menu
1.   Emergency Backup
2.   Backup Controller As Images
3.   Restore Controller Images
5.   Bootstrap To Cfg Menu
```

4. Select Restore Controller as Images and press ENTER. You will see a screen similar to the following.

```
**** BACKUP/RESTORE MENU ****
** Device selection **
1. Memory card (MC:)
2. Ethernet (TFTP:)
3. USB (UD1:)
```

5. Select Memory card (MC:) or USB (UD1:) and press ENTER. You will see a screen similar to the following.

```
***** RESTORE Controller Images *****
 Current module size:
    FROM: 32Mb
    SRAM: 3Mb
 CAUTION: You SHOULD have image files
   from the same size of FROM/SRAM.
   If you don't, this operation causes
   fatal damage to this controller.
   Are you ready ? [Y=1/N=else] :
```



Caution

The following operation erases all of SRAM and FROM. Be sure you are prepared to erase all of SRAM and FROM before you execute in Step 6 .



Caution

Do not turn off the controller until the Image Restore is complete. Turning off controller power during the startup sequence might cause another image backup to occur when the power is turned on again.

Do not perform an image restore if the FROM/SRAM module is NOT on the main board.

6. **If you want to continue**, press 1 and then press ENTER.

   To cancel, press 0 and then press ENTER.

   Restoring will take several minutes (approximately 3–6 minutes). You will see messages displayed on the screen stating that SRAM and FROM are being cleared and that files are being loaded. You will see a screen similar to the following.

```
Reading MC:\FROM25.IMG  ... Done
Reading MC:\FROM26.IMG  ... Done
Reading MC:\FROM27.IMG  ... Done
Reading MC:\FROM28.IMG  ... Done
Reading MC:\FROM29.IMG  ... Done
Reading MC:\FROM30.IMG  ... Done
Reading MC:\FROM31.IMG  ... Done
Reading MC:\SRAM00.IMG  ... Done
Reading MC:\SRAM01.IMG  ... Done
 -- Restore complete --
Press ENTER to return >
```

7. When all of the files have been loaded, the controller will display the Main BMON menu.
8. Select Configuration Menu, and then select Cold Start to perform a Cold start of the controller.

9.7 ASCII UPLOAD

9.7.1 Overview

When robots are equipped with the ASCII upload option, you can load teach pendant programs that are saved as ASCII files (.LS) directly onto the robot. ASCII teach pendant programs can be stored offline and manipulated without the specialized software required for handling binary programs. You can use general purpose text editors such as those used for email to do the following.

- Archive programs offline for code reuse in the same or different robots.
- Examine human-readable hardcopy programs away from a robot.
- Edit programs away from a robot.
- Generate programs using standard text-based scripting tools such as Perl.
- Use less expensive robot simulators than those limited to binary support.

Robots equipped with the ASCII Upload option can read text-based, programs that were printed by the same or a compatibly configured robot.

The ASCII Upload expects the ASCII file to be of the format described in Section 9.7.5 . This format is the same as that printed out by the controller.



Warning

If data in the ASCII file is incorrect or created without taking into account the robot and its location in the workcell, the ASCII Upload might produce a teach pendant program which, when loaded on a robot, could cause unexpected motion. This could result in damage to equipment and injury to personnel.

Always step through the teach pendant program on the robot with motion locked, then run through the program with low speed and your hand on the teach pendant EMERGENCY STOP button. Refer to the "Testing and Running Production" chapter in the application-specific *Setup and Operations Manual* for more information. Otherwise, you could injure personnel or damage equipment.

9.7.2 Loading an ASCII Teach Pendant Program From the Teach Pendant

Use Procedure 9-28 when you load an ASCII teach pendant program and when you load an ASCII teach pendant program with a different filename.

Note The programs you are loading must not be SELECTED or the load will fail with the message "Memo-015 program already exists".

Procedure 9-28 Loading an ASCII Teach Pendant Program

Conditions

- The memory card must be connected properly.
- The program you want is on the memory card.
- The program features are compatible with the controller's configuration.

Steps

1. Press MENUS.
2. Select FILE.
3. Set the default device to the memory card.
4. Select the desired filename.
   a. Press F2, [DIR] and move the cursor to *.LS and press ENTER. You will see a screen similar to the following.

```
FILE
 MC:*.LS
 1 TESTSPOT LS          16541
 2*         *          (all files)
 3*         KL         (all KAREL sources)
 4*         CF         (all command files)
 5*         TX         (all text files)
 6*         LS         (all KAREL listings)
 7*         DT         (all KAREL data files)
 8*         PC         (all KAREL p-code)
 9*         TP         (all TP programs)
10*         MN         (all MN programs)
```

5. Load the program.
   a. Move the cursor to the desired program and press F3, LOAD.

b. Press YES. You will see a screen similar to the following.

```
FILE
 MC:*
 1 TESTSPOT  LS          16541
 2*          *          (all files)
 3*          KL         (all KAREL sources)
 4*          CF         (all command files)
 5*          TX         (all text files)
 6*          LS         (all KAREL listings)
 7*          DT         (all KAREL data files)
 8*          PC         (all KAREL p-code)
 9*          TP         (all TP programs)
10*          MN         (all MN programs)
Loaded MC:TESTSPOT.LS
```

Note You will see the TESTSPOT.TP file in the SELECT menu.

Note If there are errors, you will not be able to edit the ASCII program file on the controller. You will need to edit the ASCII program file with a text editor on another computer.

Note In Step 6 through Step 8 you will be loading an ASCII teach pendant program with a different filename (ARCFILE.LS to ARCPROG.LS).

6. Set the default device to the memory card.

7. Select the desired filename.

a. Press F2, [DIR] and move the cursor to *.LS and press ENTER. You will see a screen similar to the following.

```
FILE
 MC:*.LS
 1 ARCFILE  LS          16541
 2*          *          (all files)
 3*          KL         (all KAREL sources)
 4*          CF         (all command files)
 5*          TX         (all text files)
 6*          LS         (all KAREL listings)
 7*          DT         (all KAREL data files)
 8*          PC         (all KAREL p-code)
 9*          TP         (all TP programs)
10*          MN         (all MN programs)
```

b. Move the cursor to the desired program.

8. Load the program from the file.

a. Press NEXT, > to scroll to the remainder of the function keys and then press F2, COPY. You will see a screen similar to the following.

```
FILE Copy
MC:*.LS
From:              MC:
  ARCFILE.LS
To Device: ***
To Directory:
To Filename:
  ARCFILE.LS
```

b. Press F4, [CHOICE].

c. Move the cursor to Mem Device (MD:) and press ENTER.

d. Press F4, CHANGE. You will see a screen similar to the following.

```
FILE Copy
From                MC:
  ARCFILE.LS
To Device:          MD:
To Directory:
To Filename:
  ARCFILE.LS
```

Note The file name you enter must match the program name in the source file that defines the */PROG* selection or the load will fail.

e. Type the program name with a .LS file extension.

f. Press ENTER. You will see a screen similar to the following.

```
FILE Copy
MC:*.LS
From:              MC:
  ARCFILE.LS
To Device:         MD:
To Directory:
To Filename:
  ARCPROG.LS
```

g. Press F1, DO_COPY. You will see a screen similar to the following.

```
FILE
 MC:*.LS
 1 ARCFILE  LS         16541
 2*         *         (all files)
 3*         KL        (all KAREL sources)
 4*         CF        (all command files)
 5*         TX        (all text files)
 6*         LS        (all KAREL listings)
 7*         DT        (all KAREL data files)
 8*         PC        (all KAREL p-code)
 9*         TP        (all TP programs)
10*         MN        (all MN programs)
Copied MC:ARCFILE.LS to MD:ARCPROG.LS
```

9.7.3 Loading an ASCII Teach Pendant Program from KCL

ASCII teach pendant programs can be loaded from KCL using the COPY command. For the file to successfully load, you must specify the destination device as MD:, the file extension must be .LS, and the filename must match the program name defined in the /PROG section. The following are examples:

- To load as ASCII teach pendant program: **KCL>CopyMC: TESTSPOT.LS to MD:**
- To load as ASCII teach pendant program with a different filename: *KCL>Copy MC:ARCFILE.LST to MD:ARCFILE.LS*

9.7.4 Viewing ASCII Upload Errors

When ASCII Upload detects a syntax error in the source file, it posts warning alarms indicating the location of the errors and terminates the load. The alarms can be viewed using the teach pendant ALARM HISTORY screen. The Alarm History screen allows you to display the items in Table 9–20 and perform the operation in Table 9–21 .

Table 9–20. Viewing ASCII Upload Error Screen Items

ITEM	DESCRIPTION
ASBN Alarms	The facility code ASBN are alarms generated by the ASCII Upload option.

Table 9–21. Viewing ASCII Upload Error Screen Operations

ITEM	DESCRIPTION
[TYPE]	Press this key to access various application-specific options.
HIST	Press this key to enter the alarm history screen.
ACTIVE	Press this key to access the active alarms.
CLEAR	Press this key to clear an ASCII Upload error.
HELP	Press this key to access detailed information on a particular ASCII Upload error.

Procedure 9-29 Viewing ASCII Upload Errors

Conditions

- A .LS file was loaded and it failed.

In Procedure 9-29 you will be viewing Alarm History log.

Steps

1. Press MENUS and select ALARMS. You will see a screen similar to the following.

```
Alarm : Active
There are no active alarms
Press F3(HIST) to enter alarm history screen.
```

2. Press F3, HIST and move your cursor to ASBN-008 alarm. You will see a screen similar to the following.

```
Alarm:Hist

    1 ASBN-090 Undefined macro
    2 ASBN-090 Undefined macro
    3 ASBN-008 file 'MD:MENUTEST.LS'
    4 ASBN-009 on line 22, column 13
```

Note You will see one or two ASBN-alarms followed by ASBN-008. The alarm will give you the file name and there will be one or more ASBN-009 alarms showing the line and column.

3. Move the cursor to ASBN 009 and press F5, HELP to view the DETAIL Alarm screen. The error shows the line and column after the offending word. The cause code shows an ASBN error indicating what is wrong.

```
DETAIL Alarm
ASBN-009 on line 23, column 14
ASBN-092 Undefined instruction
06-SEP-01 9:31
WARN
 Alarm:Hist

1 ASBN-090 Undefined macro
2 ASBN-008 file MD:MENUTEST.LS
3 ASBN-009 on line 23, column 14
```

Note If there are errors, you will not be able to edit the ASCII program file on the controller. You will need to edit the ASCII program name with a text editor on another computer.

9.7.5 Example ASCII File

This section details the syntax of an ASCII teach pendant program.

See Figure 9–9 for an example ASCII file. It contains examples of the following possible sections of an ASCII file:

- /PROG - Program name
- /ATTR - File attribute data
- /APPL - Tool application data
- /MN - Teach Pendant instructions
- /POS - Position data

- /END - End of file

Figure 9–9. Example ASCII File

```
/PROG  TEST                                  ------- Program data
/ATTR                                        ------+
OWNER           = MNEDITOR;                        | File
COMMENT         = "TEST";                          | Attribute
PROG_SIZE       = 366;                             | Data
CREATE          = DATE xx-11-13  TIME 07:10:05;    |
MODIFIED        = DATE xx-11-13  TIME 07:10:05;    |
FILE_NAME       = ;                                |
VERSION         = 0;                               |
LINE_COUNT      = 2;                               |
MEMORY_SIZE     = 870;                             |
PROTECT         = READ_WRITE;                      |
TCD:  STACK_SIZE        = 0,                       |
      TASK_PRIORITY     = 50,                      |
      TIME_SLICE        = 0,                       |
      BUSY_LAMP_OFF     = 0,                       |
      ABORT_REQUEST     = 0,                       |
      PAUSE_REQUEST     = 0;                       |
DEFAULT_GROUP   = 1,*,*,*,*;                       |
CONTROL_CODE    = 00000000 00000000;       ------+
/APPL                                        ------+
PAINT_PROCESS ;                                    |Application
DEFAULT_USER_FRAME : 1 ;                           |Data
DEFAULT_TOOL_FRAME : 1 ;                           |
START_DELAY        : 0 ;                           |
TRACKING_PROCESS   : NO ;                  ------+
/MN                                           ------+
   1:J P[1] 100% FINE ;                            |Instructions
   2:J P[2] 100% FINE Gun=ON    ;          ------+
/POS                                         ------- Start of ----
P[1]{                                                Position Data
   GP1:
        UF : 1, UT : 1,                    CONFIG : 'N, -1, , 0',
        X =  1584.56  mm,  Y =   299.91  mm,  Z =  -300.03  mm,
        W =  -120.00 deg,  P =     0.00 deg,  R =  -115.21 deg
};
P[2]{
   GP1:
       UF : 0, UT : 1,                     CONFIG : 'N, -1, , 0',
        X =  1584.56  mm,  Y =   299.91  mm,  Z =  -300.03  mm,
        W =  -120.00 deg,  P =     0.00 deg,  R =  -115.21 deg
};
/END
```

Chapter 10

ADVANCED FUNCTIONS

Contents

Chapter 10	ADVANCED FUNCTIONS	10–1
10.1	ADAPTIVE PROCESS CONTROL	10–7
10.1.1	Overview	10–7
10.1.2	Setting Up Input Parameters	10–9
10.1.3	Setting Up Output Parameters	10–10
10.1.4	Setting Up Map Functions	10–13
10.1.5	Setting Up Options	10–15
10.1.6	Adding the APC Instructions To a Program	10–16
10.2	ANGLE ENTRY SHIFT FUNCTION (OPTION NOT AVAILABLE IN NORTH AMERICA)	10–17
10.3	AUTO SINGULARITY AVOIDANCE	10–23
10.3.1	Overview	10–23
10.3.2	How it Works	10–25
10.3.3	Compatibility	10–26
10.3.4	Limitations	10–26
10.3.5	Cautions	10–27
10.3.6	How to Use Auto Singularity Avoidance	10–30
10.3.7	System Variable Setup	10–37
10.3.8	Troubleshooting	10–37
10.4	AUTOMATIC COLLISION RECOVERY	10–40
10.4.1	Overview	10–40
10.4.2	Enabling and Disabling Automatic Collision Recovery	10–41
10.4.3	Automatic Collision Recovery Setup	10–41
10.4.4	Mode 1 of Automatic Collision Recovery	10–43
10.4.5	Mode 2 of Automatic Collision Recovery	10–45
10.4.6	Limitations	10–46
10.5	BASIC PROCESS AXES (OPTION)	10–47
10.5.1	Overview	10–47
10.5.2	Setup	10–48
10.5.3	TP Instructions	10–51

10.5.4 PROCESS AXES STATUS (OPTION) 10–54
10.5.5 Torque Monitor Function 10–55

10.6 BRAKE CHECK FUNCTION 10–56
10.6.1 Overview 10–56
10.6.2 Setup 10–56
10.6.3 Start Brake Check 10–57
10.6.4 Result of Brake Check 10–58
10.6.5 Limitations 10–58
10.6.6 Caution 10–59

10.7 BUMP BOX 10–59
10.7.1 Overview 10–59
10.7.2 Setting Up Bump Box Schedules 10–60
10.7.3 Setting Up Bump Box I/O 10–67
10.7.4 BumpBox with Coordinated Motion - Overview 10–69
10.7.5 Bump Frames for Normal and Coordinated Motion 10–73

10.8 CENTER FINDER 10–82
10.8.1 Overview 10–82
10.8.2 Using Center Finder 10–82
10.8.3 Troubleshooting 10–91

10.9 COLLISION GUARD 10–91
10.9.1 Overview 10–91
10.9.2 Limitations 10–92
10.9.3 Falsely Detected Collisions 10–92
10.9.4 Setup 10–93
10.9.5 Programmed Motion 10–95

10.10 COLLISION SKIP 10–98
10.10.1 Overview 10–98
10.10.2 Requirements 10–98
10.10.3 Operation 10–99

10.11 CONDITION MONITOR FUNCTION 10–103
10.11.1 Overview 10–103
10.11.2 Monitors 10–104
10.11.3 Monitor State 10–105
10.11.4 Monitor Instructions 10–106
10.11.5 Condition Handler Program 10–107
10.11.6 Conditions 10–107
10.11.7 Condition Menu 10–108
10.11.8 Restrictions 10–111

10.12 COORDINATES OFFSET FUNCTION 10–117
10.12.1 Using the Coordinates Offset Function 10–117
10.12.2 Tool Frame Offset Function 10–121
10.12.3 User Frame Offset Function 10–125

10.13 CYCLE TIME LOGGING AND DISPLAY 10–129
10.13.1 Overview 10–129
10.13.2 Modes 10–131
10.13.3 Tracking Cycle Time 10–136

10.14 DATA MONITOR 10–139
10.14.1 Overview 10–139
10.14.2 Data Monitor Setup 10–142
10.14.3 Data Monitor Schedule 10–148
10.14.4 Data Monitor Programming 10–153

10.15 DIAGNOSTIC VIDEO MONITORING 10–154

10.15.1 Overview 10–154
10.15.2 DVM Setup 10–155
10.15.3 Controlling DVM Execution 10–161
10.15.4 Displaying DVM and DVM Logs 10–162

10.16 DISTANCE BEFORE 10–165
10.16.1 Overview 10–165
10.16.2 Specification 10–165
10.16.3 Program Instruction 10–167
10.16.4 Distance Value 10–167
10.16.5 Action 10–169
10.16.6 Changing the Trigger Condition 10–170
10.16.7 Alarms Posted When Distance Before is Not Triggered 10–172
10.16.8 Single Step Execution 10–173
10.16.9 Hold and Resume 10–173
10.16.10 Resume After Jogging 10–176
10.16.11 Power Failure Recovery 10–177
10.16.12 Adding the Distance Before Motion Option (Procedure) 10–177
10.16.13 Cautions and Limitations 10–180
10.16.14 System Variables 10–181
10.16.15 Error Codes 10–182

10.17 EASY NORMAL UTILITY (formerly known as Auto Normal Utility) 10–183
10.17.1 Overview 10–183
10.17.2 Using the Easy Normal Utility 10–184
10.17.3 Setup 10–187
10.17.4 Execution 10–189

10.18 ENHANCED MIRROR IMAGE (OPTION) 10–190

10.19 ERROR RECOVERY (OPTION) 10–209
10.19.1 Overview 10–209
10.19.2 Features 10–210
10.19.3 Limitations 10–211
10.19.4 I/O Interface 10–212
10.19.5 Setup 10–214
10.19.6 Programming 10–220
10.19.7 Testing 10–224
10.19.8 Error Recovery Manual Function 10–224
10.19.9 I/O Timing Sequence 10–227

10.20 EXECUTING MULTIPLE PROGRAMS (MULTI-TASKING) 10–232
10.20.1 Overview 10–232
10.20.2 Guidelines 10–233
10.20.3 Synchronizing the Execution of Multiple Programs 10–234
10.20.4 Effect of Multi-Tasking on Dedicated I/O Signals 10–234
10.20.5 Standard Operator Panel (SOP) Cycle Start Execution 10–234
10.20.6 Program Number Select (PNS) Execution 10–235
10.20.7 RUN Program Instruction Execution 10–237
10.20.8 Single Step Program Execution 10–237

10.21 EXTENDED USER FRAMES 10–239
10.21.1 Configuring the Number of Frames 10–240
10.21.2 Effect on the Jog Menu 10–240
10.21.3 Programming with Extended User Frames 10–241

10.22 GRAVITY COMPENSATION 10–242
10.22.1 System Variables 10–242
10.22.2 MOTION Screen 10–243

10.23 GROUP MASK EXCHANGE 10–244
10.23.1 Overview 10–244
10.23.2 Setting Up Group Mask Exchange 10–245
10.23.3 Troubleshooting Group Mask Exchange 10–246
10.24 HIGH SPEED INPUT FUNCTIONS 10–248
10.24.1 Overview 10–248
10.24.2 Frequency Interface 10–248
10.24.3 High Speed Scanning (for Line Tracking Part Detect). 10–248
10.24.4 Position Snap & Map 10–249
10.25 LIMIT SET 10–249
10.26 MENU UTILITY (OPTION) 10–252
10.26.1 Overview 10–252
10.26.2 Prompt Box Message Menu 10–253
10.26.3 Prompt Box Yes/No Menu 10–259
10.26.4 List Menu 10–265
10.26.5 Status Menu 10–273
10.26.6 Operator Entry Menu 10–285
10.26.7 Using Your Teach Pendant Menus 10–294
10.27 MIRROR IMAGE UTILITY 10–295
10.28 MONITORING PROGRAMS 10–311
10.28.1 Montioring Teach Pendant Programs 10–311
10.29 MOTION GROUP DO OUTPUT FUNCTION (MULTIPLE MOTION GROUP OPTION) 10–313
10.29.1 Overview 10–313
10.29.2 Restrictions 10–314
10.29.3 Operations 10–314
10.29.4 Setup 10–315
10.29.5 Subprogram and Multi-tasking Execution 10–316
10.30 MOTION START DELAY DETECTION FUNCTION 10–317
10.30.1 Overview 10–317
10.30.2 Restrictions 10–317
10.30.3 Detection Information and System Variables 10–318
10.30.4 Motion Start Delay Detection Function Example 10–319
10.31 OFFLINE SINGULARITY AVOIDANCE (Formerly Singularity Avoidance) 10–320
10.31.1 Overview 10–320
10.31.2 How it Works 10–321
10.31.3 Compatibility 10–323
10.31.4 Limitations 10–324
10.31.5 Procedure 10–324
10.31.6 Troubleshooting 10–329
10.32 OPERATION LOG BOOK [OPTION] 10–330
10.32.1 Overview 10–330
10.32.2 Recorded Events 10–333
10.32.3 Operations 10–339
10.33 ORIGINAL PATH RESUME 10–344
10.34 PAYLOAD CONFIRM 10–348
10.34.1 Overview 10–348
10.34.2 Payload Setting and Payload Over Confirm 10–349
10.35 POSITION REGISTER LOOK-AHEAD EXECUTION FUNCTION 10–353
10.35.1 Overview 10–353

10.35.2 Program Instructions 10–354
10.35.3 Program Example 10–355
10.35.4 Execution 10–356
10.36 PROGRAM SHIFT UTILITY 10–356
10.37 REFERENCE POSITION UTILITY 10–368
10.38 RESUME OFFSET 10–372
10.38.1 Limitations of Resume Offset 10–372
10.38.2 Constant Path Resume Offset 10–373
10.38.3 Limitations of Constant Path Resume Offset 10–373
10.38.4 Non-Constant Path Resume Offset 10–374
10.38.5 Limitations of non-Constant Path Resume Offset 10–374
10.38.6 Selecting Resume Offset Type 10–375
10.38.7 Resume Offset Setup Menu 10–375
10.38.8 System Variables 10–377
10.39 ROBOT-PC COMMUNICATIONS MACROS 10–378
10.39.1 Overview 10–378
10.39.2 Send Event ( ) 10–378
10.39.3 Send Data( ) 10–380
10.39.4 Send SysVar( ) 10–381
10.39.5 Get Data( ) 10–383
10.39.6 Request Menu( ) 10–384
10.40 SAFETY SPACE 10–385
10.40.1 Overview 10–385
10.40.2 Requirements 10–385
10.40.3 Operation 10–386
10.41 SERVO TOOL (Option) 10–387
10.41.1 Benefits 10–387
10.41.2 Limitations 10–388
10.41.3 Program Instruction 10–388
10.41.4 Setup 10–390
10.41.5 Detection 10–391
10.42 SHAPE GENERATION II (OPTION) 10–393
10.42.1 Overview 10–393
10.42.2 Quick Start Hints 10–394
10.42.3 Shape Setup 10–395
10.42.4 Setting Up Default Shape Data 10–397
10.42.5 Shape-Specific Shape Data for Standard Shapes 10–400
10.42.6 Setting Up Custom Shape Data 10–413
10.42.7 Generating a Shape Program from UTILITIES Shape Gen 10–416
10.42.8 Servo Schedules 10–424
10.42.9 Tuning Guidelines 10–426
10.43 SIMPLE TEACH SUITE 10–429
10.43.1 Overview 10–429
10.43.2 Auto Path Smoothing 10–430
10.44 SIMULATED INPUT SKIP 10–440
10.44.1 Overview 10–440
10.44.2 Setup 10–440
10.44.3 Operation 10–443
10.45 SINGULARITY CHECK FUNCTION 10–445
10.46 SOFT FLOAT (OPTION) 10–446
10.46.1 Overview 10–446

10.46.2 Restrictions 10–447
10.46.3 Setting Up Joint Soft Float Schedules 10–449
10.46.4 Setting Up Cartesian Soft Float Schedules 10–451
10.46.5 Setting Up Pushout Soft Float Schedules 10–453
10.46.6 Program Instructions 10–456
10.46.7 General Soft Float Guidelines 10–459
10.46.8 Cartesian Soft Float Guidelines 10–462
10.46.9 Other Guidelines 10–468

10.47 SPACE CHECK FUNCTION (OPTION) 10–470
10.47.1 Setting Up Space Check 10–474
10.47.2 Limitations 10–481

10.48 TCP SPEED OUTPUT (OPTION) 10–482
10.48.1 Overview 10–482
10.48.2 Setting Up TCP Speed Output 10–483
10.48.3 TCP Speed Output Program Instruction 10–485

10.49 TCP SPEED PREDICTION (OPTION) 10–487
10.49.1 Overview 10–487
10.49.2 Setting Up and Running TCP Speed Prediction 10–488
10.49.3 System Variables 10–489

10.50 TEACH PENDANT SHIM 10–501
10.50.1 Overview 10–501
10.50.2 Setting Up and Using TP Shim 10–504
10.50.3 Using TP Shim History to Apply Previous Shims 10–507

10.51 TIME BEFORE/AFTER MOTION OPTION INSTRUCTION 10–510
10.51.1 Overview 10–510
10.51.2 Program Execution 10–510
10.51.3 Execution Timing 10–513
10.51.4 Recording a TIME BEFORE/AFTER Instruction 10–514
10.51.5 TIME BEFORE Instruction Program Example 10–516
10.51.6 Programming Hints 10–517

10.52 TORQUE LIMIT 10–518
10.52.1 Introduction 10–518
10.52.2 Specification 10–519
10.52.3 Restriction 10–520

10.1 ADAPTIVE PROCESS CONTROL

10.1.1 Overview

Adaptive process control (APC) allows you to map a single input to a single output. The purpose is to change the robot's motion dynamically based on an input so that a process can be completed successfully as the specific input parameter changes. Although, grinding is the most typical application for this feature, APC might apply to other process applications such as thermal spray, dispensing, and painting.

To set up APC you must set up the following:

- Input
- Output
- Map functions
- Options

Inputs to the adaptive process control can be:

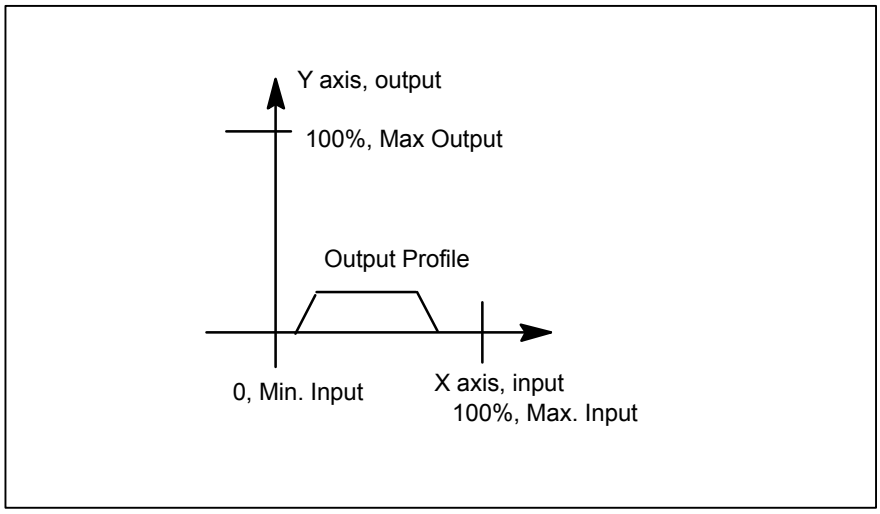
- Analog inputs
- System variables

Allowable outputs can be:

- Speed changes
- Analog output signals
- System variables

A user-definable table maps the input parameter to the output parameter. The table is defined as an x-y table, where the input parameter is the x-axis, and the output parameter is the y-axis. Each axis has 20 points and it is linear interpolated between two adjoining points. The x and y axes are defined by the minimum and maximum input/output range, and each point in the axis represents five percent of the total range. See Figure 10–1 for more details.

Figure 10–1. User-definable Table



APC uses a motion system task to monitor a specific input parameter, and change the output parameter based on a user defined function. The input signal detection is done in every ITP time (8 ms as defined in $SCR.$ITP_TIME).

A typical application for APC is material removal. If there is more material to remove, the grinder has to put out more torque, and the robot has to slow down so excessive material can be removed. The input parameter can be the analog measurement of the torque of the grinder, and the output parameter is the motion speed.

You can also use APC to monitor any other system variables during run-time to improve process quality. For example, APC can be used to monitor the welding current for the arc welding application or to monitor the material flow setting of a DispenseTool application. APC can be used with all of the tools. It also is an independent software option which does not require any other hardware software to support its functionality.

APC is a generic option. DispenseTool, ArcTool, and PaintTool can have similar features that monitor an input parameter when a robot is processing a job. However, these features are tied to a specific hardware device, and cannot be applied to any user defined function. For example, when you use through arc seam tracking software (TAST), you are able to monitor the welding current and change the robot's position accordingly. However, unlike APC, TAST cannot apply to monitor the grinder current for material removable applications, and the internal function that maps the welding current to the change of the robot's position cannot be defined by the user.

Since APC is a generic software option, it might not provide the fine tuned performance such as TAST. However, APC has a user-definable mapping function that is much more versatile than tool specific process control. It is also the only option available.

Based on your selection of the output parameter, APC will change the motion speed and affect the process execution. The performance of APC depends on many factors, such as the input signal delay

time, and the noise to the input signal. Therefore, this function can best be used by an experienced robot programmer who has a good knowledge of the specific application.

10.1.2 Setting Up Input Parameters

You can set up input parameters from the Input Parameters DETAIL screen.

The Input Parameters DETAIL screen displays the items in Table 10–1 .

Table 10–1. Input Parameters

ITEM	DESCRIPTION
AIN	This item defines the analog input port selection.
Port Number	This item defines the analog port number range from 0 - 999.
Maximum Voltage	This item defines the minimum AIN port voltage range from -24 - 24 volts.
Minimum Voltage	This item defines the maximum AIN port voltage range from -24 - 24 volts.
System Variables	This item defines the use of the system variable for the input parameter.
Name	This item defines the name of the system variable.
Max Value	This item defines the maximum value for the system variable as used by the APC table.
Min Value	This item defines the minimum value for the system variable as used by the APC table.

Use Procedure 10-1 in setting up input parameters.

Procedure 10-1 Setting Up Input Parameters

Conditions

- If the analog ports are required, make sure the hardware connection is secured and functioning properly. Refer to Chapter 4 *INPUT/OUTPUT (I/O) SETUP* for details.

Steps

1. Press MENUS.
2. Press SETUP.
3. Press F1, [TYPE].
4. Select Adapt Cntrl. You will see a screen similar to the following.

```
Adaptive Process Control Setup Menu
 1.  Input Parameters:  AIN
 2.  Output Parameter:  Speed
 3.  Map Function:  INCOMPLETE
 4.  Options:  COMPLETE
```

5. Move the cursor to Input Parameter and press F4, [CHOICE].
6. Move the cursor to the input parameter you want and press ENTER.
7. Press F3, DETAIL, to display the DETAIL setup screen of the selected parameters as listed below. If the input parameter is set to analog input port, press DETAIL to get to the analog input port setup menu. Refer to the following screen for AIN information.

```
Analog Port Setup
1.  Port Number:  1
2.  Maximum voltage(v):  10
3:  Minimum voltage(v):  0
```

Refer to the following screen for System Vars information.

```
System Variable Setup
1.  Name: [               ]
2.  Max Value:  0.000
3.  Min Value:  0.000
```

8. Move the cursor to each item and set it as desired.

10.1.3 Setting Up Output Parameters

You can set up output parameters from the Output Setup DETAIL screen.

The Output Parameters DETAIL screen displays the items in Table 10–2 .

Table 10–2. Output Parameters

ITEM	DESCRIPTION
Speed	This item defines the robot motion speed as the adjusted output parameter.
Max Override	This item defines the maximum speed override value range from 1 - 100%
Min Override	This item defines the minimum speed override value range from 1 - 100%.
AOUT	This item defines the analog output port as the output parameter.
Port Number	This item defines the analog output port number.
Maximum Voltage	This item defines the maximum output voltage -24 - 24 volts.
Minimum Voltage	This item defines the minimum output voltage -24 - 24 volts.
System Variables	This item defines the system variable as the output parameter.

Use Procedure 10-2 to set up output parameters.

Procedure 10-2 Setting Up Output Parameters

Conditions

- If the analog ports are required, make sure the hardware connection is secured and functioning properly. Refer to Chapter 4 *INPUT/OUTPUT (I/O) SETUP* for details.

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Adapt Cntrl. You will see a screen similar to the following.

```
Adaptive Process Control Setup Menu
 1.  Input Parameters:  AIN
 2.  Output Parameter:  Speed
 3.  Map Function:  INCOMPLETE
 4.  Options:  COMPLETE
```

5. Move the cursor to Output Parameter and press F4, [CHOICE].
6. Move the cursor to the output parameter you want and press ENTER.
7. Press F3, DETAIL, to display the Speed Output SETUP menu, the analog output setup menu, and the System Variable Output SETUP menu when the cursor is on the output parameter line.

8. Move the cursor to Output Parameter.
9. Press F4, [CHOICE], and select Speed.
10. Keep the cursor on Output Parameter and press F3, DETAIL, to display the Speed Output SETUP menu. You will see a screen similar to the following.

```
Speed Output Setup
  Override Change
    1.  Max Override 100%
    2.  Min Override 1%
```

11. Move the cursor to select the override you want to change and type the new value. Press ENTER.
12. Press PREV to return to the APC main menu.
13. At the APC main menu move the cursor to Output Parameter.
14. Press F4, [CHOICE], and select AOUT.
15. Keep the cursor on Output Parameter and press F3, DETAIL, to display the Analog Output SETUP menu. You will see a screen similar to the following.

```
Analog Output Setup
1.  Port Number:  1
2.  Maximum voltage(v):  10
3:  Minimum voltage(v):  0
```

16. Move the cursor to select the items you want to change and type the new values. Press Enter.
17. Press PREV to return to the APC main menu.
18. At the APC main menu move the cursor to Output Parameter.
19. Press F4, [CHOICE], and select System Vars.
20. Keep the cursor on Output Parameter and press F3, DETAIL, to display the System Variable SETUP menu. You will see a screen similar to the following.

```
System Variable Setup
1.  Name: [               ]
2.  Max Value:  0.000
3.  Min Value:  0.000
```

10.1.4 Setting Up Map Functions

Map functions define the relationship between inputs and outputs. In a grinding process, for example, you might want to slow down the robot when the grinder attached to the robot encounters a larger reaction force (more material to be removed). The relationship between the speed (output) and the grinding force (input) is defined in the map function. You can set up map functions to change the output column as needed. A value of 0-100% represents the full range of minimum value to the maximum value. If the actual input is less than the pre-defined minimum value, APC reads it as if it is the minimum input. If the actual input is greater than the pre-defined maximum value, APC reads it as if it is the maximum value.

You can set up map functions from the Function Parameter DETAIL screen.

The Function Parameter DETAIL screen displays the items in Table 10–3 .

Use Procedure 10-3 to set up map functions.

Table 10–3. Map Functions

ITEM	DESCRIPTION
Input	This item defines the percentage of the input parameter. For example, using an AIN port that has the maximum voltage set to 12 volts and the minimum voltage set to 0 volts. Then, the input % is set to 10% which represents 1.2 volts, and 50% which represents 6.0 volts.
Output	This item defines the percentage of the output parameter. If the motion speed is used for the output parameter, and if the maximum override is 80% and the minimum override is 20% then 20% output represents the robot running at 32% speed.

Procedure 10-3 Setting Up Map Functions

Conditions

- If the analog ports are required, make sure the hardware connection is secured and functioning properly. Refer to Chapter 4 *INPUT/OUTPUT (I/O) SETUP* for details.

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].

4. Select Adapt Cntrl. You will see a screen similar to the following.

```
Adaptive Process Control Setup Menu
 1.  Input Parameters:  AIN
 2.  Output Parameter:  Speed
 3.  Map Function:  INCOMPLETE
 4.  Options:  COMPLETE
```

5. Move the cursor to Map Function and press F3, DETAIL. You will see a screen similar to the following.

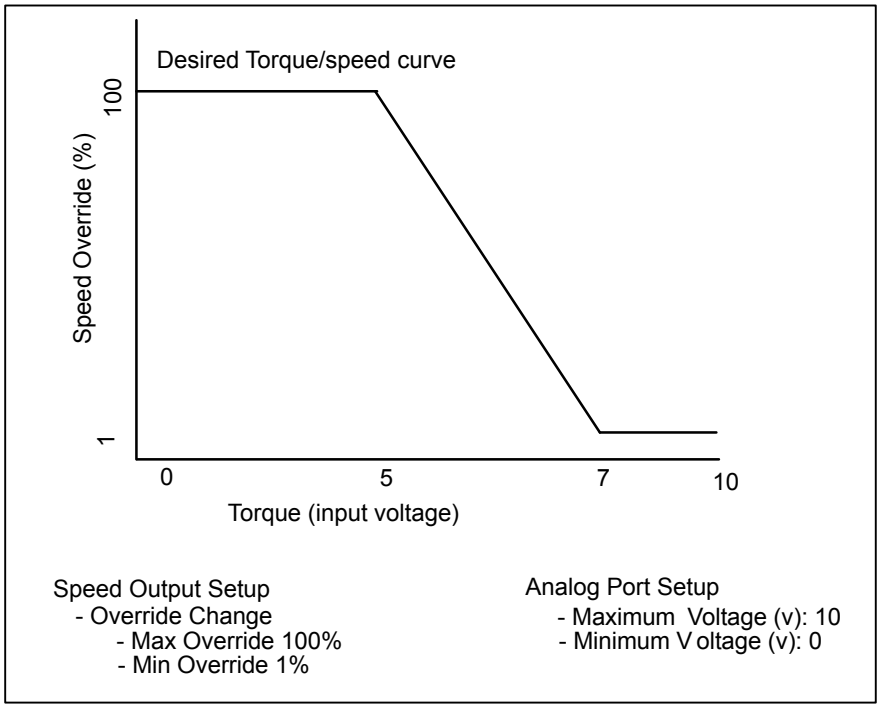
```
Function Setup Menu
 Input:                              Output:
1.  0%                               100%
2.  5%                               100%
3.  10%                              100%
4.  15%                              100%
...
10.  50%                             100%
11.  55%                             75%
12.  60%                             50%
13.  65%                             25%
14.  70%                             0%
15.  75%                             0%
...
20.  100%                            0%
```

6. Move the cursor to each output percentage you want to change. Type the number and press ENTER.

7. When you are finished typing the output percentage, press PREV to return to the Adaptive Control Menu. The value of the function will be COMPLETE.

Note See Figure 10–2 for information on the change speed override based on torque.

Figure 10–2. Change Speed Override Based on Torque (Analog Input)



10.1.5 Setting Up Options

The options menu allows you to set the recording position. Use F4 to enable an option, or use F5 to disable an option. The position recording, if enabled, allows you to record a position when the input parameter exceeds some pre-defined threshold. The position is recorded to the system variable $APC.$recd_pos.

Table 10–4 lists the option you can set up.

Use Procedure 10-4 to set up options.

Table 10–4. Options

ITEM	DESCRIPTION
Position Record (Enable or Disable)	This item when enabled, allows APC to record a position when the input parameter exceeds the predefined threshold.
Threshold %	This item defines the recording threshold.

Procedure 10-4 Setting Up Options

Conditions

- If the analog ports are required, make sure the hardware connection is secured and functioning properly. Refer to Chapter 4 *INPUT/OUTPUT (I/O) SETUP* for details.

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Adapt Cntrl. You will see a screen similar to the following

```
Adaptive Process Control Setup Menu
 1.  Input Parameters:  AIN
 2.  Output Parameter:  Speed
 3.  Map Function:  INCOMPLETE
 4.  Options:  COMPLETE
```

5. Move the cursor to Options and press F3, DETAIL. You will see a screen similar to the following.

```
Option Setup Menu
Position Recording:  DISABLE
 Threshold:  50%
```

6. Move the cursor to each item and set it as desired.
7. When you are finished, press PREV to return to the Adaptive Control Menu. The value will be INCOMPLETE or COMPLETE.

10.1.6 Adding the APC Instructions To a Program

Use Procedure 10-5 to add an APC instruction in your program.

Procedure 10-5 Adding APC Instructions

Steps

1. Press MENUS.
2. Select SETUP.
3. Select Adaptive Process Control.
4. Select Setup Input Parameter.
5. Select Setup Output Parameter.
6. Define the function you want to perform.
7. In the program, add the CALL APCT_ON instruction when you need to enable the APC.

   Note After the $AP_CTRL.$enable is turned on (when APCT_ON is called) the software will monitor the teach pendant program status. When the program is aborted, APC will set the variable to OFF. If you want to terminate the APC in the middle of the program, insert the CALL APCT_OFF instruction in your teach pendant program.

8. Add the CALL APCT_OFF instruction to the teach pendant program when you want to turn off the APC before the end of the program.

```
Teach Pendant Program
JP [1] 100% CNT100
LP [2] 50mm/sec FINE move to process point
CALL APCT_ON
LP[3] 50mm/sec FINE
LP[4] 50mm/sec FINE
LP[5] 50mm/sec FINE
CALL APCT_OFF
JP[1] 100% CNT100
```

10.2 ANGLE ENTRY SHIFT FUNCTION (OPTION NOT AVAILABLE IN NORTH AMERICA)

If you are going to make multiple holes arranged at equal distances on the circumference of a circle (such as mounting bolt holes in a vehicle wheel), you can reduce the total number hours for a teach process by teaching data for one hole and having data for the other holes generated by a program shift function.

The conventional representative point teach method or the direct shift amount input method (using values in the rectangular coordinate system) is not suitable for this shift function.

This function determines the required shift amount according to three representative points taught to determine a rotation axis and directly entered rotation angles so that it can perform a programmed shift.

The angle entry shift function performs a programmed shift according to three or four representative taught points and directly entered rotation angles. Specifying a number of repetitions causes the function to make multiple shifts at equal distances on the circumference of the same circle.

Function

For the angle entry shift function, the following items must be specified.

Transformation-source program name /transformation range

All or part of the existing programs are specified as transformation sources. The specification method and the meaning are the same as for the program shift function.

Transformation-destination program name/insertion line

The angle entry shift function processes the transformation results by either of the two:

- Creating new programs
- Inserting lines into existing programs

This item is used to specify the name of a new or existing program. The specification method and the meaning are the same as for the program shift function.

Representative Points P1, P2, P3, and P0

There are two methods; in one method, a rotation axis is specified and in the other method, no rotation axis is specified.

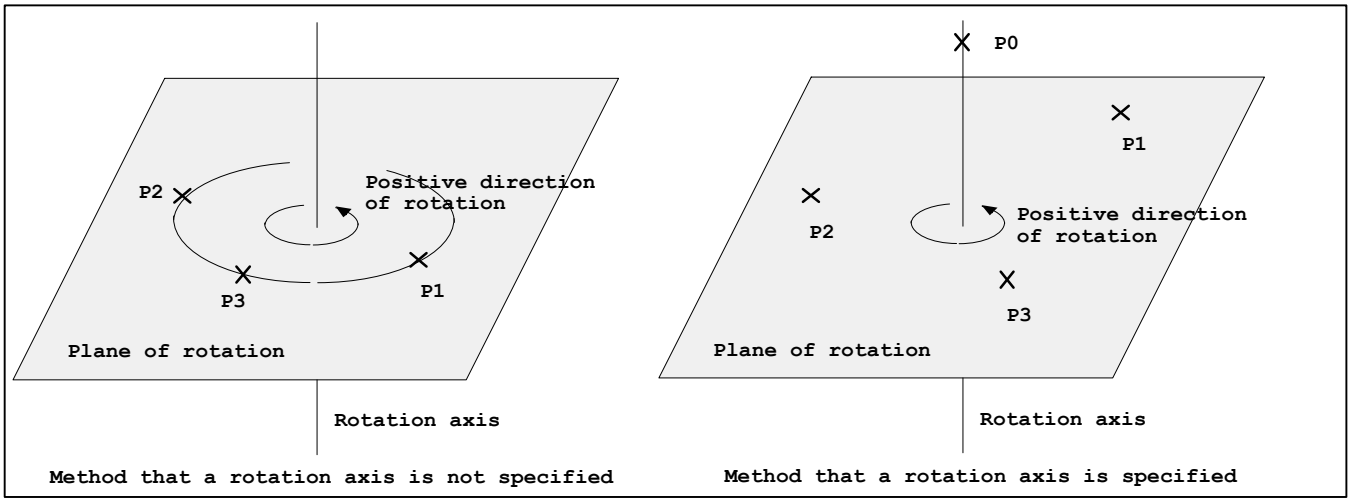
- **Method that a rotation axis is not specified (default setting)** Three points P1, P2, and P3 are specified on the circumference of a circle. P0 is not used. The direction of rotation from P1 toward P2 is defined as positive. The plane and center of rotation are determined from these three points.
- **Method that a rotation axis is specified** P0 is specified on the rotation axis, and P1, P2, and P3 are on the plane of rotation. The direction of rotation is defined as positive for rotation from P1 toward P2. The plane of rotation are determined from P1, P2, and P3, and an axis that is perpendicular to the plane of rotation and passes through P0 is specified as the rotation axis.

If transformation is performed without specifying a rotation axis, the calculated center of rotation is automatically specified as P0. The value for P0 can be directly changed later. So, the center of rotation can be corrected by enabling the rotation axis in the second and subsequent transformation phases.

Rotation Angle

The rotation angle is specified in degrees for the rotation center and plane determined by the three representative points. The rotation angle is directly entered as a signed real number. The positive number corresponds to the direction of rotation from P1 toward P2. See Figure 10–3 .

Figure 10–3. Two Methods of Angle Entry Shift Function



Repetition Count

The repetition count indicates how many times transformation is to be performed. If places to be operated on are arranged at equal distances on the circumference of the same circle, specifying a repetition count enables simultaneous transformation for multiple places. If the repetition count is 2 or greater, a comment line is added at the beginning of a portion created by shifting.

Consider the following example:

Transformation-source program: program A

```
1: J P[1] 100% FINE
2: L P[2] 1500mm/sec FINE
```

If this program is transformed with the rotation angle as 20deg, the repetition count as 3, and the transformation-destination program as program B, the following program (program B) is generated with a comment added automatically.

Transformation-destination program: program B

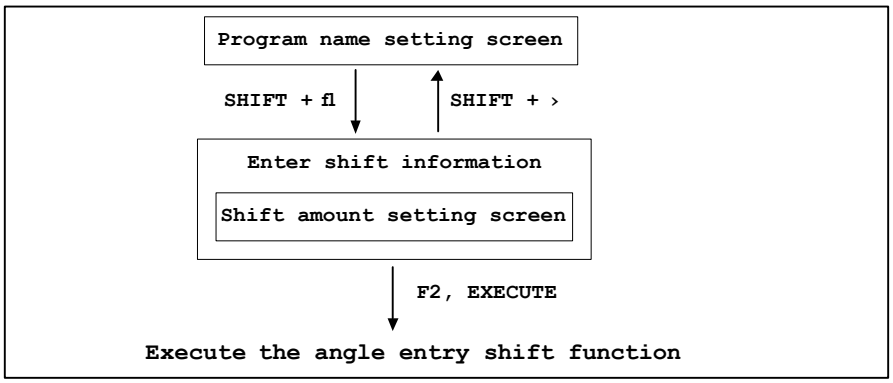
```
1: !Angle entry shift 1 (deg 20.00)
2: J P[1] 100% FINE
3: L P[2] 1500mm/sec FINE
4: !Angle entry shift 2 (deg 40.00)
5: J P[3] 100% FINE
6: L P[4] 1500mm/sec FINE
7: !Angle entry shift 3 (deg 60.00)
8: J P[5] 100% FINE
9: L P[6] 1500mm/sec FINE
```

The position data for program A has been transformed as follows:
P[1]: Position rotated by 20deg from P[1] for program
P[2]: Position rotated by 20deg from P[2] for program A
P[3]: Position rotated by 40deg from P[1] for program A
P[4]: Position rotated by 40deg from P[2] for program A
P[5]: Position rotated by 60deg from P[1] for program A
P[6]: Position rotated by 60deg from P[2] for program A

Function and Operating Procedure

The screen to set up the angle entry shift function consists of two pages; program name setting screen and shift amount setting screen. See Figure 10–4 .

Figure 10–4. Angle Entry Shift Screen Structure



The program name setting screen works similarly to that for the program shift function. The shift amount setting screen is described below.

The angle entry shift function is set up on the UTILITIES Angle entry shift screen. First the program name setting screen appears. On this screen, specify the transformation-source program name, transformation range, transformation-destination program name, and insertion lines. Refer to the description of the program shift utility in Section 10.36 .

After you specify all the necessary information, press the SHIFT + ↓(down arrow) key to move to the shift amount setting screen.

The shift amount setting screen is as shown below:

```
ANGLE ENTRY SHIFT
 Shift amount
  Position data
X:*****.**   Y:*****.**   Z:*****.**
1 Rotation plane          P1: [ ]
2                         P2:
3                         P3:
4 Rotation axis enable:        FALSE
5 Rotation axis        PO: not used
6 Angle(deg):                  0.00
7 Repeating times                 1
```

Each item is specified as described below. The specification method slightly varies depending on whether to specify a rotation axis.

- Method when a rotation axis is not specified:

If "Rotation axis enable:" is set to "FALSE," specify P1, P2, and P3, which are three points on the circumference of a circle whose center coincides with the rotation axis to be shifted. The farther these points are separated, the higher the precision of transformation. A method to teach the representative points is the same as for the program shift function. Move the cursor to the point to be taught and press the SHIFT + F5, RECORD.

To specify a rotation angle, set the cursor on the "Angle (deg)" line and enter a signed real number representing the angle to be rotated through (a plus sign is omissible). The direction from P1 (the first point to be specified) toward P2 is assumed to be positive.

To perform transformation two or more times, set the "Repeating times" item to the desired count.

After the settings above are completed, press F2, EXECUTE. If you answer the question "Execute transform?" by pressing F4, YES, the angle entry shift function is executed.

At this point, P0 is automatically defined to be the center of the circle whose circumference coincides with P1, P2, and P3.If a satisfactory precision is not obtained at the first transformation, change the center of the circle at the second transformation by directly specifying the rotation axis (setting "Rotation axis enable:" to "TRUE"). This method can fine-adjust transformation.

- Method when a rotation axis is specified

Setting "Rotation axis enable:" to "TRUE" causes the transformation to be performed with the rotation axis specified. In this case, you specify three points in the rotation plane as P1, P2, and P3 (they need not be on the circumference of the same circle).

The teach method is the same as for when a rotation axis is not specified. The farther these three points are separated, the higher the precision.

When this method is used, "not used" beside "P0:" disappears, indicating that P0 can be taught.

Any point on the rotational axis can be taught as P0. The accuracy of the intersection point between rotational plane and axis increases as this point gets closer to the plane defined by P1, P2 and P3.

P0 can be specified by recording position, referring position, or direct entry. The methods of recording position and referring position are the same as those for the other representative points. The direct entry method only applies to P0. Pressing the ENTER key with the cursor at the "P0" line causes the rotation axis direct entry screen, shown below, to appear.

```
Angle Entry Shift
Shift amount
Rotation center axis direct entry
 1 Frame:                  USER FRAME 1
 2 X:                               0.00
 3 Y:                               0.00
 4 Z:                               0.00
```

On this screen, the location of P0 can be specified by entering coordinate value in an arbitrary coordinate system.

To change a coordinate system, set the cursor to the "Frame:" line and press F4, [CHOICE]. A screen appears. On this screen, select the desired coordinate system. The selected coordinate system is only used to display/specify X, Y, and Z coordinates on this screen. If the coordinate system is changed, "X:", "Y:", and "Z:" values are transformed to values in the new coordinate system and re-displayed.

It is also possible to directly enter values by setting the cursor to "X:", "Y:", or "Z:".

If all necessary settings are completed, it is possible to execute the shift function on this screen.

When F2, EXECUTE, is pressed, the message "Execute transform?" appears. Responding to this question with F4, YES, executes the angle entry shift function.

When the shift function is executed with a rotation axis specified, the specified value is used for P0. Unlike the method with no rotation specified, no automatic calculation is performed to specify a value for P0.

To return to the shift amount setting screen (previous screen), press the PREV key.

Turn Number

In some robot models, some of the wrist axes can rotate more than 360 deg. For these robots, even if the tool center point remains at the same position, the axis angle might be in a phase different by 360

deg. In this case, if you select an incorrect angle, the axis might rotate largely when a program after transformation is executed, resulting in a dangerous behavior of the robot.

The shift function performs optimization to minimize such danger. The optimization is not necessarily perfect, however, and the following message is output together with the choices of axis angles in some cases.

```
Repeat 3: Select P[1]:J6.(deg 183)
  183Â°-177Â°*uninit*
QUIT>
```

- Normally select the value assigned to the F1 key "183°" in the above example), because this is determined through optimization by the shift function.
- The value assigned to the F2 key is a value before optimization. This value is not normally used. It can be used if the value at the F1 key fails.
- Press F3, *uninit*, to cancel the taught data. If the transformation-destination program is executed under this condition, an error occurs. The data should be taught again.
- Press F5, QUIT, to abort transformation.

If a repetition count is specified for the angle entry shift function, optimization selects an angle close to the result of the previous transformation.

If the angle entry shift function performs one rotation's worth of transformation, an axis is rotated by 360°. This will generate data corresponding to a position beyond a stroke limit in many cases. To solve this problem, take either of the following actions:

- Design the transformation-source program so that the stroke limit will be not exceeded even after one rotation. This method is preferred.
- Invert the sign of the rotation angle and regenerate the position data so that it will not reach the stroke limit. In this case, two programs are generated and they have a different rotation direction. These programs must be combined later by editing. Use this method as required.

10.3 AUTO SINGULARITY AVOIDANCE

10.3.1 Overview

Robot singularity causes regional structure degeneracy or wrist degeneracy depending on the structure of a robot. For most six-axis robots, the singularity that occurs at the elbow (at the boundary between

"up" and "down" configurations) or at the boundary between "front" and "back" configurations causes regional structure degeneracy. This kind of singularity easily can be avoided by restricting the workspace of a robot. However, the wrist singularity might happen virtually any place inside the workspace. When a robot travels through or near a wrist singular position, motion performance becomes undesirable because:

- Wrist axis 4 and axis 6 will rotate a large angle often within a short time period.
- The TCP speed of the robot will slow down as a result of limiting the wrist axes joint speed.
- The path might deviate from the programmed path if motor speed limit is exceeded.

Without the Auto Singularity Avoidance option, the system uses the system variables $param_group[].$auto_sngstp and $param_group[].$t1t2_sngstp to control the robot behavior when singularity is encountered in the following way:

- Auto Mode
  - If $param_group[].$auto_sngstp = FALSE (default), the robot is allowed to move through singularity but with undesirable motion performance.
  - If $param_group[].$auto_sngstp = TRUE, the robot will stop when singularity occurs.
- T1/T2 Mode
  - If $param_group[].$t1t2_sngstp = TRUE (default for HandlingTool), the robot will stop when singularity occurs.
  - If $param_group[].$t1t2_sngstp = FALSE (default for P-Series robots), the robot is allowed to move through singularity but with undesirable motion performance.

Auto Singularity Avoidance provides a real-time solution to wrist singularity. **With** the Auto Singularity Avoidance option, the system will:

- Move the robot through wrist singularity smoothly without excessive wrist rotation. As a result, the TCP speed can be maintained.
- Support both teach pendant program motions and jogging motions with the singularity avoidance features.

When this function is active, the system variables $param_group[].$auto_sngstp and $param_group[].$t1t2_sngstp will be ignored and the system will allow the robot to move through singularity smoothly.

When this function is not active, the system behaves the same way as before as if Auto Singularity Avoidance option is not loaded.

In order to use auto singularity avoidance, you must have a 6-axis robot with a wrist, or a P-500 robot, and a FANUC robot controller.

10.3.2 How it Works

For Linear motion with the auto singularity avoidance function enabled, the system will:

- Predict singularity based on one segment when given the start and destination positions.
- Treat motion in the conventional way if singularity is not predicted.

If singularity is predicted at any point along the segment, the following motion control method will be used:

1. Determine the best wrist configuration (flip or nonflip) of the destination position.

   Note The taught wrist configuration might not be used. Hence the actual wrist configuration might be different from the taught one, as shown in Figure 10–5 .

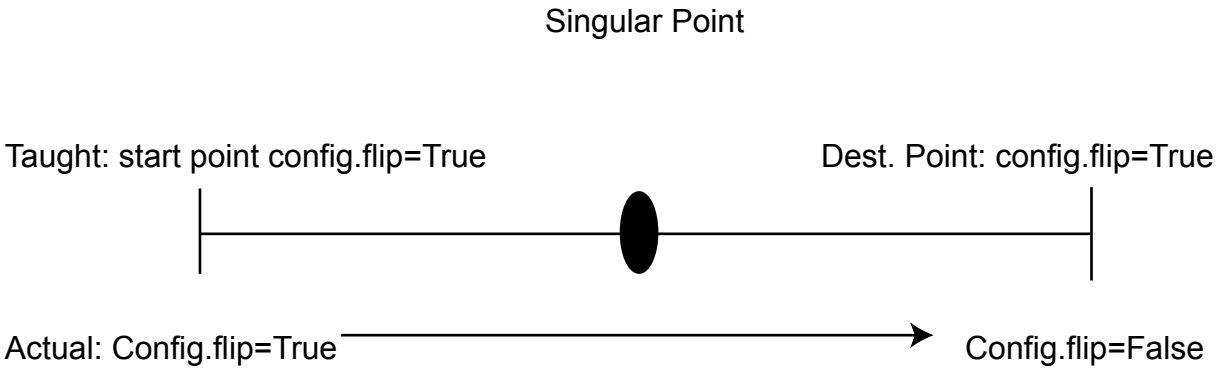
2. Directly control movement of axis 4 from the start point to the destination point.
3. Maintain TCP positions during the motion.
4. Minimize orientation errors during the motion.

By default, singularity avoidance feature for linear jogging is enabled with a default singularity zone. With singularity avoidance feature enabled, Linear jogging of the robot will result in the following behavior:

- Away from the singularity zone, conventional jogging is used.
- Inside the singularity zone, the singularity avoidance control method takes effect. Therefore, the robot can move through singularity smoothly without high wrist axes speed.

Figure 10–5 shows an example of how the auto singularity avoidance works with linear motion.

Figure 10–5. Taught Configuration vs. Actual Configuration



Note that when this function is enabled, the system will ignore flip/nonflip configuration of the taught positions you specified and will determine automatically the new configuration for you.

10.3.3 Compatibility

Auto singularity avoidance works with:

- Baseline system
- Extended Axis option
- Multi group motion option
- Accupath option

10.3.4 Limitations

The following limitations apply to auto singularity avoidance.

1. The function only supports LINEAR motion type. That is, Circular motion type is not supported.
2. Auto singularity avoidance does **not** support:
   - Line Tracking, Circular Tracking, or Rail Tracking

     Note For P-500 robots with PaintTool, Auto Singularity Avoidance supports Line Tracking if the conveyor rail is parallel or close to parallel to the rail (Axis 1) of the P-500.

   - Coordinated Motion option
   - Remote TCP option
   - Continuous Turn

3. In the case where Accupath option is used, when singularity is encountered, constant path may not be maintained around the singularity.

10.3.5 Cautions

Keep the following important information in mind when you set up and use auto singularity avoidance.

1. With the auto singularity avoidance function, actual wrist configuration (flip/nonflip) might be different from the taught destination positions.

   The function might change configuration internally not only for the destination position in the motion line where singularity is detected but also for the subsequent destination positions in the following motions inside the special teach pendant program.

   For example, suppose that the taught points all have "F" (flip) configuration.

```
20 L P[16] 1000 mm/s CNT100 --- singularity is predicted
21 L P[17] 500mm/s   CNT50 --- no singularity
22 L P[18] 200mm/s FINE --- no singularity
```

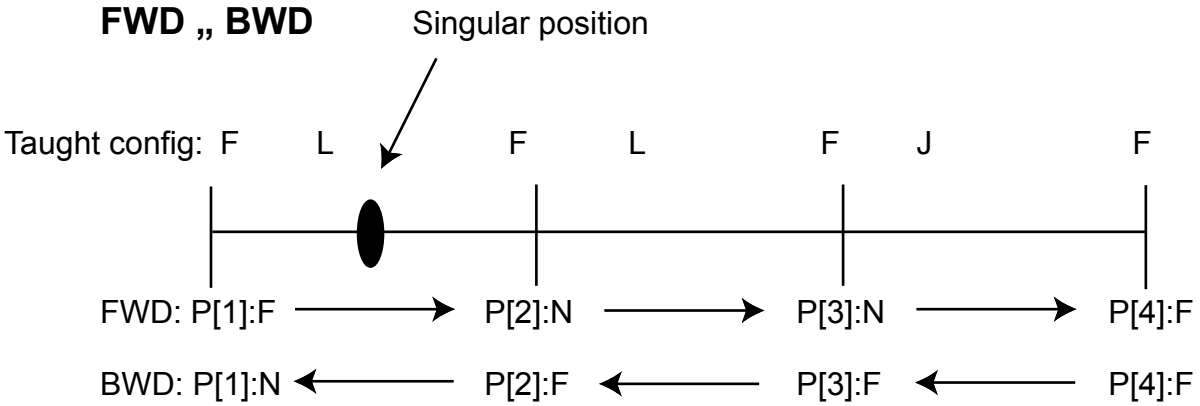
   If singularity is detected for line 20 during the execution of the special teach pendant program, the function will internally use “N” (non-flip) the configuration of the destination position in the following way:

```
20 L P[16] 1000 mm/s CNT100 --- change config to "N"
21 L P[17] 500mm/s   CNT50 --- change config to "N"
22 L P[18] 200mm/s FINE --- change config to "N"
```

   The system will use internally a different configuration type at points P[16], P[17], and P[18]. Note that the Cartesian position is reached but with different wrist joint angles.

2. Since the function will change configuration internally, single step forward and backward might produce different behavior. See Figure 10–6 for an example.

Figure 10–6. Single Step Forward (FWD) vs. Backward (BWD)



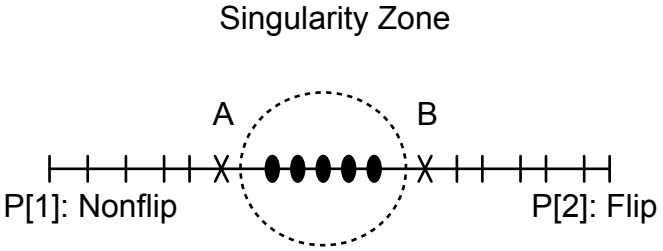
The following rules will be applied:

- JOINT motion will reach the taught configuration and LINEAR motion should keep the same configuration as the one at the previously reached position unless singularity occurs.
- In Figure 10–6 , suppose that the taught configurations at all the points are "flip" (F). When moving single step forward from P[1] to P[4], since the first segment is LINEAR motion from P[1] to P[2] and there is singularity, the configuration at P[2] is changed to "non-flip" (N) to avoid the singularity. The second linear motion from P[2] to P[3] does not have any singularity and should keep the same configuration as the one at P[2]. Therefore, the configuration at P[3] is "non-flip". The last motion is JOINT and will reach the taught configuration--"flip".
- When moving single step backward from P[4] to P[1], the first motion is JOINT from P[4] to P[3] and will reach the taught configuration --"flip" (F) at P[3]. The second motion is LINEAR from P[3] to P[2] and has no singularity. Hence the configuration at P[2] is kept the same as the one at P[3]. The last motion is LINEAR from P[2] to P[1] and there is singularity. The actual configuration at P[1] is changed to "non-flip" (N) to avoid the singularity. As shown Figure 10–6 , the single step forward and backward produce different behavior in the example.

  To prevent the above from happening, during single step forward motion, the system will post a warning message "MOTN-208 Config Not Reached (L:%d)" at the point where the actual configuration is different from the taught one. By observing the message, the user should re-touch up the taught point at the specific line shown in the warning message this will update the taught configuration to the actual configuration. As a result, when stepping backward, the motion will behave the same way as stepping forward.

3. Jogging and program motion might be different when the robot moves near singularity.

Figure 10–7. Program Motion vs. Jogging: Behavior Might Be Different Near Singularity



The behavior of LINEAR jogging through a singularity zone:

As shown in Figure 10–7 , the jogging motion from P[1] to P[2] can be partitioned into three regions: (1) from P[1] to A; (2) from A to B and (3) from B to P[2]. The regions (1) and (3) are outside of a pre-defined singularity zone, whereas region (2) is inside the zone. With Auto Singularity Avoidance Jogging function enabled, when you jog linearly from P[1] to A, the jogging has the conventional jogging behavior and axis 4 may rotate gradually in one direction. From A to B, linear jogging will use new orientation control method, moving smoothly through singularity, and the configuration at B is different from the one at A. From B to P[2], the jogging yields the conventional jogging behavior again and axis 4 gradually rotates in the same direction as you jog from P[1] to A. In summary, when linearly jogging a robot from P[1] to P[2] in the above example, you might notice that:

- Axis 4 gradually rotates back and forth
- The configuration changes from P[1] to P[2]

The behavior of LINEAR program motion through a singularity zone:

As shown in Figure 10–7 , during the program motion from P[1] to P[2], the system will predict the singularity and use new orientation control method for the entire motion. As a result, you might notice that:

- The robot moves smoothly through the singularity
- Axis 4 gradually rotates in one direction only
- The configuration changes from P[1] to P[2]

4. The destination position is inside the singularity zone. By default, the system will change the taught position in the following way:
   - Keep location intact
   - Use the same joint angle of Axis 4 as the previous point
   - Minimize the orientation error between the modified orientation and the taught orientation

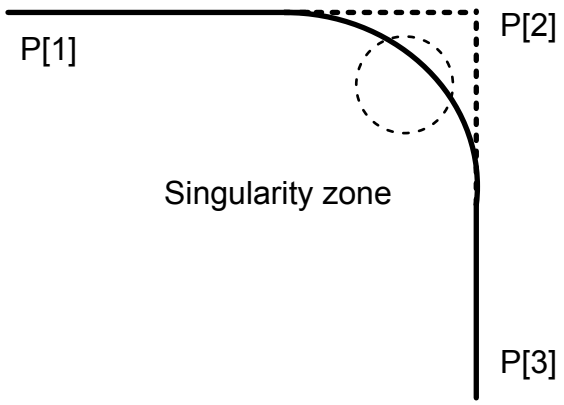
   This results in a more intuitive motion behavior. But if the exact taught position required, you can overwrite this default system behavior by setting $ra_paramgrp[].$use_strt_j4 = FALSE.

When the system internally modifies the taught orientation to control the rotation of axis 4, it will post a warning message "MOTN-209 Modify Singular Dest(L:%d)" to indicate the taught point at the specific line is singular and is modified internally (minimal orientation error is expected and the location of the modified destination point is the same as the taught one).

5. The function might not help for the singularity at the corner path.

   For example, you might teach three points as shown in the following figure. Singularity might not be detected on either of the two segments. However, singularity might occur on the corner path (blending of the two segments). Since in planning it is impossible to know the blending with one segment information, the function does not help in this situation.

Figure 10–8. Unforeseeable Singularity at a Corner Path



10.3.6 How to Use Auto Singularity Avoidance

To use auto singularity avoidance with program motion, refer to Procedure 10-6 .

To use auto singularity avoidance with jogging, refer to Procedure 10-7 . This procedure describes how singularity avoidance can be enabled or disabled for jogging. By default, singularity avoidance is enabled during jogging.

Procedure 10-6 Using Auto Singularity Avoidance with Program Motion

Warning

When you execute a program with the auto singularity avoidance function, you are changing the configuration of taught points. This affects the motion behavior of the robot. When you execute the program, make sure the program is running at safety speed to verify that it is working correctly; otherwise, you could injure personnel or damage equipment.

Note The Auto Singularity Avoidance feature under program motion can be enabled or disabled in the teach pendant program header field called "Enable Singularity Avoidance".

Conditions

- The teach pendant program contains the motion group that has a 6-axis robot with a wrist or a P-500 robot.
- The system uses a controller with Auto Singularity Avoidance software loaded.
- Auto Singularity Avoidance is enabled for the appropriate motion group ($ra_paramgrp[].$enable = TRUE by default).

Steps

1. Press Select.
2. Move the cursor to select the program you want to modify.
3. Press F2, DETAIL. You will see a screen similar to the following.

```
Program detail               JOINT  10 %
                                    1/6
 Creation Date:           29-Jul-xxxx
 Modification Date:       29-Jul-xxxx
 Copy Source:      [                ]
 Positions: FALSE  Size:     135 Byte

  1  Program name:          [TESTMAIN]
  2  Sub Type:      [None             ]
  3  Comment:       [                 ]
  4  Group Mask:           [1,*,*,*,*]
  5  Write protect:         [OFF     ]
```

4. Press F3, NEXT. You will see a screen similar to the following.

```
TESTMAIN          LINE 0     T2
                             JOINT  10 %
    Singularity Avoidance Func     1/1
    Program Header


  1 Enable Singularity Avoidance: TRUE
```

5. Press F4, TRUE, to set Enable Singularity Avoidance to TRUE.

The system will try to maintain the consistency between the singularity avoidance header flag of program motion and the singularity avoidance jog mode.

By default, the system uses the following rules to set the "Enable Singularity Avoidance" program header field.

- When a **new** program is created, the status is set based on the jog mode.
  - — If singularity avoidance jog mode is enabled (this is the default), "Enable Singularity Avoidance" is set to TRUE. Refer to Procedure 10-7 .
  - — If singularity avoidance jog mode is disabled, "Enable Singularity Avoidance" is set to FALSE. Refer to Procedure 10-7 .

- Since you can manually change the "Enable Singularity Avoidance" program header field, you can overwrite the default system behavior by manually enabling or disabling singularity avoidance function for a teach pendant program.
- By default, for **old** programs, a new program header field "Enable Singularity Avoidance" is created and set to FALSE by the system. Hence, by default, **old** programs will behave the same as the old way.

Note If you are using singularity avoidance jogging to touch up a point in the teach pendant program that is a disabled singularity avoidance function, then the program might NOT run through because the taught configuration might be different.

6. Run the teach pendant program. Refer to Chapter 7 *TESTING A PROGRAM AND RUNNING PRODUCTION* for more information about testing a program.

Under T1/T2 mode, during the execution of a teach pendant program with Auto Singularity Avoidance function enabled, the system may display the following warning messages on the top of teach pendant:

- MOTN-205 (program name, xx) Singularity
- MOTN-208 Config Not Reached (L: xx)
- MOTN-209 Modify Singular Dest(L: xx)

where xx indicates the line number in the program. Under Auto mode, no such message will be displayed.

MOTN-205 (program name, xx) Singularity: This warning message shows that the system predicts a singularity for the motion line xx in the named program. Most likely, the system will use different configuration for the motion line and the subsequent motion line(s) to avoid the singularity.

MOTN-208 Config Not Reached (L:xx): This warning message is displayed only when the program is executed with single step forward. It shows that the actual configuration is different from the taught one at the specific motion line. It is recommended that the user re-touch up the point at the particular line to make the single step forward and back behave the same way.

MOTN-209 Modify Singularity Dest(L:xx): This warning message is displayed only when $ra_paramgrp[].$use_strt_j4 = TRUE and the taught point is in a pre-defined singularity zone (the absolute angle value of axis 5 is less than $ra_paramgrp[].$sngrlty_tol). It indicates that the destination point at the specific line is singular and its orientation is modified to control the motion of axis 4. To reach exactly the taught point even if it is singular, set $ra_paramgrp[].$use_strt_j4 = FALSE.



Warning

The behavior of teach pendant program motion with singularity avoidance function enabled might NOT reach the taught configuration. When you execute the program, make sure the program is running at a safe speed to verify that it is working correctly; otherwise, you could injure personnel or damage equipment.



Warning

The behavior of teach pendant program motion with singularity avoidance function enabled or disabled might be very different. The flip/non-flip configuration will be different if there is singularity. When you execute the program, make sure the program is running at a safe speed to verify that it is working correctly. Pay close attention to wrist axes motion; otherwise, you could injure personnel or damage equipment.

Warning

With singularity avoidance function enabled, and a robot is traveling through or near a singular position, an orientation error is expected. When you execute the program, make sure the program is running at safety speed to verify that it is working correctly; otherwise, you could injure personnel or damage equipment.

Warning

With singularity avoidance function enabled, single step forward and backward might result in a different configuration. When you execute the program, make sure the program is running at safety speed to verify that it is working correctly; otherwise, you could injure personnel or damage equipment.

Procedure 10-7 Using Auto Singularity Avoidance with Jogging

Warning

When you execute a program with the auto singularity avoidance function, you are changing the configuration of taught points. This affects the motion behavior of the robot. When you execute the program, make sure the program is running at safety speed to verify that it is working correctly; otherwise, you could injure personnel or damage equipment.

Conditions

- The motion group you want to jog has a 6-axis robot with a wrist or a P-500 robot.
- The system uses a controller with Auto Singularity Avoidance software loaded.
- Auto Singularity Avoidance is enabled for the appropriate motion group ($ra_paramgrp[].$enable=TRUE by default).

Steps

1. Select the motion group you want to jog if your system is configured as multiple motion groups:

   Note The group should contain a 6-axis robot with a wrist or a P-500 robot.

   a. Press and hold the SHIFT key and press COORD to display the JOG menu.

   b. Move the cursor to GROUP and press the appropriate numeric key.

2. Select a coordinate system among the frames WORLD, TOOL, USER, JGFRM, by pressing COORD.

By default, the system enables singularity avoidance jogging. From the status line indicator, if singularity avoidance jogging function is enabled, the jog mode will be displayed in the upper right hand corner of the teach pendant screen similar to the following:

```
Program name          G1          S/WORLD   10%
```

If you see "S/" followed by a coordinate system on the teach pendant, skip Step 3 for singularity avoidance jogging.

The system will try to maintain the consistency between the singularity avoidance header flag of program motion and the singularity avoidance jog mode.

By default, the system uses the following rules to enable/disable singularity avoidance for jogging.

- By default, when the system comes up without any program selected, singularity avoidance jog mode is enabled.
- When a teach pendant program is being selected or edited, the jog mode status is set based on the "Enable Singularity Avoidance" header of the program.
  - — If the header field "Enable Singularity Avoidance" of the program is TRUE, singularity avoidance JOG mode is enabled.
  - — If the header field " Enable Singularity Avoidance" of the program is FALSE, singularity avoidance JOG mode is disabled.

This is an attempt to make the robot jogs in the same way as it would move with program motion.

However, you can overwrite this default system behavior by manually enabling/disabling singularity avoidance JOG mode as follows.

3. Enable singularity avoidance jogging function.

a. Press FCTN

b. Move the cursor to "T/SINGULARITY JOG" and press ENTER. The status line indicator for singularity avoidance jog is displayed in the upper right hand corner of the teach pendant screen similar to the following.

```
Program name          G1            S/WORLD   10%
```

c. To deselect singularity avoidance jogging function:

a. Press FCTN.

b. Move the cursor to "T/SINGULARITY JOG" and press ENTER. The status line indicator should be similar to the following:

```
Program name            G1           WORLD   10%
```

Note Singularity avoidance jogging function is NOT available or applicable in the following jog modes:

- JOINT (joint jogging)
- WRIST JOINT JOG (wrist joint jogging)
- RTCP JOG (remote TCP jogging)
- COORD JOG (coordinate jogging)

4. Turn the teach pendant ON.

5. Choose jog speed by pressing and releasing the appropriate jog speed key.



Warning

In the next step, the robot will move. To stop the robot immediately at any time during jogging, release the DEADMAN switch or press the EMERGENCY STOP button.

6. To jog, press and hold the SHIFT key and continuously press the jog key that corresponds to the direction in which you want to move the robot. To stop jogging, release the jog key.

7. When you are finished jogging, turn the teach pendant OFF and release the DEADMAN switch.



Warning

When using singularity avoidance jog mode, if the robot travels through or near a singular position, an orientation error is expected. Make sure that the motion is all right. Otherwise, you could injure personnel or damage equipment.

10.3.7 System Variable Setup

The system variable $ra_paramgrp[] is set on each motion group for the singularity avoidance function and contains a variety of variables.

Refer to the *Software Reference Manual* for a complete list of all system variables and their settings.

10.3.8 Troubleshooting

Refer to Table 10–5 for troubleshooting information.

Table 10–5. Troubleshooting Solutions

Problem	Possible Cause	Solution
Limit error during program motion	Tool set up is not correct.	Set the tool frame correctly.
	The header of the teach pendant program is different from its original one.	Make sure the header of the program is really what you want.
	The start joint angle is already too close to the limit.	Change the flip/non-flip configuration at the start point.
	The joint angle of the axis with limit error is accumulated in one direction during the program motions.	Look at current joint angles and adjust taught points so that limit error won't occur. Insert some JOINT motion to "re-wind" the axis.

Table 10–5. Troubleshooting Solutions (Cont'd)

Problem	Possible Cause	Solution
Position Config change	The program is called by another program and has disabled singularity avoidance and the first motion in the program is a Cartesian motion (Linear/Circular).	Use JOINT motion right before the motion line where the error occurs.
	The program has disabled singularity avoidance and calls another singularity avoidance teach pendant program.	If appropriate, enable the header to apply singularity avoidance function.
	The program has disabled singularity avoidance and the point(s) in the program is taught with singularity avoidance jogging enabled.	Adjust the configuration of the taught point(s) or enable Auto Singularity Avoidance for the teach pendant program.
	$ra_paramgrp[]. $enable=FALSE	Set $ra_paramgrp[1]. $enable=TRUE and then turn the controller off and back on for the change to take effect.
Single step backward is different from forward	Singularity occurs at a LINEAR motion that is prior to a JOINT motion.	The difference is expected with singularity avoidance function enabled. Use single step forward mode to execute the program motion and re-touch up the taught point where the warning message "Config not reached" is posted.

Table 10–5. Troubleshooting Solutions (Cont'd)

Problem	Possible Cause	Solution
Large rotation during JOINT motion.	JOINT motion will reach the taught configuration and the taught turn numbers.	Use MROT option at the JOINT motion line to get minimal rotation of joint angles during the JOINT motion. However, by doing so, the behavior of JOINT motion will be different from conventional JOINT motion if the taught position is represented in Cartesian space (e.g. xyzwpr). **Warning** **Motion behavior might be different from conventional joint motion. Please be cautious when executing the program. You could injure personnel or damage equipment.**
No singularity avoidance header is displayed in the teach pendant program.	$ra_paramgrp. $enable=FALSE	Set $ra_paramgrp[1]. $enable=TRUE and then turn the controller off and back on for the change to take effect.
	The teach pendant program does not contain a motion group that supports singularity avoidance function.	There is no need to display the header.
	$racfg. $no_header=TRUE	Set $racfg. $no_header=FALSE

Table 10–5. Troubleshooting Solutions (Cont'd)

Problem	Possible Cause	Solution
In singularity	Different type of singularity occurs (for example, not a wrist singularity).	This function is not helpful for the singularity like that.
	Singularity occurs at a Circular motion.	Rearrange the taught point(s).
	Singularity occurs at a corner path but cannot be predicted on a segment basis.	Use WJNT motion option for the motion line and manually set an appropriate flip/non-flip configuration for the destination position.
When taught point is inside a predefined singularity zone, it is not reached.	When $ra_paramgrp[]. $use_strt_j4=TRUE, the system will change the taught orientation internally at the taught point that is inside the singularity zone to limit the change in axis 4 and to maintain the taught TCP location. Under T1/T2 mode, the system will post a warning message "Modify Singular Dest (L:%d)" in this case.	Touch up the reached point as a taught point or set $ra_paramgrp[]. $use_strt_j4=FALSE.

10.4 AUTOMATIC COLLISION RECOVERY

10.4.1 Overview

Automatic Collision Recovery uses both the *Collision Guard* and *High Speed Skip* software options to recover automatically from the SRVO-050 Collision Detect error. Without Automatic Collision Recovery, when a collision detect error occurs, the robot program pauses with a SRVO-050 alarm. If Automatic Collision Recovery is enabled, the robot program will detect the SRVO-050 alarm, RESET the alarm and CONTINUE the robot operation from a user-specified LABEL of the robot program. It is the user's responsibility to handle the error recovery when the program is resumed. When Automatic Collision Recovery is installed, it will automatically install the High Speed Skip software option to the robot. However, the Collision Guard software option must be purchased and installed on the robot before using Automatic Collision Recovery.

Warning

Inappropriate use of Automatic Collision Recovery can damage the robot, end-of-arm tooling, and objects in the work cell; and can injure personnel. Use the SKIP function and resume the program correctly through safe recovery procedures.

Auto Collision Recovery can be used in two different modes:

- Mode 1: The robot program continues the production program from a user-specified label after a collision.
- Mode 2: The robot program runs a user recovery program (specified by the user) after a collision.

10.4.2 Enabling and Disabling Automatic Collision Recovery

Automatic Collision Recovery is enabled by default after it is installed.

To disable Automatic Collision Recovery, set the variable [HTCOLREC]COL_REC = FALSE.

You can also execute the program HTCOLSET with the argument 0 or 1 to disable or enable Automatic Collision Recovery. For example, CALL HTCOLSET(1) is the same as setting COL_REC = TRUE.

10.4.3 Automatic Collision Recovery Setup

Automatic Collision Recovery can be used in two different modes:

- **Mode 1** (See Program EXAMPLE1 in Section 10.4.4) - The robot program continues the production program from a user specified label after a collision. This is the default mode.
- **Mode 2** (See Program EXAMPLE2 in Section 10.4.5) - The robot program runs a user-specified recovery program (created by the user) after a collision.

Setup Tasks

Before you can use Automatic Collision Recovery you must complete the following tasks:

- Set up robot payload information. Refer to the “Robot Payload Setting” section in your application-specific *Setup and Operations Manual*.
- Set up Collision Guard Sensitivity and enable Collision Guard. Refer to the “Collision Guard” section in your application-specific *Setup and Operations Manual*.
- Verify that a SRVO-050 error is generated by the robot when a collision occurs.
- Set up the Auto Collision Recovery variable data. Refer to Table 10–6 and Procedure 10-8 .

- Write a test program (see examples in Section 10.4.4 and Section 10.4.5).
- Run the test program to verify the collision recovery process.

Table 10–6. Automatic Collision Recovery Variable Data

Item	Descriptions
REG_INDEX	This item sets the Register Index, which will be set to 1 during collision recovery. Make sure the value is within the range of the numeric registers available on the controller.
DO_INDEX	This item sets the Digital Output to turn ON during Collision recovery. Make sure the value is within the range of the available digital outputs on the controller.
ERR_DO_INDEX	This item sets a Digital Output to turn ON if an unexpected error occurs during Automatic Collision Recovery.
RECOVERY_TSK	For Mode 2, enter the name of the program you want to run when the collision detect error occurs. **Do not enter the .TP or .pc extension.**
ERR_DELAY (ms) Default: 2000	This item is the wait time *after* the error occurs but *before* the RESET is issued. Adjust the value if necessary, depending on your cycle time and safety requirements.
RESET_DELAY (ms) Default: 5000	This item is the wait time from when Automatic Collision Recovery issues the RESET until the program that triggered collision detect is resumed. This delay is necessary to wait for the servos to re-start. Adjust the value if necessary, depending on your cycle time and safety requirements. **Note** If you lower the RESET_DELAY too much, you will get an error when the program attempts to resume before the servos re-start.
USE_REC_TASK	For Mode 2, set to TRUE .

Procedure 10-8 Setting Auto Collision Recovery Variable Data

The Automatic Collision Recovery variables are defined in a program called HTCOLREC.PC. Set up these variables according to the requirements of your application:

1. Press SELECT and move the cursor until you see HTCOLREC.PC.
2. Press ENTER.
3. Press DATA.
4. Press F1, TYPE and select KAREL Vars. You will see a screen similar to the following:

```
1 COL_REC       TRUE
2 COL_RECOV     AUTO_COL_REC
3 COL_DBG       FALSE
4 ABORT_DELAY   16
```

5. Move the cursor to COL_RECOV and press F2, DETAIL. You will see a menu similar to the following:

```
COL_RECOV                          1/9
  1 REG_INDEX       1
  2 DO_INDEX        1
  3 ERR_DO_INDEX    2
  4 RECOVERY_TSK    *uninit*   <- For Mode 2, specify your recovery
  5 ERR_DELAY       2000          program name here.  See EXAMPLE2.
  6 RESET_DELAY     5000
  7 USE_REC_TSK     FALSE
  8 $DUMMY7         255
  9 $DUMMY8         255
```

6. Set up Auto Collision Recovery variable data per your application requirements.

10.4.4 Mode 1 of Automatic Collision Recovery

In Mode 1, the Robot program continues the production program from a user-specified label after a collision. Mode 1 is the default mode for Automatic Collision Recovery.

An example program that detects the collision error is shown below.

```
/PROG  EXAMPLE1
   1:  LBL[10] ;
   2:  CALL HTCOLSET(1) ;…….enable collision recovery process
   3:  SKIP CONDITION ERR_NUM=11050      ;
   4:J P[1:APPROACH] 25% CNT100    ;
   5:L P[2:PICK POS] 100mm/sec FINE Skip,LBL[100],PR[3:Clear]=LPOS    ;
   6:  ! Recovered from Collision at PR[3]
   7:  CALL HTCOLSET(0) ;……disable collision recovery process
...8:  RO[1:GRIPPER]=OFF ; --open gripper
   9:  !add recovery actions here ;
  10:  ! add recovery actions here ;;
  11:  JMP LBL[10] ;……..repeat pick up operation
  12:  LBL[100] ;……continues if there is no collision
```

```
13:  CALL HTCOLSET(0) ;……disable collision recovery process
14:  RO[1:GRIPPER]=ON ;---close gripper
15:J P[3:RETRACT] 25% CNT100    ;
16:  LBL[200]
```

Programming Considerations

In this program, a SKIP condition has been set up to trigger when the ”SRVO-050 Quick Stop” error occurs. On line 5, during the motion to P[2], the high-speed skip function is used. Refer to the “Quick Skip” section in your application-specific *Setup and Operations Manual* for more information on the high-speed skip function.

- If the SKIP condition is triggered, the current position will be recorded into Position Register PR[3], and the next line will be executed. This is where we perform recovery actions and then jump to LBL[10] to repeat the operation.
- If SKIP is not triggered, at the end of the motion the program will branch to LBL[100] in line 12, where it completes the operation successfully and terminates the program.

In the case of a STOP error such as Quick Stop, the execution will pause at the next line. Automatic Collision Recovery, which is running in the background, sees the STOP error and attempts to resume the program. When program execution resumes, you must perform appropriate recovery procedures, such as moving the robot at slow speed from the current position to a safe position.

If you do not program a high-speed SKIP instruction, Automatic Collision Recovery will resume the program from where it stopped. This will probably trigger more collision errors. Careful handling of collision errors through the SKIP function is extremely important.



Caution

If a collision occurs during Automatic Collision Recovery, collision recovery will be attempted again. Limit the number of times this happens, otherwise you might damage the robot, tooling, and fixtures.

When the robot is performing collision recovery, it is possible to have a digital output turned ON and/or a numeric register set to 1. You set these values when you set up Automatic Collision Recovery variable data. Refer to Procedure 10-8 .

When a collision occurs after these variables have been set, the register will be set to 1 and the Digital Output will be turned on. Your program’s error handling logic must re-set the register and/or digital output; the HandlingTool software does not reset these registers or outputs.

You can use the register value or digital output value in your program to determine if the robot has encountered a collision and to branch appropriately. It is expected that you will reset the register to zero and set the digital output to OFF.

Testing

Test your program thoroughly at low speed to verify the collision recovery process.

10.4.5 Mode 2 of Automatic Collision Recovery

In Automatic Collision Recovery Mode 2, the robot program runs a user-specified recovery program (created by the user) after a collision.

In some applications, when a collision detect error occurs, it is better to abort all programs, run an error recovery program, and then re-start the original program. The original program can be written to branch based on the state of the Digital Output defined by DO_INDEX or the value of the numeric register defined by REG_INDEX. When the same program is run in recovery mode, it will do the appropriate branching in the program to recover smoothly.

To force Automatic Collision Recovery to follow Mode 2, be sure you have

- Set RECOVERY_TASK to the name of the program you want to run when the collision detect error occurs. Enter only the program name; do not enter the ".TP" or ".pc" extension.
- Set USE_REC_TASK to TRUE.

For example, if you set RECOVERY_TASK to a program you have called ERR_REC, in this mode all the running programs are aborted. Then ERR_REC is run, and when it is finished, the main program is run again.

You might also need to adjust the value of the variable [HTCOLREC]ABORT_DELAY (a main variable in HTCOLREC.PC). The value of ABORT_DELAY is the amount of time the robot will wait after doing an ABORT ALL and issuing a RUN command to run the original program. Although the original program aborts and runs internally, you must be aware that the controller requires a small amount of time to abort all the tasks. The default value is 16 ms. If the delay is reduced to zero, the original program which goes into a paused state when collision occurs, might not be aborted but will be requested to re-start. This would result in an error.

To adjust the value of [HTCOLREC]ABORT_DELAY,

1. At the SELECT menu, move the cursor to HTCOLREC.PC and press ENTER.
2. Press DATA to display the program variables. Set ABORT_DELAY as desired.

The following Mode 2 example program calls a recovery program called ERR_REC, created by the user.

```
/PROG  EXAMPLE2
   1:  LBL[10] ;
   2:  CALL HTCOLSET(1) ;…….enable collision recovery process
```

```
   3:  SKIP CONDITION ERR_NUM=11050   ;
   4:J P[1:APPROACH] 25% CNT100    ;
   5:L P[2:PICK POS] 100mm/sec FINE Skip,LBL[100],PR[3:Clear]=LPOS    ;
   6:  ! Recovered from Collision at PR[3]
   7:  CALL HTCOLSET(0) ;……disable collision recovery process
...8:  RO[1:GRIPPER]=OFF ; --open gripper
   9:  CALL REC_PROG ;…….recovery program is after collision
  10:  ! add recovery actions here ;;
  11:  JMP LBL[10] ;……..repeat pick up operation
  12:  LBL[100] ;……continues if there is no collision
  13:  CALL HTCOLSET(0) ;……disable collision recovery process
  14:  RO[1:GRIPPER]=ON ;---close gripper
  15:J P[3:RETRACT] 25% CNT100    ;
  16:  LBL[200]
```

Testing

Test your program thoroughly at low speed to verify the collision recovery process.

10.4.6 Limitations

The following limitations apply to Automatic Collision Recovery.

Caution

If a collision occurs when the robot is in collision recovery, the robot will attempt collision recovery again. Do not let this happen repeatedly, or you could damage the robot, tooling, and fixtures.

Warning

Using Automatic Collision Recovery inappropriately can cause injury to personnel and damage to the robot, end-of-arm tooling, and objects in the workcell. Use safe recovery procedures to resume the program correctly.

- Automatic Collision Recovery requires Collision Guard to be installed on the robot. You must set the sensitivity of Collision Guard appropriately.
- Move the robot at slow speeds when you expect a collision for which you plan to use Automatic Collision Recovery.
- Automatic Collision Recovery is supported in Production Mode only.

- Automatic Collision Recovery does not work and is not supported when you are running programs through SHIFT+FWD with the TP enabled. The TP enable key must be OFF in order for this feature to work.
- The collision detect error stops the robot and pauses the program that caused the error. During automatic error recovery, **all** paused tasks are resumed. If a user task (KAREL or TP) is paused at the time of the collision detect error, that task will also be resumed. You should note this limitation and safeguard your programs when they resume due to Automatic Collision Recovery. Suggested safeguards include the following:
  - — Set up a register value to be = 1 or a Digital Output to be ON when a collision occurs. Your program should check these values upon program resumption in all your tasks.
  - — If your application programs also manage programs paused due to error, you might have to modify the programs, because Automatic Error Recovery resumes **all** paused tasks.
- Use the SKIP function and handle the resuming of the program correctly through safe recovery procedures. If Automatic Collision Recovery is used inappropriately, the robot, end-of-arm tooling, and objects in the workcell could be damaged and personnel could be injured.
- If a collision occurs when the robot is in collision recovery, the robot will attempt collision recovery again. If this happens repeatedly you could damage the robot, tooling, and fixtures.
- If the program that caused collision detect is aborted by the user or the controller due to internal safety reasons, collision recovery will not occur. For collision recovery to execute, the program must be in a paused state after collision detect.
- While running a program from KCL (KAREL Command Language), Automatic Collision Recovery might try to resume a different program than the one you expect. To make sure that Automatic Collision Recovery resumes the correct task, you must either select the program from the SELECT menu or set the system variable, $TP_DEFPROG to the program you are running from KCL.

10.5 BASIC PROCESS AXES (OPTION)

10.5.1 Overview

The Basic Process Axes software option (R689) provides the user with the capability of configuring and operating a motor or motors independent of any motion group. A “Process Axis” has the following characteristics:

- The axis accepts only speed commands; it’s position cannot be commanded directly
- The axis does not belong to any motion group

Process axes are intended to be used to drive devices for which position control is not needed, e.g. a pump or a drill bit.

Process axes are also used by other software options, which integrate the operation of the process axes directly into the application tool itself. Since these options access the process axes directly, they cannot be loaded on the controller with the Basic Process Axes option. That is, Basic Process Axes is mutually exclusive with these options.

The software options that are mutually exclusive with Basic Process Axes are as follows:

- Integral Pump Control (R554)
- FRA Waterborne (R556)
- ISD Gear Meter (R563)
- ISD Dual Meter (R562)
- ISD Single Acting (R564)
- ISD Double Acting (R561)
- Pneumatic Servo (R595)
- Servo Torch (J691, J982)
- Pump Only Motion (J801)

Accordingly, the Basic Process Axes option should be loaded and used in simple, generic cases when the use of a process axis is desired but it does not fit one of the above options. With this option, the process axes can be controlled by TP instructions or by group inputs.

The Basic Process Axes option provides the following functions:

- The ability to setup and configure process axes, up to a maximum of 8.
- TP instructions to send a speed command to the axes, and to stop the axes.
- A torque monitoring function to post an alarm and stop program execution and robot motion when the process axes encounter excessive torque.
- A setup menu to specify a scale factor for speed commands, torque limit values, group outputs for speed data, and group inputs for speed commands.
- A status screen to display speed and torque data for the process axes.

10.5.2 Setup

The Basic Process Axes option has a setup menu available to configure its features. The setup menu covers the following items:

- Torque Monitoring function
- Scale factor for the speed value in the TP instruction
- Group Input for speed command

- Group Output for speed display

Refer to table Table 10–7 for further details on the items included in the setup menu. Use procedure Procedure 10-9 to setup the Basic Process Axes option.

Table 10–7. Basic Process Axes setup menu items

ITEM	DESCRIPTION
Process Axis Number default: 1	This item indicates which process axis corresponds to the data being displayed in the menu. This field cannot be edited directly, but is changed by pressing F3 [AXIS] and entering the new axis number.
Current Limit default: 0.0	This item defines a maximum value for the threshold of the commanded current of the process axis motor, to be used by the torque limit function. A value of 0.0 indicates that the torque limiting function is disabled. See the section on torque limiting for more details
Cur Lim fault after default: 0	This item is also part of the torque limiting function and defines the number of milliseconds for which the current value indicated in the “Current Limit” item must persist before an alarm is posted.
Scale factor for TP Inst default: 1.0	This item is used in conjunction with the next item (gear ratio) to determine the relationship between the speed indicated in the SET ISDT SPEED instruction and the RPM at the motor shaft.
Current gear ratio is default: (none)	This item is included for display purposes only and must be set from within the Process Axis Setup routine at controlled start. (Further details can be found in the Software Installation manual).
Speed Command Group Input default: 0	This item defines a group input number that can be used to set the speed command for this axis. Note that cycling power is required for changes in this item to take effect.
Actual Speed Group Output default: 0	This item defines a group output to display the current motor RPM for this process axis. Note that cycling power is required for changes in this item to take effect.

Procedure 10-9 Setting up Basic Process Axes

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Process Axes. You will see a screen similar to the following

```
SETUP ISDT
                                        1/5
      Process Axis Number:  1

      Current Limit:               0.0 amps
      Curr Lim fault after          0 msec
      Scale factor for TP Inst:    1.0
      Note:  Scale factor is applied in
      addition to gear ratio!!
      (Current gear ratio is  10:1 )
      Speed Command Group Input: 0
      Actual Speed Group Output: 0
```

5. **To change the process axis number being setup**, press F3, AXIS, and enter the desired process axis number. If the number entered does not correspond to a valid, properly setup process axis, the following message will be displayed:

```
Invalid Axis Number!
```

6. **To display help information**, press F2, HELP. When you are finished displaying help, press PREV.
7. Move the cursor to the item you wish to change, and enter the new value for that item.

If a nonzero value is set for “Speed Command Group Input” in this setup menu, then the resulting behavior of that process axis will be as follows:

- The group input will be monitored continuously, so that the value of the input will be the instantaneous speed command to the process axis motor.
- The value stored in the group input will be treated as a 16–bit integer, converted to RPM at the motor shaft as follows:
  - — The most significant bit will be treated as the sign bit, where 0 means positive and 1 means negative. (i.e. for a negative speed value add 32768).
  - — The value designated by the remaining 15 bits will be multiplied by 10 to obtain the RPM to be commanded to the motor.

- The process axis that uses this group input number *will be in motion* as long as the following conditions are met:
  - — The axis' servo amplifier is enabled
  - — The axis' brake is released
  - — The group input has a nonzero value

For example, if a value of "11" were entered for "Speed Command Group Input" for process axis #1 in the setup menu, then in order to drive the process axis #1 motor at 1000 RPM (at the motor shaft), the group input #11 must be set to a value of 10000. If instead a speed of —1000 RPM were desired, then group input #11 should have the value 42768, which is 10000 + 32768

Note If a group input is setup to be used as the speed command for *any* process axes, then the TP instructions are automatically disabled. That is, TP instructions and Group Inputs cannot be simultaneously used to control the process axes.

If a nonzero value is set for "Actual Speed Group Output" in the setup menu, then the actual RPM at the motor shaft of the process axis will be copied in real time to that group output, as a 16–bit integer value, with the most significant bit representing the sign bit. As with the speed command group input, when the most significant bit is 1, then the group output is indicating a negative speed value.

For example, if a value of "12" were entered for "Actual Speed Group Output" for process axis #1 in the setup menu, then if group output 11 displays a value of 1000, it means that process axis #1 has a speed of 1000 RPM at the motor shaft. If group output 11 displays 33768, then the speed of process axis #1 is —1000 at the motor shaft.

10.5.3 TP Instructions

The Basic Process Axes option includes two teach pendant instructions for setting process axes speeds:

- SET ISDT SPEED
- STOP ALL ISDT

10.5.3.1 SET ISDT SPEED Instruction

The SET ISDT SPEED instruction is used to send speed commands directly to the process axes and can include speed data for up to 8 axes. The instruction allows two fields for each axis, one for axis number and one for the speed value. The following screen is an example of the usage of this instruction:

```
TEST
                                          2/3
         1:J  P[1] 100% FINE
         2:  SET ISDT SPEED A1=100
         3:J  P[2] 100% FINE
      [End]




      Enter Value

       ADD AX
```

In this example, the cursor is on the field corresponding to the axis number (“1” in the example). To the right of this field, following the “=” is the speed value that will be sent to process axis #1, which in this example is 100 user speed units. (See section Section 10.5.2 for more details on the relationship between user speed units and motor RPM). If this instruction needs to set speeds for additional process axis, pressing F1 (labled “ADD AX”) will add an additional set of fields to the instruction as shown in the following screen:

```
TEST
                                          2/3
         1:J  P[1] 100% FINE
         2:  SET ISDT SPEED A1=100, A...=...
         3:J  P[2] 100% FINE
      [End]




      Enter Value

       ADD AX    DIRECT    INDIRECT
```

The cursor can then be moved to the “...” fields to enter a new axis number and speed value. Note also in the above screen that the cursor is on the speed field, which means that function keys F2 and F3 are available to allow the speed value to be specified as a direct integer value or indirectly via numeric register.

There is no error checking performed on the fields of the SET ISDT SPEED instruction at the time the instruction is added to, or edited in, a TP program. However, error checking is performed at the time the instruction is actually executed. For each field in the instruction, first the axis number will be checked to verify that it is a properly setup process axis. If that process axis has not been setup, then the following alarm will be posted:

```
ISDT-012  Invalid Axis # in L:# Instr
```

The error text will indicate the process axis number in the instruction that is invalid, as well as the line number in the TP program containing the offending instruction. If the same process axis number is used in multiple fields in the instruction, then the following alarm will be posted:

```
ISDT-013  Duplicate Ax:# in L:# Instr
```

The alarm text will indicate which axis number was repeated, and also the line number from the TP program. If the speed value specified exceeds the speed limit of that process axis (as specified in the process axis setup routine at controlled start), then the following alarm will be posted:

```
ISDT-001  Joint speed limit (G:0 A:#)
```

If the speed value specified results in a motor RPM that exceeds the maximum RPM for this motor, then the following alarm is posted:

```
ISDT-002  Motor speed limit (G:0 A:#)
```

Finally, if there is a process axis that has been properly setup at controlled start but that does not have a corresponding field in the SET ISDT SPEED instruction, then when the instruction is executed that process axis will remain unaffected: it will continue moving at its current speed, or remain stopped if it was stopped.

10.5.3.2 STOP ALL ISDT Instruction

The STOP ALL ISDT instruction is used to immediately send a zero speed to all properly setup process axis. It does not have any arguments. An example of the usage of this instruction appears in the screen shot below:

```
TEST
                                          2/3
         1:J  P[1] 100% FINE
         2:  SET ISDT SPEED A1=100, A2=200
         3:J  P[2] 100% FINE
         4:  SET ISDT SPEED A2=0
         5:J  P[3] 100% FINE
         6:  SET ISDT SPEED A1=50
         7:J  P[4] 100% FINE
         8:  SET ISDT SPEED A1=100, A2=50
         9:J  P[5] 100% FINE
         10:  STOP ALL ISDT
      [End]
```

In this example program, once the robot has moved to P[1], process axis #1 will begin moving at speed 100 while process axis #2 will begin moving at speed 200. Then the robot will move to P[2], after which process axis #2 will stop (i.e. speed will change to 0), but process axis #1 will continue moving at speed 100. Next, after the robot moves to P[3], process axis #1 will change its speed from 100 to 50, while process axis #2 will remain stopped. Then the robot will move to P[4], after which both process axes will change speed, process axis #1 from 50 to 100 and process axis #2 from 0 to 50. Then the robot will move to P[5], and finally on the last line of the program both process axis #1 and #2 will stop.

10.5.4 PROCESS AXES STATUS (OPTION)

The Process Axes Status screen displays speed and torque data for all process axes defined on the controller. This screen is only available when the Basic Process Axes option is loaded on the controller. See figure Figure 10–9 for an example screen and table Table 10–8 for a description of the items on this screen.

Figure 10–9. Process Axes Status Screen

```
STATUS ISDT

             User    Motor     Motor
      Axis   Speed   RPM       Amps
        1      0.0     0.0      0.00
        2     51.2   301.0      1.41
        3   ******  ******    ******
        4   ******  ******    ******
        5   ******  ******    ******
        6   ******  ******    ******
        7   ******  ******    ******
        8   ******  ******    ******
```

Table 10–8. Process Axes Status Items

ITEM	DESCRIPTION
Axis	This item is the axis number whose data is displayed in this row.
User Speed	This item is the current speed of the axis in user units (i.e. units of the TP instruction). A "******" indicates that the axis is not defined.

Table 10–8. Process Axes Status Items (Cont'd)

Motor RPM	This item is the current speed of the axis in RPM at the motor shaft. A "******" indicates that the axis is not defined.
Motor Amps	This item is the commanded current in amps from the servo amplifier for this axis. A "******" indicates that the axis is not defined.

10.5.5 Torque Monitor Function

The Basic Process Axes option includes a torque monitor function, which allows the user to specify a level of torque (more specifically, of commanded motor current) and a corresponding length of time beyond which a stop error will be posted. This is useful for protecting equipment connected to the process axis in such a case as the torque capacity of the motor exceeds that of the hardware being driven.

The torque monitor function is setup and enabled by the first two items in the process axis setup screen (see section for further details). These items are the current limit and the fault time. The current limit is specified in amps and represents an absolute value of the commanded current from the servo amplifier. The fault time is specified in milliseconds and represents the amount of time for which the absolute value of the commanded motor current must exceed the specified current limit for the alarm to be posted.

When the commanded motor current has exceed the "Current limit" value from the Process Axis Setup menu for at least the number of milliseconds defined by "Curr Lim fault after" in the setup menu, then the following alarm will be posted:

```
ISDT-014  Torque Limit on Ax:#
```

This alarm will stop program execution and robot motion, in addition to stopping the process axes.

If the commanded motor current exceeds the current limit for only a brief period of time, specifically a period of time that is less than "Curr Lim fault after" in the setup menu, and then returns to a value below the limit, then no alarm will occur.

If a value of zero is specified under "Curr Lim fault after" in the setup menu, then the alarm will be posted immediately when the commanded current exceeds the limit.

If a value of zero is specified for the current limit in the setup menu, then the torque monitor function is disabled and motor current will not be checked.

10.6 BRAKE CHECK FUNCTION

10.6.1 Overview

Brake check function (A05B-2500-J951) is a preventive maintenance function. It is used to check that the motor does not move when motor torque is applied while brake is engaged. This allows user to perform early detection of potential brake failure. User is informed of the checking result by warning message or system variables.

10.6.2 Setup

10.6.2.1 Enable/Disable of Brake Check function

For robots that have been tuned for Brake Check function, it is enabled by default once the option is loaded. The system variable that enables/disables brake check function is:

$BCCFG.$BCK_ENABLE TRUE: Enable (default) / FALSE: Disable

Please confirm this variable is TRUE (Enable) before use. If this variable is FALSE (Disable), brake check will not start.

10.6.2.2 Enable/Disable switch for each axis

User can selectively enable/disable brake check for individual axis. To execute brake check on a selected axis, the following system variable needs to be TRUE (Enable):

$BCK_GRP[group].$CHK_AXIS[axis] TRUE: Enable / FALSE: Disable

On robots that are tuned for Brake Check function, the default value is TRUE (Enable). For non-robot axis (such as aux axis, servo gun, ... etc.), the default value is FALSE (Disable).

To execute brake check on dual drive axis, this variable needs to be set TRUE (Enable) for both master and slave axis.

10.6.2.3 Setting Brake Torque

Brake Check function requires brake torque information for each axis in order to determine torque command during brake check. The following system variable needs to be set with the brake torque value [kgf*cm] for each axis.

$BCK_GRP[group].$BRK_TORQUE[axis] brake torque [kgf*cm]

On robots that are tuned for Brake Check function (including robots that use Dual Drive), this variable is set automatically. However, on non-robot axis (such as aux axis, servo gun, or etc.), this variable is not set. Manual setting is required if brake check is needed on non-robot axis. Please refer to AC SERVO MOTOR DESCRIPTIONS "9.1 BRAKE SPECIFICATIONS" on specification value of brake torque. Please note that the unit of this variable is [kgf*cm].

If brake check is required on non-robot Independent Dual Drive axis, please set this variable as follows:

Case-1: Either master or slave motor has one brake.

Set this variable to the brake torque value [kgf*cm] on only the axis that has brake. For the axis that has no brake, please set 0.0 (default).

Case-2: Both master and slave motors have brakes.

Set this variable to the brake torque value [kgf*cm] on both master and slave axes. The brake torque value of both the master and the slave axis must be same. Brake Check function does not support different brake torque value in this case.

10.6.3 Start Brake Check

To start brake check, please set the following system variable to TRUE.

$BCCFG.$BCK_START TRUE: Brake check is running / FALSE: Brake check is finished

User can perform brake check by using parameter instruction of Teach Pendant program or by manually setting system variable. This variable will automatically return to FALSE when brake check is completed. Please do not move the robot until checking is completed (this variable automatically returns to FALSE). If brake check function is disabled ($BCCFG.$BCK_ENABLE=FALSE), brake check will not be performed, and $BCCFG.$BCK_START will immediately return to FALSE even if this variable is set to TRUE.

Sample Teach Pendant program that executes brake check:

```
  1: J P[1] 100% FINE
2: L P[2] 2000mm/sec FINE
3: WAIT 1.00 (sec)
4: $BCCFG.$BCK_START=1
5: WAIT $BCCFG.$BCK_START=0
6: L P[3] 2000mm/sec FINE
```

Brake check cannot be started while the robot is moving, so this program waits for robot to stop completely by WAIT instruction on L3. Then, this program starts brake check at L4 and waits until

brake check is completed at L5. As previously described, brake check can only be executed while robot is completely stopped.

Note In case of brake failure, the motor can move during brake check. Therefore, it is important that robot is at a position with sufficient clearance. Total motion command per single check varies by robot and by axis. However, it is up to 200,000[pulse]. This is about 0.4[rev] in motor revolution and about 1.4[deg] in degree of joint angle assuming a gear ratio of 1/100.

In case power is shut OFF during brake check, the following alarm will be posted at next power ON. This alarm indicates that there was abnormal termination at last check.

BRCH-003 Last check was failed

The following system variable will be also set to TRUE at next power ON.

$BCCFG.$LAST_FAILED TRUE: Last check was failed / FALSE: Last check completed normally

In this case, the results of brake check will not be generated, therefore please execute brake check again at next power ON. If next brake check completes normally, this variable will return to FALSE automatically.

10.6.4 Result of Brake Check

$BCCFG.$BCK_START automatically returns to FALSE when brake check completes. The result is also posted as warning message. If all brake passes as “Good” brake, the following message will be posted.

BRCH-001 All brake are normal

By contrast, if brake failure is detected as “Bad” brake, the following message will be posted.

Ex) Brake failure is detected on group1/J2.

BRCH-002 Brake abnormal (G:1,A:2)

The results are also recorded in the following system variable at the same time. The value of this variable is kept until next brake check starts.

$BCK_GRP[group].$CHK_RESULT[axis] TRUE: Brake failure / FALSE: Normal brake (default)

10.6.5 Limitations

Brake check function cannot be executed under the following conditions, or alarm will be posted.

- Calibration has not been done.

- Robot is moving.
- Servo is OFF.
- Servo parameter is being updated.
- Dual motor is in adjust mode.
- Even though the axis has brake and $BCK_GRP[group].$CHK_AXIS[axis] is TRUE, brake torque ($BCK_GRP[group].$BRK_TORQUE[axis]) is not set.
- Teach Pendant is enabled.

10.6.6 Caution

- It takes about 10 sec for single brake check execution.
- When the motor is heating up, the motor torque can decrease, so it may not be able to detect brake failure. To detect it with high accuracy, please execute brake check preferably when the motor is in low-temperature, e.g. first thing in the morning.

10.7 BUMP BOX

10.7.1 Overview

Bump Box is an optional feature that allows you to offset a programmed robot path or the shape center when used in conjunction with the Shape Generation option. The offsets can be applied from the teach pendant or by setting the digital inputs. The offsets can be applied only in x, y, or z directions. You have the option to remove the offsets applied by Bump Box at any time.

All the information needed to apply offsets (*bump*) is contained in a Bump Box schedule which must be set up before using Bump Box. In order to apply the offsets by setting digital inputs, Bump Box I/O must be set up properly.

Bump Box applies offsets to a path in the following ways:

- You set up the Bump Box schedules that contain the information such as the program name and the line numbers needed to apply the offsets.
- You set up the Bump Box I/O which includes the digital inputs required for applying the offsets and a group input required for selecting the Bump Box schedule.
- When you apply the offsets from the teach pendant or set a digital input, based on the information contained in the schedule, the position data in the teach pendant program is modified. The robot will move along the bumped positions.
- When you remove the offsets, the original positions in the teach pendant program are restored.

10.7.2 Setting Up Bump Box Schedules

You need to set up the Bump Box schedules and I/O before you can apply offsets to a path. You set up the schedules from the Bump Box schedules screen. You can define up to 255 bump box schedules. By default, 15 bump box schedules are available. You can change the number of schedules by modifying the system variable $BB_TOT_SCH. Use Procedure 10-10 to change the number of bump schedules. Use Procedure 10-11 to set up a bump box schedule.

Bump Box Schedule Screen

The bump box schedule screen allows you to view limited information for several schedules at once. DETAIL allows you to view and set the complete information for a single schedule. See Table 10–9 for a description of the items on the Bump Box DETAIL screen.

Table 10–9. Bump Box Schedule Items

ITEM	DESCRIPTION
Program name	This item displays the name of the program you want to bump.
Group Number	This item is the group number to be bumped. Only one group can be specified in a bump box schedule. The group must be a robot group.
CutID NOTE By default, this item and the choices for Offsets frame are called "cut," which means cutting applications. This can be changed by changing the system variable $BBAPPL_ID.	This item is significant only when shape generation is loaded. It displays the shape or cut number in the program followed by the line numbers corresponding to shape macros. If there is a "CALC xxxx" macro call in the program, the first line number corresponds to the macro and the second line number corresponds to the motion line that has the shape center position. If the program has PROCESS ON/PROCESS OFF macro calls, the line numbers correspond to these macros. If there are no macro calls, or if shape generation is not loaded, this item displays *****.
WeldID (only for ArcTool)	This item displays the weld number in the program followed by the Arc Start and Arc End line numbers for that weld in square brackets. The weld in each program are numbered serially.
Lines	This item displays the range of lines that the offsets will be applied to. You can edit these items as necessary.

Table 10–9. Bump Box Schedule Items (Cont'd)

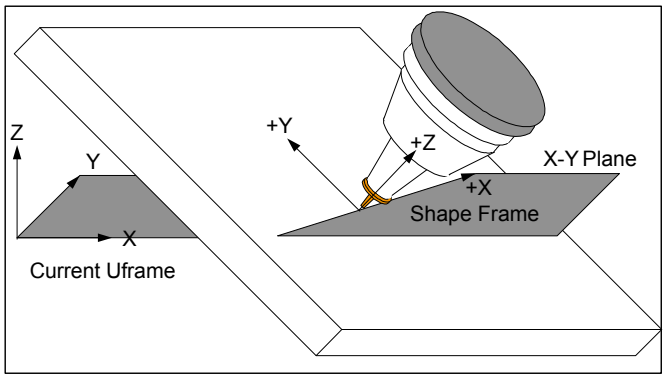
ITEM	DESCRIPTION
Offsets frame **Note** The choices for this item are available when the shape generation option is loaded.	This item allows you to select a frame of reference for applying offsets. • Shape frame–A shape frame is calculated from a shape center position and current Uframe. The shame frame x-direction is parallel to the object surface where the current Uframe x-y plane intersects the object at the shape center position. The shape frame direction is also parallel to the object surface but perpendicular to the x-direction. The shape frame z-direction is into the tool along the Tool-z vector. See the following figure.  • World/Uframe– In "World/Uframe" relative, the offsets are applied directly to the x, y, and z components of the positions. The World/Uframe of reference allows you to apply offsets to the positions in the following ways: — X-offset — Y-offset — Z-offset

Table 10–9. Bump Box Schedule Items (Cont'd)

ITEM	DESCRIPTION
Offset frame for ArcTool only	This item allows you to select a frame of reference for applying offsets. These are three choices: • **Weld & Tool** — This is the default frame of reference. The first and second points of the taught weld path determine the weld direction. X-offset will be along the path, positive direction will be from first point to the second point. Z-offset will be along the Tool Z, positive direction will be away from the weld into the torch. Y direction will be perpendicular to both X and Z. If the lines specified in the schedule involve circular motion, then the start point and end point of the circular path are used for determining the offset directions. The Weld & Tool frame of reference allows you to apply offsets to the positions in the following ways: — Along (in the X direction) — Across (in the Y direction) — Stickout (in the Z direction) • **Weld** — This item determines the weld direction. In "Weld" relative, the first and second points of the taught weld path determine the weld direction. The X-offset will be along the path, and the positive direction will be from first point to the second point. The Z-offset will be along the World/Uframe Z which fixes +Y to the left of the weld. The Weld frame of reference allows you to apply offsets to the positions in the following ways: — Along (in the X direction) — Across (in the Y direction) — Z-offset (in the Z direction) • **World/Uframe** — In "World/Uframe" relative, the offsets are applied directly to the x, y, and z components of the positions. The World/Uframe of reference allows you to apply offsets to the positions in the following ways: — X-offset — Y-offset — Z-offset

Table 10–9. Bump Box Schedule Items (Cont'd)

ITEM	DESCRIPTION
Offsets frame **Note** The choices for this item are available when the shape generation option is not loaded.	This item allows you to select a frame of reference for applying offsets. • **Cut & Tool** - This is the default frame of reference. The first and second points of the taught path determine the path direction. The x-offset will be along the path; the positive direction will be from the first point to the second point. The z-offset will be along the Tool z; the positive direction will be into the torch. The y direction will be perpendicular to both x and z. If the lines specified in the schedule involve circular motion, then the start point and end point of the circular path are used for determining the offset directions. The Cut & Tool frame of reference allows you to apply offsets to the positions in the following ways: • Along (in the x direction) • Across (in the y direction) • Stickout (in the z direction) • **Cut** - The first and second points determine the path direction. The x-offset will be along the path, and the positive direction will be from first point to the second point. The z-offset will be along the World/Uframe z-axis, which fixes +y to the left of the path. The Cut frame of reference allows you to apply offsets to the positions in the following ways: • Along (in the x direction) • Across (in the y direction) • Z-offset (in the z direction) • **World/Uframe** - In "World/Uframe" relative, the offsets are applied directly to the x, y and z components of the positions. The World/Uframe of reference allows you to apply offsets to the positions in the following ways: • X-offset • Y-offset • Z-offset
Offset increment	This item is the amount the positions will be offset for each bump that you apply from the teach pendant or by setting a digital input.

Table 10–9. Bump Box Schedule Items (Cont'd)

ITEM	DESCRIPTION
Applied	The Applied column displays the offsets applied in the z (Along), y (across) and z (stickout) directions. When the cursor is in this column, the function keys F2 and F3 change to BUMP_UP and BUMP_DN respectively. To bump in x, y or z directions move the cursor to the appropriate field and press F2, BUMP_UP, to bump in the positive direction and F3, BUMP_DN, to bump in the negative direction.
Limit	The LIMIT column allows you to set the absolute maximum allowable offsets in each of the directions.

Procedure 10-10 Changing the Number of Schedules

Steps

1. Perform a Controlled start.
   a. **If the controller is on**, press FCTN, select CYCLE POWER, select YES, and press ENTER. **If the controller is off**, turn on the power disconnect circuit breaker.
   b. When the BMON screen is displayed on the teach pendant, press and hold the PREV and NEXT keys. You will see a screen similar to the following.

```
-----------CONFIGURATION MENU-----------
   1 Hot start
   2 Cold start
   3 Controlled start
   4 Maintenance
Select >
```

   c. After the configuration menu appears on the screen, release the PREV and NEXT keys.
   d. Select 3, Controlled start.
2. Press MENUS.
3. Select Variables.

   To move quickly through the information, press and hold down the SHIFT key and press the down or up arrow keys.
4. Move the cursor to $BB_TOT_SCH.
5. Type the number of schedules you want and press ENTER.
6. Press FCTN.
7. Select CYCLE POWER, select YES, and press ENTER.

After the cold start has completed, the controller is ready to use. The number of Bump Box schedules available will match the number of schedules you defined in this procedure.

Procedure 10-11 Setting Up a Bump Schedule

Steps

1. Press MENUS.
2. Press F1, [TYPE].
3. Select BumpBox. You will see a screen similar to the following. If you see the detailed screen, press PREV to see the list of Bump Box schedules.

```
TEST
UTILITIES BumpBox
   Program  Grp  ID     Lines   Status
1  EX_CIRC    1 SHAP01     3-4   BUMPED
2  EX_RECT    1 SHAP01     3-4   BUMPED
3  ******     0 ******     0-0   ******
4  ******     0 ******     0-0   ******
5  ******     0 ******     0-0   ******
6  ******     0 ******     0-0   ******
7  ******     0 ******     0-0   ******
8  ******     0 ******     0-0   ******
9  ******     0 ******     0-0   ******
```

4. To view the details of a schedule, move the cursor to the schedule and press F2, DETAIL.
5. If the current program is not in the list, move the cursor to an unused schedule (*******) and press F2, DETAIL. The current program name will be entered automatically.
6. To clear a schedule, move the cursor to the schedule and press F3, CLEAR. If the schedule has already been bumped you will be given a choice to restore the original positions in the teach pendant program.
7. To reset the offsets applied in a schedule, press F4, RESET. You will be given a choice to restore the original positions in the teach pendant program.
8. To copy one schedule to the other, move the cursor to the schedule you want to copy from and press F6, COPY. Type the schedule number to which you want to copy the schedule information. If the schedule you want to copy to has already been bumped, you will be given a choice to restore the original positions in the teach pendant program.
9. To clear all the schedules, press F8, CLR_ALL.
10. To reset the offsets applied in all the schedules, press F9, RST_ALL.
11. To see the details of the schedule, press F2, DETAIL. You will see a screen similar to the following.

```
TEST
UTILITIES BumpBox
   Bump schedule:  2      STATUS:  BUMPED
    1 Program name:          EX_RECT
    2 Group number:          1
    3 Cut ID:                SHAP01[3-4]
    4 Lines:                    3-4
    5 Offsets frame:         Shape frame
    6 Offset increment:      5.0 mm
   OFFSET         APPLIED      LIMIT(+/-)
    7 X offset:     0.0         25.0 mm
    8 Y offset:     0.0         25.0 mm
    9 Z offset:     0.0         25.0 mm
```

12. To select a different teach pendant program for this schedule, move the cursor to the Program name item and press F4, [CHOICE].

13. If a controller has multiple robot groups defined, each robot can be bumped independently by a unique bump schedule. To select a different robot to bump, move the cursor to Group number and type the robot group number.

14. To select a different Cut ID move the cursor to the Cut ID item and press F4, [CHOICE].

15. If you do not want to use the default range of lines move the cursor to the Lines item and enter the starting and ending line numbers.

16. To select the Offsets frame press F4, [CHOICE].

17. To set the offset limits, move the cursor to the LIMIT column and enter the desired offsets.

18. Press F2, SCHEDULE to display the details of another schedule.

19. To clear the schedule, press F1, CLEAR. If the schedule has already been bumped you will be given a choice to restore the original positions in the TP program.

20. To reset the offsets applied, press F8, RESET. You will be given a choice to restore the original positions in the teach pendant program.

21. After setting all the items, move the cursor to the APPLIED column to apply the offsets. Press F2, BUMP_UP, to apply the offsets in the positive direction and press F3, BUMP_DN, to apply the offsets in the negative direction. The applied column is updated as the schedule is being bumped. The status of the schedule changes to BUMPED.

22. If you try to change the Program name, Cut ID, Lines, or Offsets frame after the schedule has been bumped, you will be given a choice to restore the original taught positions.

Note If the positions in the teach pendant program are touched up after they have been bumped, Bump Box cannot restore the original positions.

10.7.3 Setting Up Bump Box I/O

In order to apply offsets using the digital inputs, the Bump Box I/O must be set properly. You must set up one group input and several digital inputs and outputs. The group input specifies the Bump Box schedule number. Six consecutive digital inputs are used to apply offsets in positive and negative (BUMP_UP and BUMP_DN) x, y and z directions. Two digital outputs are used. One acknowledges the offsets and another indicates a limit error.

The group input and digital inputs and outputs must be properly configured before you can use them. Refer to Chapter 4 *INPUT/OUTPUT (I/O) SETUP* for information on configuring digital I/O. The number of points in the group input determines the number of schedules that can be bumped using I/O. For example, if there are three points in the group input only schedules 1 through 7 can be bumped. If there are four points then schedules 1 through 15 can be bumped. If there are n schedules, schedules 1 through ($2^n - 1$) can be bumped.

When Bump Box I/O is enabled, Bump Box monitors the six digital inputs selected as Bump inputs. When any one of the six inputs turn from OFF to ON, Bump Box applies the offsets in the direction corresponding to the input according to the schedule given by the group input value. If the requested bump exceeds the limits specified in the schedule, the limit error digital output is set to ON for the duration specified by the pulse width. After successfully applying the offsets, the acknowledge offset digital output is set to ON for the duration specified by the pulse width.

If Bump Box I/O is disabled, then Bump Box stops monitoring the digital inputs and changing the state of the digital inputs has no effect. When Bump Box I/O is enabled, offsets are applied using digital inputs and cannot be applied from the teach pendant.

Use Procedure 10-12 for setting up the Bump Box I/O. Refer to Table 10–10 for descriptions of Bump Box I/O items.

Table 10–10. Bump Box I/O Items

ITEM	DESCRIPTION
Bump Schedule	This item allows you to change the group input that will be used for selecting a bump schedule.
Bump Inputs	This item allows you to change the first of the six consecutive digital inputs that will be used for applying offsets in + x, -x, +y, -y, +z, and -z directions respectively.
Acknowledge offset output	This item allows you to change the digital output that will acknowledge the offsets.
Acknowledge pulse width	This item displays the Acknowledge pulse width in milliseconds.
Limit error output	This item allows you to change the digital output that indicates Limit errors.
Limit error pulse width	This item displays the Limit error pulse width in milliseconds.

Procedure 10-12 Setting Up Bump Box I/O

Conditions

- You have configured the group and digital I/O you want to use to apply offsets. Refer to Section 4.2 for information on setting up group and digital I/O.

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Bump Box. You will see a screen similar to the following.

```
SETUP BumpBox I/O
Bump Box I/O:  DISABLED
   1 Bump schedule:           GI[  0]
   2 Bump Inputs:             DI[  0 - 0 ]
   3 Acknowledge offset output:        DO[  0]
   4 Acknowledge pulse width:          100 ms
   5 Limit error output:               DO[  0]
   6 Limit error pulse width:          100 ms
```

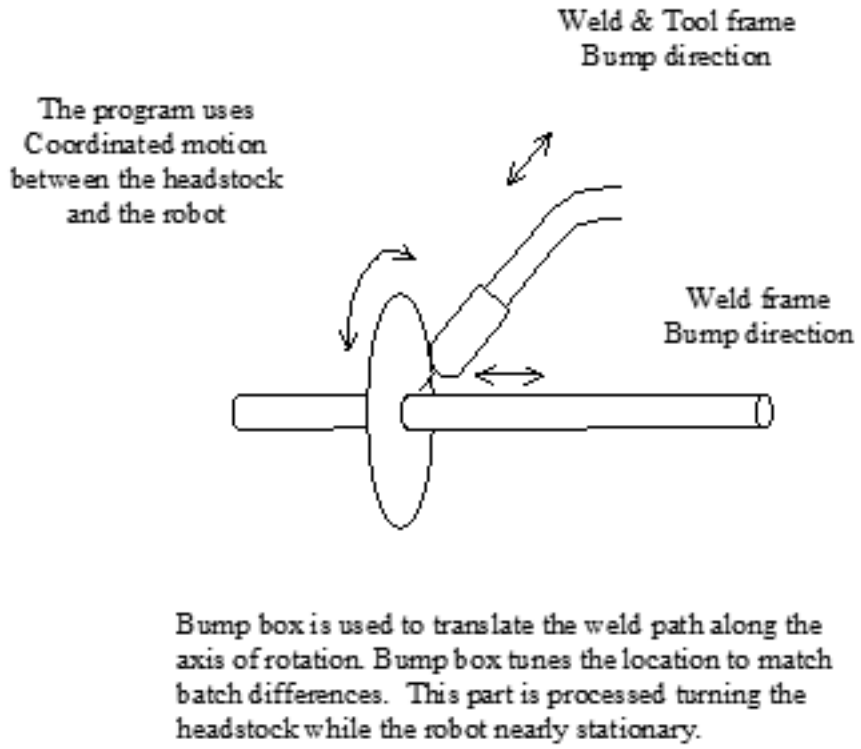
You can enable or disable Bump Box operation by setting Bump Box I/O

5. To enable applying offsets using digital I/O, press F4, ENABLED.
6. To disable applying offsets using digital I/O, press F5, DISABLED
7. Move the cursor to the Bump Schedule item and type the number of the group input. The value of the group input must be the Bump box schedule number.
8. Select the first of the six consecutive digital inputs required for Bump Box operation. The last digital input is automatically updated when you change the first digital input.
9. Type valid numbers for the acknowledge offset and limit error digital outputs and the corresponding pulse widths.
10. Move the cursor to Acknowledge offset output and type number of the digital output you have set up for it.
11. Move the cursor to Limit error output and type the number of the digital output you have set up for it.
12. Move the cursor to Acknowledge pulse width and type the value you want for it.
13. Move the cursor to Limit error pulse width and type the value you want for it.

10.7.4 BumpBox with Coordinated Motion - Overview

Bump Box will work with programs using Coordinated Motion instructions within the weld path. You can "bump" coordinated motion instructions in multi-group programs. The bump has the expected results for WELD, WELD & TOOL, and WORLD/UFRAME bump frames, but it is done with respect to the coordinated motion frame or the coordinated motion leader frame. This can be used to "bump" part paths in the coordinated leader frame defined on the part, and also to offset the weld position along the axis of rotation, as shown in the figure below:

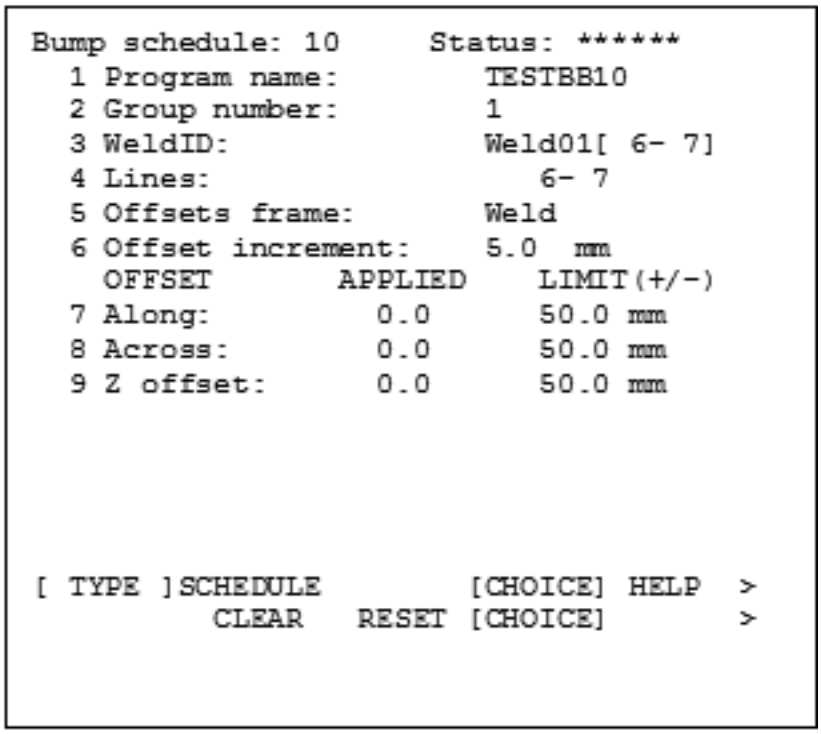
Figure 10–10. BumpBox with coordinated Motion



The information needed to apply CD motion offsets (bump) is contained in a Bump Box schedule, which must be set up before using Bump Box. In addition, coordinated motion data for the leader and follower groups must also be setup. The "Setup/BumpBox" menu and "Utility /BumpBox" menu have not been changed. The interface to apply offsets has not changed: offsets can be applied from the teach pendant or by setting and enabling digital inputs in the Setup/BumpBox menu. The offsets can be applied only in three directions (labels change based on Bump Frame selected in a given bump schedule. "Reset" can be used to remove the cumulative BumpBox offsets applied to coordinated motions; you will be prompted to reset the positions to their original values. "Clear"

can be used to reset the BumpBox schedule settings to their default values; you will be prompted to reset the positions to their original values.

Figure 10–11. BumpBox Schedule


Bump schedule: 10 Status: ******
1 Program name: TESTBB10
2 Group number: 1
3 WeldID: Weld01[6- 7]
4 Lines: 6- 7
5 Offsets frame: Weld
6 Offset increment: 5.0 mm
OFFSET APPLIED LIMIT(+/-)
7 Along: 0.0 50.0 mm
8 Across: 0.0 50.0 mm
9 Z offset: 0.0 50.0 mm
[TYPE]SCHEDULE [CHOICE] HELP >
CLEAR RESET [CHOICE] >

- Multiple leader and follower group devices may be contained in the same program, but offsets will be generated with respect to only one leader. The motion instruction must use COORD [ldr] or COORD [R[#]] to indicate which of the leader groups is used with the follower motion group defined in the "Motion group" setting in the Bump Schedule. The CD motion of the additional leader group(s) is not guaranteed to perform to the user's expectation. Motion of the additional follower group(s) is not guaranteed to perform to the user's expectation. The user must validate the operation of BumpBox in these complicated programs.
- Changing the leader group within the "bump lines" will cause an alarm to be posted and the operation will abort prior to changing any position data.
- The leader and follower frames do not have to be aligned for CD bump calculations to be performed, but the user may have to train a CD leader frame to align a bump direction to a part manipulated by the leader group. Uframe, Utool, and leader frame are supported on follower robots.
- Robot-to-Robot coordinated motions are not supported in the "bump lines" of the selected schedule; an alarm will be posted and the operation will abort prior to changing any position data.
- Non-CD Shape motion lines can be bumped. Shape generation programs using CD motion in the bump lines are not supported; an alarm will be posted and the operation will abort prior to changing any position data.
- The first line of the "Bump lines" cannot be the VIA point of a circular motion (C P[ ].) An alarm will be posted and the operation will abort prior to changing any position data.

Leader Group Devices

Two Types of positioner devices are used for leader groups in CD pairs:

- Most Basic positioner leader groups are single axis devices, either a headstock or a tabletop positioner. They can also have multiple axes if the axes are either parallel or perpendicular to each other.
- General positioner type can be used interchangeable with basic positioner type and it is typically used in multi-axis leader devices. General positioner type must be used in cases where two or more axes are not perpendicular or parallel to each other.

Coordinated motion calibration assigns the leader group "world frame" which is analogous to the robot world frame.

Multi-group or Multi-Arm Systems

- Systems including one positioner leader with one robot follower. (The leader device is generally a single axis headstock, or a tabletop positioner, but positioner leaders can be composed of up to 3 axes and also include linear or rotary extended axes.)
- Systems including one positioner leader and multiple follower robots simultaneously coordinate with the leader.
- Two positioner leaders and 2 robot followers where each robot is permanently paired to one of one of the leader groups; each pair of robot and headstock executes independently.

- Systems including multiple positioner leaders with multiple robot followers. One headstock is being loaded while the other is being welded; they rotate when the loading and welding complete. For programs including multiple follower robots, only one follower will be exactly coordinated during CD motion. The other robot will run slower unless the follower paths move the same distance on a point-by-point basis.

Bump Offsets Supported Paths

CD ‘bump” operations can to be applied to coordinated motion paths that are:

- Linear or circular paths with positions taught in a user frame of the follower
- Concentric circular paths to an axis of rotation of headstock or tabletop leaders.
- Linear or circular paths on, or parallel to, the faceplate or table-top
- Linear or circular paths contained in a CD leader frame plane with respect to the leader world frame.
- CD leader frames for the leader frame can be used to align bump directions to “bump” axis inputs.

Position Types Supported and Motion Modifier Qualifications

The following interpolation types and motion modifiers can are supported by CD BumpBox operations in coordinated motion programs:

- CD, CD[LDR], and CD[R[n]] instructions in the same bump path; the reference to leader must be consistent for all bump lines.

Note Coordinated leader device cannot be changed in the path points in the bump schedule “lines” parameter.

Supported Motion Types are

- “C P[ ] .. CD” and “L P[ ] .. CD” motions are supported in the bump “lines.”
- “A P[ ].. CD” motion is supported; resume to path feature and restart location are different than for C P[ ] motions.
- J P[ ] motions will be supported within a CD Bump “Lines”

  Note CD does not apply to J P[] motion.

- “PR[ ].. CD” is not supported. This position type will be bypassed in the Bump operation, and an alarm will be posted. The bump program will not be aborted, and all other normal positions will be bumped.
- The results of adding additional “frame” and “tool” offsets will not be guaranteed to have the expected results when “bumped” as non-CD motion positions using the same motion offset modifiers.
- Unsupported motion options for CD are unsupported motion options for CD BumpBox operations.

- Scratch motion and overlap distance are not guaranteed to be on the motion path.

10.7.5 Bump Frames for Normal and Coordinated Motion

The following table shows the Coordinated Motion bump frame types. The first three types are implemented for normal motion programs, the additional three are supported for Coordinated motion. These are analogous to the normal motion bump frame types.

Table 10–11. Bump Frames

Frame Type in Bump Schedule	Internal Name	Bump directions defined by Offset Frame... (“Frame direction” is displayed in schedule)
WELD(Normal motion)	WELD	"Along” is the direction of the robot path initial 2 points. This determines the X direction with respect to the current Uframe (if non-zero), or robot world frame. “Across” is parallel to the current Uframe (if non-zero), or robot world frame Y-axis. “Z Offset” is parallel to the current Uframe (if non-zero), or robot world frame Z-axis.
(Coordinated Motion)	(CD_Weld)	"Along” is the direction of the robot path, defined between the initial 2 points in the Schedule Bump lines. This determines the X direction (with respect to the current CD leader frame if non-zero), or leader world frame. “Across” is parallel to the current CD leader frame (if non-zero), or the leader world frame Y-axis. “Z Offset” is parallel to the current Uframe (if non-zero), or robot world frame Z-axis.

Table 10–11. Bump Frames (Cont'd)

Frame Type in Bump Schedule	Internal Name	Bump directions defined by Offset Frame... ("Frame direction" is displayed in schedule)
WELD & TOOL (Normal motion)	WELD & TOOL	"Along" is the direction of the robot path, defined between the initial 2 points in the Schedule Bump lines. This determines the offset "X" direction with respect to the current Uframe (if non-zero), or robot world frame. "Across" is perpendicular to the to the plane defined by the X direction and the Utool Z axis."Stickout" is defined as the Utool Z axis with respect to the current Uframe (if non-zero), or robot world frame.
(Coordinated motion)	(CD_WELD _TOOL)	"Along" is the direction of the robot path, defined between the initial 2 points in the Schedule Bump lines. This determines the offset "X" direction with respect to the current CD leader frame (if non-zero), or leader world frame. "Across" is perpendicular to the to the plane defined by the X direction and the Utool Z axis."Stickout" is defined as the Utool Z axis with respect to the current CD leader frame (if non-zero), or leader world frame.
WORLD/UF(Normal motion)	WORLD/UF	"X Offset", "Y Offset", and "Z Offset" are parallel to X, Y, and Z direction with respect to the current Uframe (if non-zero), or robot world frame.
(Coordinated motion)	(CD_LDRFR_UF)	"X Offset", "Y Offset", and "Z Offset" are parallel to X, Y, and Z direction with respect to the current CD leader frame (if non-zero), or leader world frame.

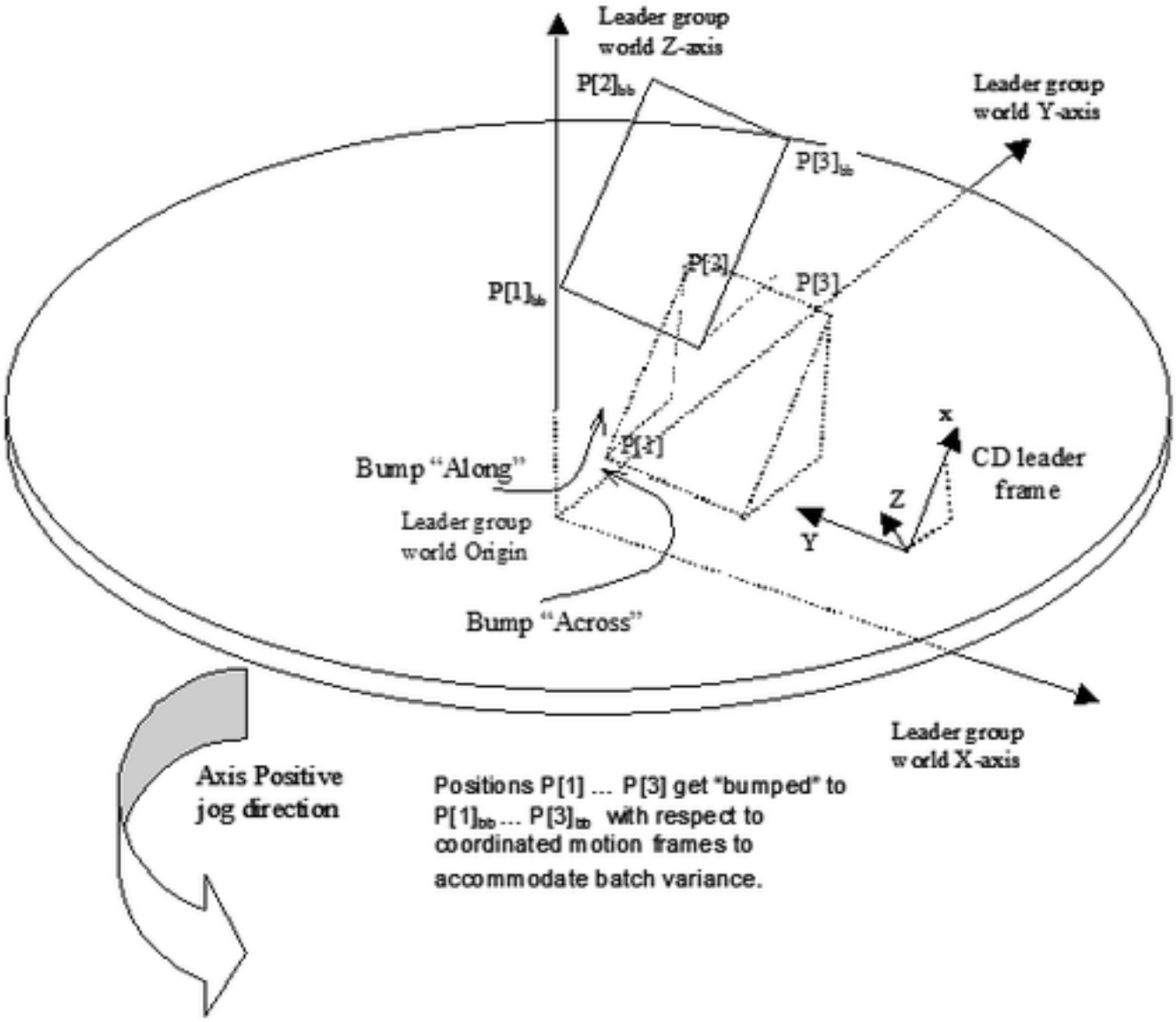
The "CD" bump frames are internal to the software and not displayed to the user in the bump schedules. In the coordinated motion bump cases, the alignment of the reference frames is not always consistent with the Bump direction "Along/X" and "Across/Y", or " on the part. This can be corrected by using the a CD Leader frame aligned so that the bump direction corresponds to the "bump offset".

Leader Group Frames

There are two frames that determine how coordinated motion “bumps” executed, These are the leader world frame and the CD leader frame. The leader world frame defines the origin and axis alignment of the leader motion group. A CD leader frame is analogous to a Jog frame with respect to the leader world frame. The CD leader frame moves when the leader group moves. These are transparent to most uses of coordinated motion, but they play an important and function in setting up the Bump directions relative to the part path.

The following figure of a tabletop positioner shows the leader world frame and also a CD leader frame. The CD leader frame defines the plane in which the path P[1] .. P[3] executes in this example. When this frame is defined, the bump offsets are calculated with respect to the X, Y, and Z-axes of this frame. In addition, the user can select to jog in this frame by pressing the COORD hardkey on the teach pendant when a follower motion group is selected. The relationship of the frames is covered in more detail below. Please refer to the “Coordinated Motion Setup and Operations Manual” section “Calibration” regarding details of that operation.

Figure 10–12. BBox with Table



Leader Group World Frame (Unknown Kinematics Type)

When a positioner group is added to the system, the installer is prompted to select a kinematics type. "Unknown kinematics " indicates that a user will define the calibration by point teaching methods in the Coordinated Motion Setup menu. "Known kinematics " indicates that the user knows the exact alignment of the positioner axes and also the exact location of the positioner with respect to the follower robot origin. This type requires information to be added immediately upon installation of axes in the positioner group.

When an axis is added to the “Unknown kinematics “group, the axis assignment and axis direction are entered. For multiple axes, the Cartesian offset of the subsequent axes is entered. These assignments are overridden when Coordinated Motion calibration is performed. Calibration consists of recording thee points for each rotary axis of the leader group in the Coordinated Motion calibration screen

The CD motion calibration normally assigns the leader world Z-axis perpendicular to the tabletop or headstock faceplate axis of rotation. The positive direction of the axis is determined by the right hand rule for current based on the direction the axis turns when its positive rotation key is pressed. The X and Z are defined normal to the Z-axis.

For basic and general positioner leaders with multiple axes, the calibration software determines the offset between axes and makes the direction assignment using the right hand rule for. The last axis, the tabletop or faceplate, is assigned as the Z-axis. The axis immediately preceding the last axis is defined as the X-axis, and Y is applied to the next preceding axis when three axes are present. For basic positioners, the X axis and Y-axis are assigned normal to the Z-axis assignment and parallel to the other axes “axis of rotation”. In the case of general positioner the DH parameters of axes is determined, and a world frame is assigned. If multiple axes have parallel axis of rotation, the same direction is assigned to all such axes.

The calibration data ($CD_PAIR[n].$CD_XF) defines the position of the leader group world origin with respect to the follower robot origin. Seldom is the origin located in the faceplate plane of the positioner device, or at the intersection point of the axes of rotation in the case of multiple axes. Please refer to the “Coordinated Motion Setup and Operations Manual” section “Calibration” regarding details of that operation.

Leader Group World Frame (Known Kinematics Type)

For the “Known Kinematics” case, the world frame of the leader is determined before the calibration data is entered. When “Known kinematics “ is selected at the time a positioner group in created, the axis assignment of +/-X, +/-Y, or +/-Z to each axis of the basic or general positioner is entered.

In addition, the Cartesian offsets between axes of multi-axis positioners are manually entered. The Coord Motion calibration consists of manually installing values for X, Y, Z, W, P, and R, based on a known alignment of the origin of the leader device with respect to the origin of the follower robot. Please refer to the “Coordinated Motion Setup and Operations Manual” section “Calibration” regarding details of that operation.

Leader Group “CD Leader Frame”

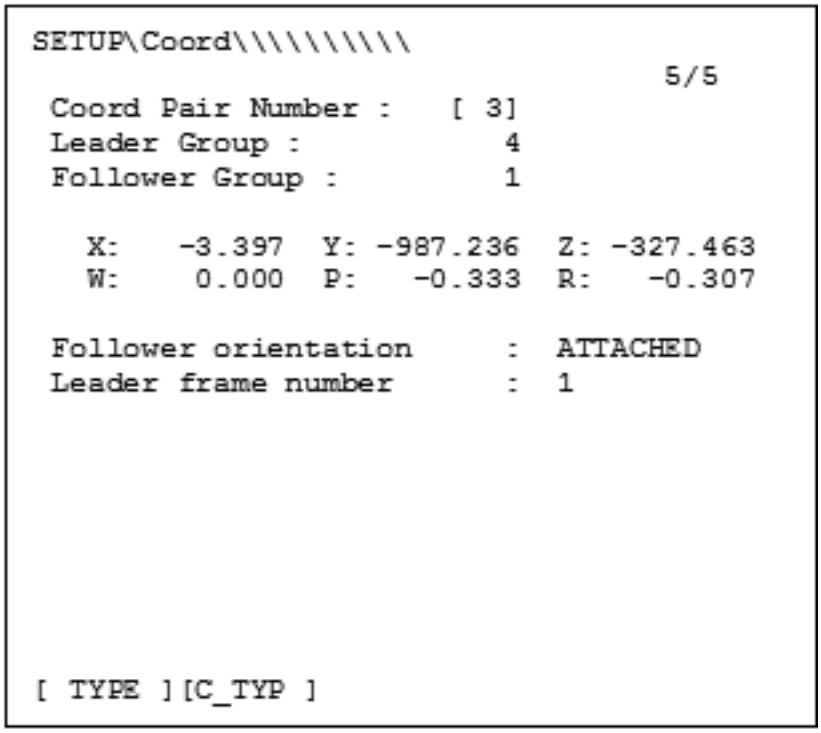
The CD leader frame is a component of the Coordinated Motion software option. It is a jogging frame for the follower robot that is taught with respect to the leader world frame. The CD leader frame is a dynamic frame, so LDR jogging the direction of the X, Y, and Z axes of the CD leader frame will appear to change as the leader axis (axes) position change.

The CD leader frame is setup in the Coordinated Motion setup screen. There are six CD leader frames that can be taught for each cd_pair that is defined, and they are taught by using a three-point teaching method:

Procedure 10-13 CD Leader Frame Setup

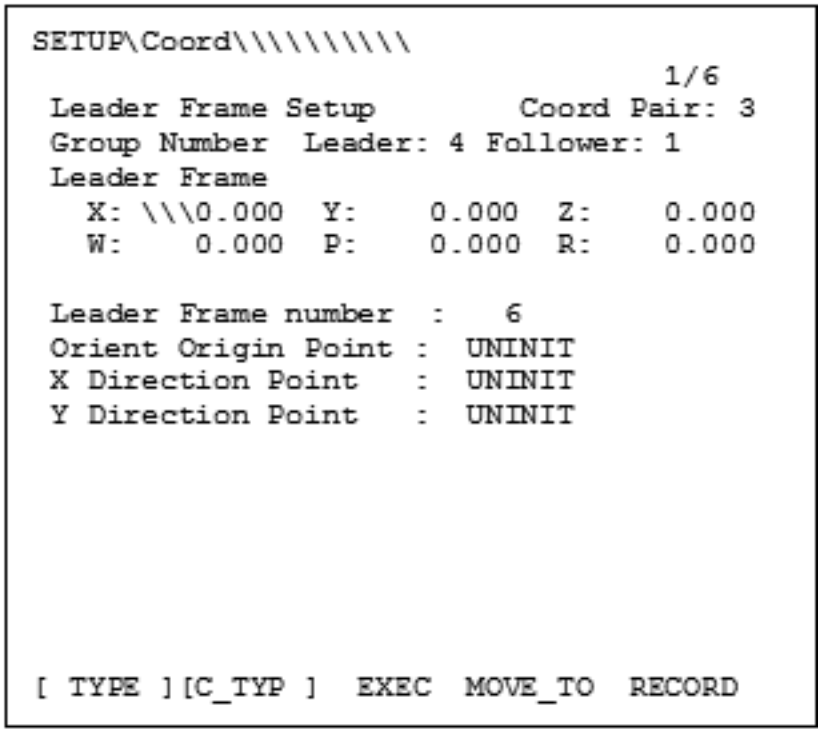
1. Go to the coordinated Motion Setup Screen.

Figure 10–13. Coordinated motion setup screen



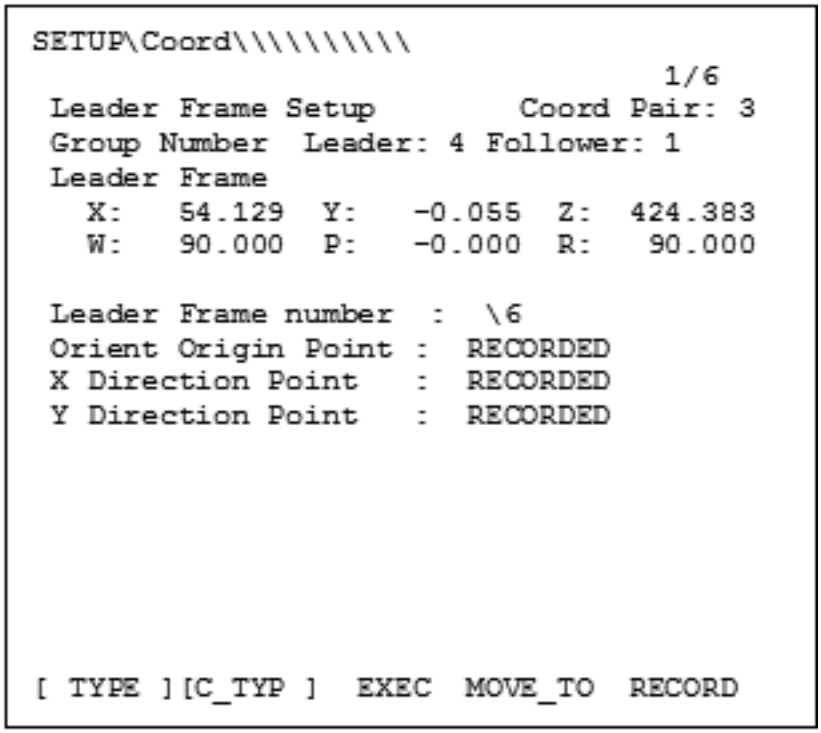
2. Press [C_TYP], then scroll to “Leader Frame” in popup menu.

Figure 10–14. Leader Frame PopUp menu



3. The frame is determined by the three-point method.

Figure 10–15. Leader Frame Setup - 3 point Method



4. After the three frame points have been recorded, press EXEC to calculate the frame. Cycle power when the frame values are shown, or continue on to training other CD leader frames and then cycle power.

You select to use the CD leader frame for jogging pressing the COORD key on the teach pendant until "LDR #" is shown as the jogging frame for the robot. To switch between CD leader frames, you need to change the CD leader frame index in the Coordinated Motion setup screen (see figure 1.3.) These frames may be aligned to the workpiece, the leader world frame, or some arbitrary frame that is suitable for jogging the robot. When leader frame is zero, and the leader device is at it zero position, LDR jogging shows the leader world frame. Please refer to the "Coordinated Motion Setup and Operations Manual" section "Calibration" regarding details of teaching the CD leader frame.

CD BumpBox uses the CD leader frame and leader world frame for bump calculations. When the current CD leader frame value is zero, the leader world frame is used for the CD World/Uframe bump frame. When the CD leader frame has a non-zero value, the CD leader frame is used for the "World/UF" reference. The CD_LDRFR_UF bump frame uses, the X, Y, and Z values. The CD_WELD Bump frame uses the Y and Z components.

Using "CD Leader Frame" to Align the Bump Frames

When the current CD leader frame value is zero, the leader world frame is used for the X, Y, and Z of the CD World/Uframe bump frame and uses the Y, Z for CD_WELD Bump frame.

If the bump frames do not have the proper alignment to the workpiece, “X, Y, Z” bump (CD_LDRFR_UF) are not aligned to the part, or the “Across” (CD_WELD), then a CD leader frame that defines these directions can be taught to correct the alignment. This is more likely to be necessary in multiple axis positioner leaders, or when the plane of motion for the robot process path is not parallel the tabletop or faceplate axis.

Note When a non-zero CD leader frame is used to correct for misalignment of bump directions, this frame must always be set to be the default CD leader frame when a bump operation is performed, otherwise the bump calculation will be incorrect. When this bump operation is completed, the CD leader frame must be set to a zero value CD leader frame so that it is not applied to other bump operations.

Controlling the Default CD Leader Frame

There are two ways to control the Default CD leader Frame

1. To switch between CD leader frames, you need to change the CD leader frame index in the Coordinated Motion setup screen.
2. The CD leader frame can be controlled in the program that is using BumpBox. By inserting a “Parameter” instruction into the program, you can force a default CD leader frame to be assigned. In the program below, CD leader frame 6 of cd_pair[3] is used (seeFigure 10–14 above) when line 3 is executed. CD leader frame 1 of cd_pair[3], a zero value, removes the CD leader frame line 8 is executed. Line 7 is the end of the bump lines.

Figure 10–16. CD Leader Frame Example

```
 1:  UFRAME_NUM=1 ;
 2:  UTOOL_NUM=1 ;
 3:  $CD_PAIR[3].$LDR_FRM_NUM=6 ;
 4:   ;
 5:L P[1] 500mm/sec FINE    ;
 6:L P[2] 1.5sec CNT100 COORD
  :  Arc Start E1[1]    ;
 7:C P[3]
  :  P[4] 60.0inch/min FINE COORD
  :  Arc End E1[2]    ;
 8:  $CD_PAIR[3].$LDR_FRM_NUM=1 ;
 9:L P[5] 500mm/sec FINE    ;
10:L P[6] 500mm/sec FINE    ;
```

10.8 CENTER FINDER

10.8.1 Overview

Center Finder is a motion option that allows you to find the center of a two-dimensional symmetric shape such as a circle, hexagon, or triangle. The found center can be used as a reference point. You might use it for position recording incremental positions from the center of a chuck, for example.

Center Finder uses the Torque Observer Sensor (TOS) or I/O method to touch points on the shape. The center is computed based on the contact positions. The same process is repeated until the current center position and the newly computed one are within a pre-defined tolerance. Center Finder then moves the TCP to the center position and records the position to a position register.

Note Because the robot gripper is very stiff, robot motor torque might be used to determine locations of the chuck.

Center Finder can never be used during production operation. It can only be used during initial teaching and setup.

Center Finder does not support any robots that have fewer than six axes, and Center Finder does not support a multi arm configuration.

Using Center Finder

After you set up Center Finder, it performs the following steps to find the center:

1. Computes the search directions.
2. Moves the robot in the search directions to detect the edge of the opening.
3. Computes the center of the opening, and moves the TCP to the center.
4. Confirms the center by repeating Step 2 until the difference between the currently computed center and the previous center is within the pre-defined tolerance.
5. Updates a position register (defined during setup) to store the center position. The function leaves the robot in the center position. This enables you to verify the acceptability of the position.

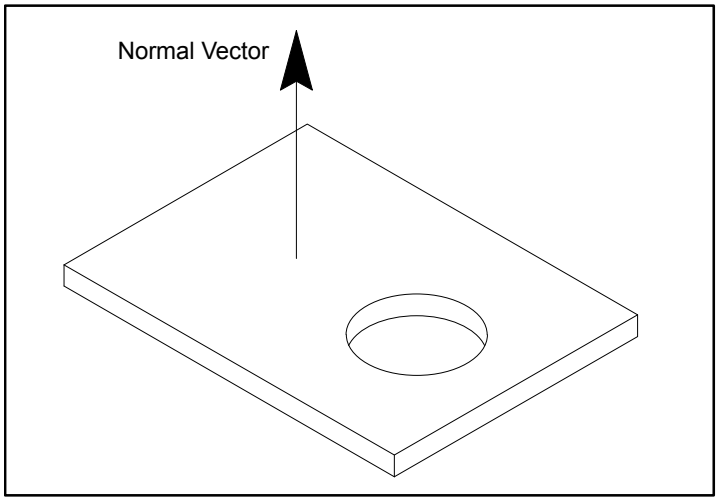
   Note You can then adjust the chuck jaws and redo the search if desired.

10.8.2 Using Center Finder

You must install the Center Finder software option before you can use it. Refer to the *Software Installation Manual* for more information about installing software options.

1. If you are using the TOOL frame, make sure the frame is set up correctly. If you are using the USER frame, make sure the frame is set up correctly.
2. Select the shape's normal (perpendicular) axis. This axis has to align to one of the motion frame's major axes. See Figure 10–17 .

Figure 10–17. Shape with a Perpendicular Axis



3. Select the search type.
4. Set up individual motion parameters if required.
5. After you have set it up, you can run the Center Finder program to find the center.

Note In a multiple-group system, the robot is the first group.

Table 10–12 and Table 10–13 list and describe the items you must set up. Use Procedure 10-14 to set up Center Finder.

Table 10–12. Center Finder Menu Items

ITEM	DESCRIPTION
Position Register	This item allows you to select which position register you want to use. When you select a different position register, the x, y, z, w, p, and r positions will display the content of the selected position register.

Table 10–12. Center Finder Menu Items (Cont'd)

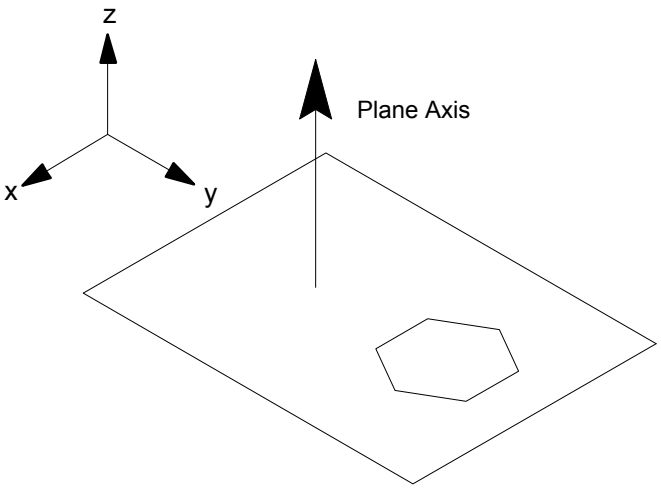
ITEM	DESCRIPTION
Motion Frame	This item allows you to select which motion frame you want to use for the contact search motions. There are six selections: • World: Robot world frame • Tool: Robot current tool frame • user: Current $MNUFRAME • World + Z: Robot world frame + Z direction offset * • Tool + Z: Robot $MNUTOOL frame + Z direction offset * • User + Z: Current $MNUFRAME + Z direction offset *
Plane Axis	This item allows you to select which axis will function as the Plane Axis. The Plane Axis is the axis normal to the plan that contains the shape. There are six selections: • X Axis • Y Axis • Z Axis • —X Axis • —Y Axis • —Z Axis 

Table 10–12. Center Finder Menu Items (Cont'd)

ITEM	DESCRIPTION
Search Type	This item allows you to set the type of search that Center Finder will perform. There are seven selections: • Collet Chuck: chuck without fingers • Three Finger Chuck: chuck with three fingers • Four Finger Chuck: chuck with four fingers • Shape: Any symmetric shape, such as circle, hexagon, oval, or rectangle • X: Center of two points in plane X axis ** • Y: Center of two points in plane Y axis ** • Z: Center of two points in plane Z axis **
Motion Setup	This item displays whether Motion Setup is complete or incomplete for Center Finder. You must set up motion for your environment the first time you use Center Finder.
Orientation Setup	This item displays whether Orientation Setup is complete or incomplete for Center Finder. Orientation Setup is optional. It allows you to adjust the robot so the tool is perpendicular to the plane that contains the geometry.

* When you select one of these options, you must run Z_MOVE to check for the z position before you can run the center finder program. Otherwise these options will not work. Refer to Procedure 10-14 for more information.

** Only one dimension is searched to find the center of two points.

Table 10–13. Center Finder Motion SETUP Menu

ITEM	DESCRIPTION
Points on Plane	This item specifies the number of searches for the shape. For x axis, y axis, and z axis search types, this setting is ignored.
Motion Speed	This item specifies the normal motion speed when the robot is looking for the contact.
Search Speed	This item specifies the contact searching speed when the robot is moving back from a contact position to a search position.
Search Distance	This item specifies the maximum amount of distance the contact search motion will travel. If the robot does not detect a contact within the search distance, an error will be posted.

Table 10–13. Center Finder Motion SETUP Menu (Cont'd)

ITEM	DESCRIPTION
Learn Distance Units: mm Range: 0.5 – 10.0 Default: 5.0	For"TOS All Axes" method only, there is a short motion before each search to identify dynamic torque values. Learn Distance is the distance in millimeters of this short motion, the direction is determined at the time of the motion.
Learn Speed Units: mm/sec Range: 1.0 – 75.0 Default: 10.0	For"TOS All Axes" method only, this is the speed of the learning motion before each search.
Convergence Error	This item specifies the iteration convergence criteria. If the current iteration is within the specified error, the center position is found.
Do Z Check	This item allows you to set the Do Z Check flag. If the flag is set to TRUE, you can run Z direction for the chuck. If the flag is set to FALSE, you cannot execute Z_MOVE.
Orientation Angle	This item specifies the maximum allowable orientation change for setting up the orientation of the tool to be perpendicular to the chuck surface.
Sensitivity Tolerance	This item allows you to set the sensitivity tolerance for the Torque Observer Sensor (TOS). The lower the number is, the more sensitive the TOS will be. **Warning** **Be careful when changing this number. Changing the sensitivity tolerance might cause a false detection. This could injure personnel or damage equipment.**
Torque Threshold	This item allows you to set the percentage of torque the robot will put out before it detects a contact.

Table 10–13. Center Finder Motion SETUP Menu (Cont'd)

ITEM	DESCRIPTION
Sensor Type Values: TOS Wrist , I/O, TOS & I/O, or TOS All Axes Default: TOS All Axes	This item indicates the kind of sensor used with TCP calibration. The three modes are: • **TOS Wrist** (Torque Observer Sensor on Wrist) - The contact torque will be monitored, and only the observed torque will be used to determine the contact position. When TOS Wrist is selected, no I/O connection is required for contact detection. • **I/O** - The specific I/O will be monitored and only the state of this input will be used to determine the contact position. Center Finder uses either DI or RDI. • **TOS & I/O** - Both torque and I/O will be monitored, and the first one to be detected will be used to determine the contact position. If the Sensor Type is set to TOS, you do not need to set up I/O. • **TOS All Axes** (Torque Observer Sensor, All Axes) - The contact torque will be monitored on all robot axes, and only the observed torque will be used to determine the contact position. This is the default value. When TOS All Axes is sected, no I/O connection is required for contact detection.
Detection Port Type Values: RDI, DI, WDI, WSI Default: DI	This item allows you to select RDI, DI, WDI, or WSI (wire stick-out input port) for the contact detection port.
Detection Port Number Default: 0	This item allows you to select the I/O port number for the contact detection port.
Sensor Enable Port Type Values: RDO, DOUT, WDO, WSO Default: DOUT	This item allows you to set the sensor enable port type to RDO, DOUT, WDO, or WSO This is optional. If your I/O detection port needs to be enable before it can do contact detection, you have to set up this port.
Sensor Enable Port Number Default: 0	This item allows you to set the digital output port number for the circuit enable port.

Procedure 10-14 Using Center Finder

Conditions

- Y ou have installed the Center Finder option. Refer to the *Software Installation Manual* .
- Tool frames have been set up accurately.

Caution

You should not move the extended axes while you set up or execute Center Finder. If the extended axes do move, Center Finder might incorrectly report the center position.

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Center Finder. You will see a screen similar to the following.

```
Center Finder Menu
Position Register [1]:
  X:  0.0      Y:  0.0     Z:  0.0
  W:  0.0      P:  0.0     R:  0.0
Motion Frame:        World
Plane Axis:          X Axis
Search Type:         Shape
Motion Setup:        Complete
Orientation Setup:   Incomplete
```

5. Move the cursor to each menu item that you want to change and set it as desired.
6. To access the Center Finder motion setup menu, select Motion Setup and press F3, DETAIL. You will see a screen similar to the following.

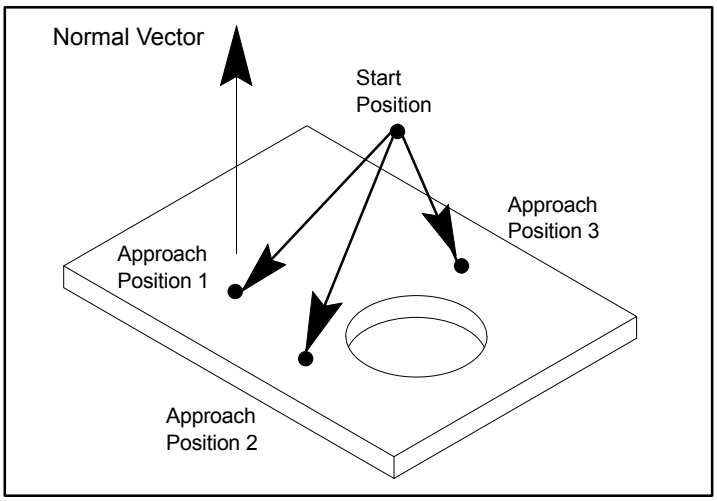
```
Motion Setup
Motion Setup
  Points on Plane:       12
  Motion Speed:          50.000  mm/sec
  Search speed:          10.000  mm/sec
  Search Distance:       50.000  [mm]
  Learn Speed:           15.000 mm/sec
  Learn Distance:         3.000  [mm]
  Convergence Error:     .200    [mm]
  Do Z check:            FALSE
  Orientation angle:     10.000  [degree]
Torque Setup
  Sensitivity:           10
  Torque Threshold:      100%
I/O Setup
  Sensor Type:I/O
  Detection Port:        RDI [  1]
  Sensor Enable:         RDO [  0]
```

7. Move the cursor to each menu item that you want to change and set it as desired.
8. Press PREV to display the Center Finder menu screen if you want to run Orientation Setup. You will see a screen similar to the following.

```
Center Finder Menu
Position Register [1]:
  X:  0.0      Y:  0.0      Z:  0.0
  W:  0.0      P:  0.0      R:  0.0
Motion Frame:        World
Plane Axis:          X Axis
Search Type:         Shape
Motion Setup:        Complete
Orientation Setup:   Incomplete
```

Note Orientation Setup is optional. It is used to align the robot's tool to the Normal vector of the shape. See Figure 10–18 .

Figure 10–18. Orientation Setup Approach Positions



9. Select Orientation Setup and press F3, DETAIL. You will see a screen similar to the following.

```
Orientation Setup
   Start Position:          UNINIT
   Approach Position 1:     UNINIT
   Approach Position 2:     UNINIT
   Approach Position 3:     UNINIT
```

10. Move the cursor to select Start Position.

11. Jog the robot to the position that you want to record as the Start Position and press F4, RECORD.

12. Select Approach Position 1. Jog the robot to the position that you want to record as Approach Position 1 and press F4, RECORD.

13. Repeat Step 12 to record Approach Position 2 and Approach Position 3. See Figure 10–18 .

14. To verify the positions you have recorded, select Start Position and press F5, MV_LIN. The robot will move from Start Position to Approach Position 1. To move the robot to each Approach Position, press F5, MV_LIN again.

15. When you are sure that you have recorded the correct approach positions and are ready to execute Center Finder, press F3, EXEC. The controller will calculate the center of the positions you recorded.

16. To apply the found tool orientation to the robot, press F2, APPLY. The robot will move and the tool should be aligned with the normal vector of the shape.

17. Press PREV to return to the Center Finder menu.

18. Move the robot tool so that it is in the shape.

19. Press F2, EXEC, to find the center of the shape.

If you encounter a problem, refer to Section 10.8.3 , "Troubleshooting."

10.8.3 Troubleshooting

Refer to Table 10–14 for information on Center Finder errors and solutions.

Table 10–14. Troubleshooting Information

ERROR	SOLUTION
Robot does not make contact with the part	• Increase the search distance in the Center Finder Motion Setup menu if the robot is moving along a correct search direction. • Re-teach the search start position so it is closer to the part.
Robot does not make solid contact with the part and returns to the search start position prematurely.	Display the Center Finder system setup sub-menu and change the Sensitivity Tolerance to a larger number.
Excessive vibration when robot is returning from the contact position back to the search start positions.	Display the Center Finder system setup sub-menu and reduce the motion speed.
Center Finder does not have enough information to determine the center and reports a "Cannot Converge" error.	Increase the number of points on the shape.

10.9 COLLISION GUARD

10.9.1 Overview

Collision Guard provides a highly sensitive method to detect that the robot has collided with an object and then stops the robot immediately. This helps to minimize the potential for damage to the end-of-arm tooling and robot. Collision Guard also helps to prevent damage during teaching. Additionally, you have the ability to change the sensitivity or even disable Collision Guard within a TP program to handle situations in which the disturbance is expected to vary widely from normal

operation. Collision Guard is in effect both during jogging motion and programmed motion. There are several ways to configure and adjust Collision Guard:

- The Collision Guard SETUP screen allows you to adjust the sensitivity of collision detection for programmed motion. Collision Guard automatically uses more sensitive limits for jogging motion. These limits cannot be adjusted.
- Within a teach pendant program, you can disable Collision Guard locally through the use of special teach pendant instructions, COL DETECT OFF and COL DETECT ON.
- Within a teach pendant program, you can adjust the sensitivity of Collision Guard locally using the COL GUARD ADJUST teach pendant instruction.

Note In order to decrease the force of collision, Collision Guard allows the robot axes to sag away from the collision for 200 milliseconds after detecting a collision. When this happens, vertical robot axes might fall slightly after detecting a collision, due to the effect of gravity.



Warning

If you have used the SHIFT + RESET key combination to clear an error condition and you continue to press the SHIFT key without releasing it, subsequent collision alarms will NOT be detected, collision detection alarms will NOT be displayed, and you could injure personnel or damage equipment.

10.9.2 Limitations

- Collision Guard cannot be permanently completely disabled.
- Collision Guard cannot be used when the robot brakes are on.

10.9.3 Falsely Detected Collisions

Collision Guard might detect a false collision when a collision has not occurred in the following cases:

- Payload information has not been set correctly.
- The ACC motion option has been used, causing jerky robot motion.
- Not enough voltage has been supplied to the controller.
- The payload is larger than the maximum payload for the robot, or the inertia of the payload is too large.
- Very high speed rotations of wrist joints occur with improperly set payload parameters.
- Jerky reverse motion (P[1]->P[2]->P[1]) occurs.

- Linear motion occurs near singularity point where axes revolve in high speed.

10.9.4 Setup

Before you can use Collision Guard, you must set it up. Setup includes:

- Setting the Collision Guard Sensitivity
- Specifying a register in which to set and store the sensitivity value for the COL GUARD ADJUST teach pendant instruction, if desired.

Refer to Table 10–15 for information on Collision Guard Setup Items. Use Procedure 10-15 to set up Collision Guard.

Table 10–15. Collision Guard Setup Items

ITEM	DESCRIPTION
Group default: 1	This item indicates the currently selected group to which the other items in the menu apply. Note that you cannot edit this field directly. Instead, the current group is selected by pressing “F4” (Group) and entering the group number you wish to display. You will only be able to select a group that supports Collision Guard.
Collision Guard Status default: ENABLED	This item specifies whether High Sensitivity Collision Detection is ON or OFF for the currently selected group: • ENABLED indicates that High Sensitivity Collision Detection is ON in all cases (programmed and jogging motion), unless it is turned OFF using the COL DETECT OFF instruction in a teach pendant program. • DISABLED indicates that High Sensitivity Collision Detection is OFF in all cases (programmed and jogging motion) and Standard Collision Detection is used.
Sensitivity default: 100% minimum: 1% maximum: 200%	This item allows you to set the level of sensitivity for Collision Guard for the currently selected group: • The lower the value, the lower the sensitivity. • The higher the value, the higher the sensitivity. In some cases, you can decrease the sensitivity value to eliminate false alarms. In some cases, you can increase the sensitivity value to provide faster response.
Sensitivity Default Register	This item allows you to specify the register that can be used with the COL GUARD ADJUST teach pendant instruction to adjust the sensitivity of Collision Guard within a program. A register number of 0 indicates that the register is not used.

Table 10–15. Collision Guard Setup Items (Cont'd)

ITEM	DESCRIPTION
Collision Guard Error	This item allows you to specify a Digital Output that will be turned ON when a Collision Detect Alarm occurs. A value of 0 indicates that no output will be turned on.
Col. Guard Enabled	This item allows you to specify a Digital Output that will be turned on whenever Collision Guard is enabled via this setup menu, and turned off whenever Collision Guard is disabled via this setup menu. A value of 0 indicates that no output will be used.

Procedure 10-15 Setting Up Collision Guard

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Coll Guard. You will see a screen similar to the following.

```
COL GUARD SETUP     G1     JOINT 100%
                                  1/3
    Group:      1
    Collision Guard status:   ENABLED
  1 Sensitivity:               100%
  2 Sensitivity Macro Reg.:   R[ 0]
  3 Collision Guard Error:    DO[ 0]
  4 Col. Guard Enabled:       DO[ 0]
```

5. **To select a group**, press F3, GROUP, and enter the number of the group you want. If the group you want does not exist, or it includes a robot that has not been tuned for Collision Guard, the group will not be changed and you will see a message similar to the following:

```
Selected group doesn't support HSCD
```

6. **To display help information**, press F2, HELP. When you are finished displaying help, press PREV.
7. Move the cursor to the items you want to set, and set them as desired.

10.9.5 Programmed Motion

You can use the following teach pendant instructions to control Collision Guard during programmed motion:

- COL DETECT ON, COL DETECT OFF
- COL GUARD ADJUST

10.9.5.1 COL DETECT ON, COL DETECT OFF

By default, Collision Guard is enabled.

- To disable Collision Guard, include the COL DETECT OFF instruction in a teach pendant program.
- To enable Collision Guard when it has been disabled previously, include the COL DETECT ON instruction in a teach pendant program. Since Collision Guard is always enabled by default, you need to use the COL DETECT ON instruction only if you have previously used the COL DETECT OFF instruction in the same teach pendant program.

Note The COL DETECT ON and COL DETECT OFF instructions will only apply to the motion groups that are included in the group mask of the teach pendant program that calls them. For example, if COL DETECT OFF is used in a teach pendant program that contains group 2 only in its group mask, then Collision Guard will only be disabled for group 2; it will remain enabled for all other groups.

Also, if the COL DETECT OFF instruction is used within a teach pendant program, and that program is aborted, then Collision Guard will be automatically turned back on. See the following screen for an example.

```
10: J P[1] 100% FINE
11: COL DETECT OFF
12: L P[2] 2000mm/sec CNT100
13: L P[3] 2000mm/sec CNT100
14: L P[4] 2000mm/sec CNT100
15: COL DETECT ON
16: J P[5] 50% FINE
```

10.9.5.2 COL GUARD ADJUST

The COL GUARD ADJUST teach pendant instruction allows you to adjust the sensitivity of Collision Guard within a teach pendant program. The sensitivity value set by this instruction will temporarily override what has been set in the COL GUARD SETUP menu. The instruction can be used in two ways:

- With no arguments. In this case, the new sensitivity value will be obtained from the register whose register number is specified in the COL GUARD SETUP menu.
- With one argument. In this case, the new sensitivity value will be obtained from the argument, which can be either DIRECT (value entered explicitly) or INDIRECT (register number specified) .

As with the COL DETECT ON and COL DETECT OFF instructions, the new sensitivity value will apply only to the motion group(s) that are indicated in the group mask of the program that uses COL GUARD ADJUST. For example, if COL GUARD ADJUST is used in a program whose group mask is group 2 only, then the new sensitivity will apply only to group 2; the sensitivity of all other groups will remain unchanged.

If the group mask of the program that uses COL GUARD ADJUST contains a group that does not support Collision Guard, a warning message will be posted as follows:

```
MOTN-404 Group # does not support HSCD
```

This message is just a warning to indicate that no action was taken on that group.

To use the COL GUARD ADJUST instruction without arguments in a TP program, the following procedure should be used:

- Set the register(s) specified in the COLL GUARD SETUP menu for the appropriate group(s) to the new sensitivity value(s). NOTE: If the TP program is a multiple group program, the appropriate register for each group needs to be set.
- Insert the COL GUARD ADJUST instruction into the TP program at the desired location.

Note If the register number in the COL GUARD SETUP menu is zero for one of the groups being adjusted, then an error will be posted, as follows:

```
MOTN-400 No Coll. Guard Reg. Defined
```

If a valid register number is specified in the setup menu, but the data in that register is not an integer between 1 and 200, then the following error will be posted:

```
MOTN-401 Coll. Guard Reg. Data Error
```

See the following screen for an example.

```
10: J P[1] 100% FINE
11: R[11]=80
12: COL GUARD ADJUST
13: L P[2] 2000mm/sec CNT100
14: L P[3] 2000mm/sec CNT100
15: L P[4] 2000mm/sec CNT100
16: R[11]=100
17: COL GUARD ADJUST
18: J P[5] 50% FINE
```

This example would apply to a single group program (say, a group 1 only program). In the example, it is assumed that the register number has been set to 11 for group 1 in the COL GUARD SETUP menu.

Alternatively, the COL GUARD ADJUST instruction can be supplied with the new sensitivity value directly. An example of how this could be done is as follows:

```
10: J P[1] 100% FINE
11: COL GUARD ADJUST 80
12: L P[2] 2000mm/sec CNT100
13: L P[3] 2000mm/sec CNT100
14: L P[4] 2000mm/sec CNT100
15: R[1]=100
15: COL GUARD ADJUST R[1]
16: J P[5] 50% FINE
```

To use the COL GUARD ADJUST instruction in this way, after inserting the instruction into the program, move the cursor to the right of the instruction. Then, either type in the new sensitivity value directly, or select F3 “INDIRECT” and type in the desired register number.

As stated above, the COL GUARD ADJUST instruction will apply to all groups in the group mask of the TP program that called it. However, if the instruction is used with an argument, the user also has the option to restrict the groups whose sensitivity will be adjusted. This is accomplished by moving the cursor to the argument field and pressing the F1 “GP_MASK” key. Doing so will display the following menu at the top of the Teach Pendant screen:

```
1 Default
2 [GP:]
```

Selecting “1” for “Default” means default behavior of the instruction, i.e. all groups in the group mask are adjusted. Selecting “ 2” inserts a “GP” modifier into the instruction as follows:

```
COL GUARD ADJUST   GP1,2: ##
```

The cursor will then be on the “##”, which is the field where the sensitivity can be directly entered. The user can then move the cursor to the left to highlight the “GP1,2” field. Once this field is highlighted, the function keys will have labels displayed for each available motion group, i.e. F1 will have “GP1”, F2 will have “ GP2”, etc. Pressing the appropriate function key will toggle the appearance of that group number in the “GP1,2” field. Once the “GP1,2” field has been set, only those groups listed in this field will have their sensitivity updated by the instruction.

By default, when a TP program that calls COL GUARD ADJUST is aborted, the sensitivity will be reset to the value specified in the COL GUARD SETUP menu for each group in the program’s group mask. This feature can be disabled by setting the following system variable:

```
$HSCDMNGRP[g].$AUTO_RESET = 0
```

where “ g” refers to the group number.

10.10 COLLISION SKIP

10.10.1 Overview

Collision Skip is used in a variety of applications such as touch sensing, height, depth and width detection, and exploring unknown or changing environments. It uses the features of the Collision Guard option to set up skip conditions so that if a robot collides with a foreign object, you can take different courses of action depending on the skip conditions.

10.10.2 Requirements

To use Collision Skip, the Collision Guard option must be installed and active.

Accurate specifications of the payload characteristics are required to maximize the accuracy of this function and to avoid false detections.

Refer to the *Software Installation Manual* for information on installing these options.

Note Collision Skip can be specified for one joint axis only. Because each call to the skip condition overrides the previous call, make sure you select only the joint axis that will be affected the most by the collision.

10.10.3 Operation

To use Collision Skip, you must place calls to the CLSKP macro in your teach pendant program. The CLSKP macro sets up the skip condition using the following syntax:

```
CLSKP(g, j, max, min)
```

In the above example,

- *g* is the group number,
- *j* is the joint axis number,
- *max* is the maximum value of disturbance torque in a collision-free run, and
- *min* is the minimum value of disturbance torque in a collision-free run.

The following steps provide an overview of the process of using Collision Skip.

1. Set the system variable $MISC_MSTR.$hpd_enb to TRUE and cycle robot power.
2. Insert a call to the CLSKP macro in your program just prior to the line where the collision is expected to occur. Use large values for the minimum and maximum to ensure that the skip condition is not triggered.
3. Run the program between the two points where the collision is expected to occur. Record the collision-free values for disturbance torque from the system variables:
   - $MOR_GRP[g].$min_dis_trq[j]
   - $MOR_GRP[g].$max_dis_trq[j]
4. Replace the minimum and maximum values in the call to the CLSKP macro with the values of the system variables.

Sample Program Using the CLSKP Macro shows a sample program that uses the CLSKP macro.

Sample Program Using the CLSKP Macro

```
!Program CLSKP_EX
!Example program for collision skip function
!Copy the lines of this program and paste them into your program and
modify
!them if necessary.
!
!Teach the point P[98] just before use of collision skip function.
Motion
!type, position number, and speed should be modified if necessary.
!
L P[98] 200 mm/sec FINE
!
!Sub program CLSP(g, j, max, min)
!Call this program just before the motion during which collision occurs.
!Description of arguments:
!g: group number to detect collision
!j: axis number to detect collision
!max: if disturbance torque becomes greater than max, collision is
detected
!min: if disturbance torque becomes less than min, collision is detected
!Set the appropriate value to g and j.  At first, use 99999 and -99999
for max
!and min respectively.  Then, execute the program from P[98] to P[99]
without
!collision, and set $MOR_GRP[g].$MAX_DIS_TRQ[j] to max and
!$MOR_GRP[g].$MIN_DIS_TRQ[j] to min.
!
CALL CLSKP(1,1,99999,-99999)
!
!Teach P[99] so that collision can occur between P[98] and P[99].
Motion type,
!position number, speed, and label number shopuld be modified if necessary.
!
L P[99] 200 mm/sec FINE Skip, LBL[99]
!
!Sub program CLSKP_CL(prnum)
!This program is called when collision occurs.
!Description of arguments:
!prnum: position register number in which the position where collision
occurs
!is stored.
!Modify prnum is necessary.
!
CALL CLSKP_CL(99)
!
!Modify label number if necessary.
!
```

```
LBL[99]
!
!End of CLSKP_EX
```

Listing for the CLSKP Macro shows the listing for the CLSKP macro. It is provided as part of the option and does not require modification.

Listing for the CLSKP Macro

```
! Program CLSKP(g,j,max,min)
R[96] = AR(1)
R[97] = AR(2)
R[98] = AR(3) * 100
R[98] = R[98] / 7282
R[98] = R[98] + 10.1
R[99] = AR(4) * 100
R[99] = R[99] / 7282
R[99] = R[99] - 10.1
SELECT R[96] = 1,JMP LBL[1] ;
         = 2,JMP LBL[2]
         = 3,JMP LBL[3]
         = 4,JMP LBL[4]
         = 5,JMP LBL[5]
END
LBL[1]
SELECT R[97] = 1,JMP LBL[11]
         = 2,JMP LBL[12]
                :
         = 9,JMP LBL[19]
END
LBL[11]
SKIP CONDITION $MISC[1].$HPD_TRQ[1] > R[98] or
                $MISC[1].$HPD_TRQ[1] < R[99]
END
LBL[12]
SKIP CONDITION $MISC[1].$HPD_TRQ[2] > R[98] or
                $MISC[1].$HPD_TRQ[2] < R[99]
End
LBL[2]
      :
LBL[5]
      :
LBL[59]
SKIP CONDITION $MISC[5].$HPD_TRQ[9] > R[98] or
                $MISC[5].$HPD_TRQ[9] < R[99]
```

```
End
! End of CLSKP
```

Sample Macro shows an example macro that can be used to record the position at which the collision occurred in one of the position registers. The position register number is the input argument to this macro. This macro is provided as part of the Collision Skip feature.

Sample Macro

```
!Program CLSKP_CL
PR[AR(1)] = LPOS
!End of CLSKP_CL
```

Use Procedure 10-16 to set up Collision Skip

Procedure 10-16 Setting Up Collision Skip

Conditions

- The Collision Guard option is installed and active. Refer to Section 10.9 for more information.

Steps

1. Set the system variable $MISC_MSTR.$hpd_enb to TRUE. Then turn off the controller and turn it back on so the change can take effect.
2. Teach a point just before the location of the anticipated collision.
3. Teach the next point so that a collision can occur between this point and the point taught in Step 2 .
4. Execute the program. This run should be collision free so that collision-free values for disturbance torque can be recorded from $MOR_GRP system variables.
5. After the collision-free run, record the value for $MOR_GRP[g].$min_dis_trq[j]. Replace min (argument in the call to CLSKP macro) with the value for $MOR_GRP[g].$min_dis_trq[j].
6. After the collision-free run, record the value for $MOR_GRP[g].$max_dis_trq[j]. Replace max (argument in the call to CLSKP macro) with the value for $MOR_GRP[g].$max_dis_trq[j].

   After you have set all the arguments, you can run the program. If there is no collision, the values of the maximum and minimum disturbance torque remain between the specified limits. If a collision occurs (between the two anticipated points in the program), the value of the disturbance torque will not be within the limits and an alternative program can be executed. Use CLSKP_CL to find the exact position at which the collision was detected.

10.11 CONDITION MONITOR FUNCTION

10.11.1 Overview

The Condition Monitor Function monitors the condition of an I/O signal, register value, or alarm status, during teach pendant program execution. As soon as the condition is triggered, the specified teach pendant program is executed and interrupts the current program.

A Condition monitor is defined by two or more teach pendant programs:

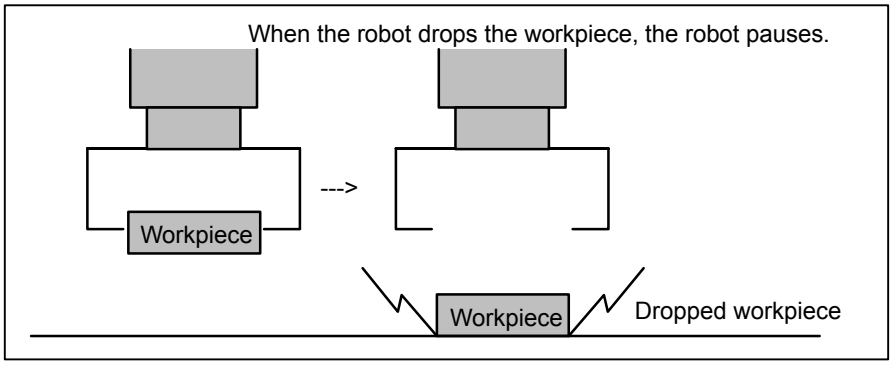
- A condition (CH) program specifying one or more sets of conditions, such as a port or register value. Each set of conditions contains the name of an action program to be called when the condition is satisfied.
- One or more action programs specifying what is to be done when a condition is satisfied.

For example, you can use the condition monitor function as follows:

If a robot is handling a workpiece and drops it, an error message is displayed and the robot pauses. See Figure 10–19 and Sample, Condition Handler, and Action Programs .

```
(CONDITION)<The condition to be monitored>:
[Dropping the work piece] => RI[2] = OFF
(ACTION)<The program executed when the condition is triggered>:
[Error message] => User alarm[ ] & [Pause robot]
```

Figure 10–19. Condition Monitor Function



Sample, Condition Handler, and Action Programs

```
SAMPLE.TP (to perform handling work)
1: MONITOR WORK_DROP       <-+
           :                 |  Monitoring section
```

```
          :                 |
9: MONITOR END WORK_DROP   <-+
[END]
WORK_DROP.CH (condition handler program)
1: WHEN RI[2] = OFF, CALL ROBOT_PAUSE
[END]
ROBOT_PAUSE.TP (action program)
1: SO[2] =ON       ! Notify the peripheral device
2: R[8] = R[8] + 1  ! Count the number of dropped times
3: User alarm[1]     ! Display alarm and pause robot
[END]
$UALRM_MSG[1] (system variable)
$UALRM_MSG[1] ='WORK WAS DROPPED'
```

10.11.2 Monitors

There are two kinds of monitors:

- The **program monitor** is started by a program monitor instruction and stops monitoring when the program executes a MONITOR END instruction or is aborted.
- The **system monitor** is started and ended using the STATUS System Monitor screen.

Program Monitor

Program monitor is used to monitor conditions in each teach pendant program. This monitor depends on the status of program execution. The program monitor only monitors while the program is executing. You start the program monitor using the teach pendant instruction MONITOR. You end the program monitor using the teach pendant instruction MONITOR END or by aborting the program. In the example shown in Example of Program Monitor (SAMPLE.TP) , the system watches the conditions specified by the CH program.

Example of Program Monitor (SAMPLE.TP)

```
5:  MONITOR <ch program name>
 6:  J P[4] 100% CNT100
 7:  J P[5] 100% CNT100
  :
  :
19:  MONITOR END <ch program name>
```

System Monitor

The system monitor does not require a program to be executing for monitoring to take place. When the program is aborted, the program monitor terminates. The system monitor is for monitoring the condition of a system, like a PLC. You can start and end the system monitor from the condition menu. Unlike program monitor, you cannot start and end the system monitor using teach pendant instructions. A MONITOR instruction in the action program of a system monitor can be used to restart the system monitor.

You can use the system variable $TPP_MON.$global_mt to select a mode type at Cold start.

- TYPE1 - If the monitor is executing before the controller is turned off, the system deletes the monitor at Cold start.
- TYPE2 - If the monitor is executing before the controller is turned off, the system starts monitoring at Cold start automatically.

Note You cannot use TYPE1 and TYPE2 together.

Note You can use system monitor and program monitor concurrently.

Changing the Monitor Type

You can change the type of monitor as follows:

```
$TPP_MON.$local_mt = 1 - Program monitor TYPE1 (default)
$TPP_MON.$local_mt = 2 - Program monitor TYPE2
$TPP_MON.$global_mt = 0 - No use system monitor (default)
$TPP_MON.$global_mt = 1 - System monitor TYPE1
$TPP_MON.$global_mt = 2 - System monitor TYPE2
```

10.11.3 Monitor State

Table 10–16 shows the state of the monitor for each operation.

@ : Start the monitor

o : Restart the monitor if the monitor was executing at paused

% : Pause the monitor (It can restart)

x : Cancel the monitor (It cannot restart)

- : It does not change the state of the monitor

Table 10–16. State of Condition Monitoring

OPERATION	Program monitor		System monitor	
	TYPE 1	TYPE 2	TYPE 1	TYPE 2
MONITOR (Teach pendant instruction)	@	@	-	-
START (Function key at condition menu)	o	o	@/o	@/o
Program is PAUSED	%	-	-	-
Program is ABORTED	x	x	-	-
MONITOR END (Teach pendant instruction)	x	x	-	-
PAUSE (Function key on condition menu)	%	%	%	%
END (Function key on condition menu)	x	x	x	x
RESTART (Function key on condition menu)	o	o	-	-
HOT START(Power off on teach pendant program execution)	%	-	-	-
HOT START(Power off at teach pendant program stop)	-	-	-	-
COLD START	x	x	x	-
CTRL START	x	x	x	x

10.11.4 Monitor Instructions

A program monitor is executed using the following two instructions:

- MONITOR <ch program>Start monitoring the conditions taught in the <ch program>.
- MONITOR END <ch program>Stop monitoring the conditions taught in the <ch program>.

You can use the system variable $TPP_MON.$local_mt to change monitoring modes while a program is PAUSED.

- TYPE1 - Stop monitoring when the program is PAUSED.

- TYPE2 - Keeps on monitoring even when the program is PAUSED.

Note You cannot use TYPE1 and TYPE2 together.

10.11.5 Condition Handler Program

You can teach the condition in a program that has a Cond sub type. When you edit the condition handler program, only the WHEN instruction is available.

```
WHEN <condition> CALL <program>
```

In a condition handler program, you can teach multiple WHEN instructions as follows.

```
1: WHEN <cond1> CALL <program1>
2: WHEN <cond2> CALL <program2>
3: WHEN <cond3> CALL <program3>
```

You can connect the multiple conditions using AND/OR as follows.

```
1: WHEN <cond1> AND <cond2> CALL <program1>
2: WHEN <cond1> OR <cond2> OR <cond3> CALL <program2>
```

Note You cannot use both AND and OR in the same WHEN instruction.

10.11.6 Conditions

Figure 10–20 through Figure 10–22 show the conditions that can be monitored.

Figure 10–20. Condition for Register, System Variable, and I/O Parameters

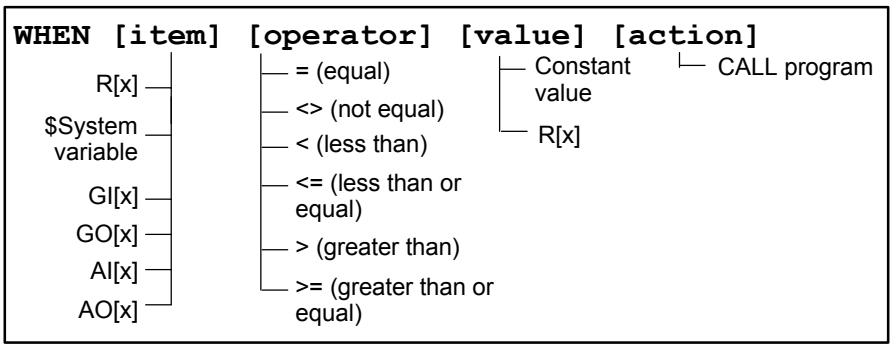


Figure 10–21. Condition2 for I/O

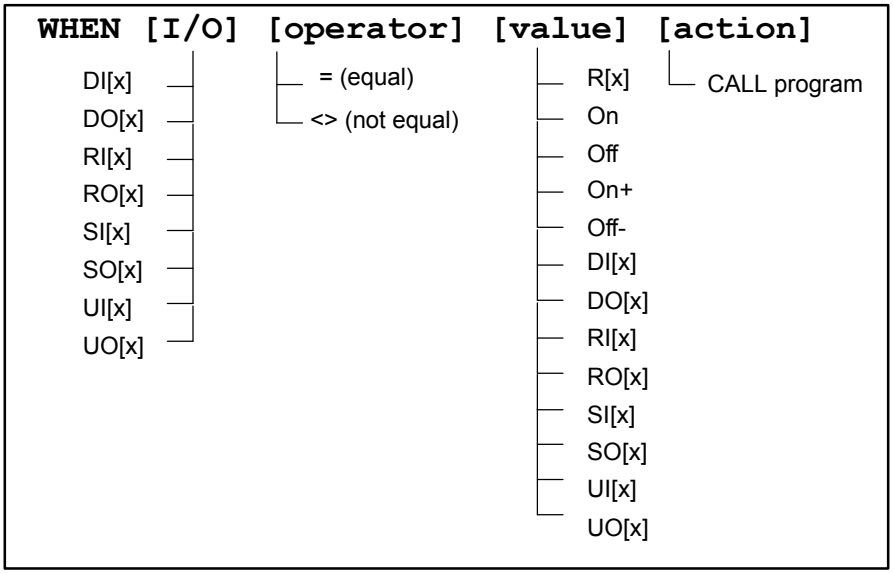
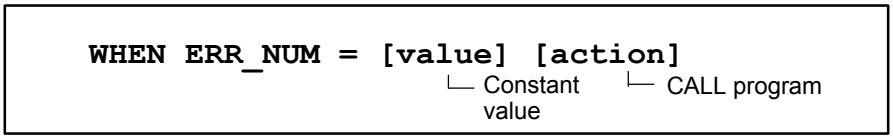


Figure 10–22. Condition for Error status



Error Number

ERR_NUM = aaabbb

aaa : Error facility code (decimal); Refer to Section A.2.2 .

bbb : Error number (decimal)

Example: WHEN ERR_NUM=11006, CALL PROG_A

This refers to the "SRVO-006 Hand broken" error because the SRVO facility code is 11.

If 0 is specified as error number "aaabbb", whenever any error occurs, the condition is satisfied.

10.11.7 Condition Menu

This menu has the following functions:

- Program monitor

— Displays the status of program monitor

— Restarts the program monitor

— Pauses the program monitor

— Ends the program monitor

- System monitor
  - — Displays the status of system monitor
  - — Starts or restarts the system monitor
  - — Ends the system monitor

To select a condition menu

1. Press STATUS.
2. Press F1, [TYPE].
3. Select Condition.

Program Monitor Menu

The Program Monitor menu is shown in Figure 10–23 . This menu lists the running or paused program condition only. Refer to Table 10–17 for a description of the items on the Program Monitor menu.

Note The Program Monitor menu does not display conditions that have not been started.

Figure 10–23. Program Monitor Menu

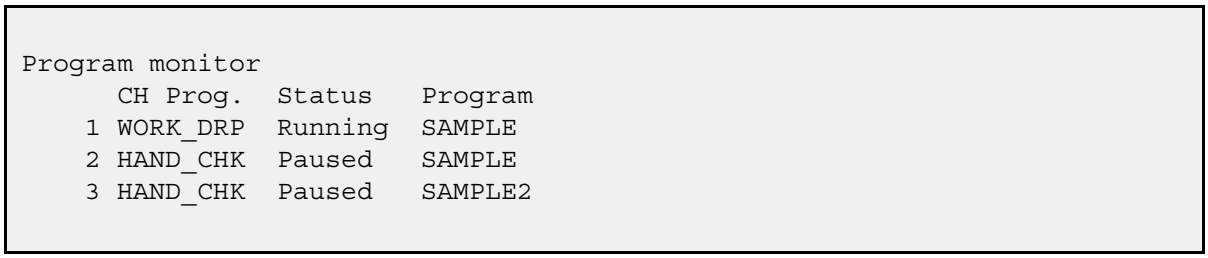


Table 10–17. Program Monitor Menu Items

ITEM	DESCRIPTION
CH Prog.	This item indicates the name of active condition handler program.
Status	This item indicates the status of the condition. • Running : The monitoring of this condition is enabled. • Paused : The monitoring of this condition is disabled.

Table 10–17. Program Monitor Menu Items (Cont'd)

ITEM	DESCRIPTION
Program	This item is the name of the program that starts the condition. If the sub program starts the monitor, the main program name is displayed.
SYSTEM	This item displays the System Monitor screen. **Note** If $TPP_MON.$global_mt is equal to 0, then this function key does not work and will display the message "System monitor is not available."
RESTART	This item restarts the paused condition.
PAUSE	This item pauses the program condition.
END	This item ends this condition. The status is set to canceled and the condition is stopped.

System Monitor Menu

Figure 10–24 shows the System Monitor menu, which displays and allows you to change system monitor information. Table 10–18 lists and describes the items on the System Monitor menu.

Figure 10–24. System Monitor Menu

```
System monitor
      CH Prog.  Status
    1 WORK_DRP  Running
    2 HAND_CHK
```

Table 10–18. System Monitor Menu Items

ITEM	DESCRIPTION
CH Prog.	This item lists the condition programs.
Status	This item indicates the status of the condition. • Running : The monitor of this condition is enabled. • Paused : The monitor of this condition is disabled. • (blank) : This condition has not yet started or has ended.
PROGRAM	This item displays the program condition screen.

Table 10–18. System Monitor Menu Items (Cont'd)

ITEM	DESCRIPTION
START	This item starts or restarts system conditions.
END	This item ends this condition. The status is canceled and the display is cleared.

You can change the type of system monitors by changing the system variable $TPP_MON.$global_mt as follows. You can only change this system variable in the system variable menu at Controlled start.

- $TPP_MON.$global_mt = 0 - No use system monitor (default)
- $TPP_MON.$global_mt = 1 - System monitor TYPE1 Refer to Section 10.11.3 , "Monitor State."
- $TPP_MON.$global_mt = 2 - System monitor TYPE2 Refer to Section 10.11.3 , "Monitor State." You can start and end the monitor at the condition menu in the STATUS menu. Refer to Section 10.11.7 , "Condition Menu."

10.11.8 Restrictions

The multiple conditions taught in the condition handler program are monitored at the same time.

```
1: WHEN <cond1> CALL <program1>
2: WHEN <cond2> CALL <program2>
3: WHEN <cond3> CALL <program3>
```

When the next conditions begin to be monitored before the last conditions are stopped, then both conditions are monitored at the same time.

A program monitor is canceled in the following cases:

- One of the conditions is triggered.
- Execute the "MONITOR END" teach pendant instruction.
- The program is aborted.
- The END function key, on the program monitor screen, is pressed.

Note In the program monitor TYPE1 ($TPP_MON.$local_mt=1), when the program is paused, the program monitor is paused. The program monitor is restarted when the program is restarted.

A system monitor is canceled in the following ways:

- One of the conditions is triggered.

- A Cold start is executed and $TPP_MON.$global_mt = 1.
- The END function key, on the program monitor screen, is pressed.

A program or system monitor can be restarted after it triggers, by having the action program it calls execute a MONITOR teach pendant instruction.

The maximum number of conditions connected with the AND/OR operator is 5. The total number of monitors is limited to 50.

```
WHEN <cond1> AND <cond2> ... AND <cond5>
WHEN <condl> OR <condm> ... OR <condp>
                  :
                  :
WHEN <conds> AND <condt> ... AND <condw>
```

You cannot execute motion instructions in the action program when the robot is moving. You cannot edit the active CH program. The group mask of the action program for a system monitor must be [*,*,*,*,*].

You can specify the group mask of the action program for the program monitor. However, the action program cannot move the robot when the robot is moving.

When the condition is triggered, the monitor state becomes "end." If you want to continue monitoring, you should teach a "MONITOR" instruction in the action program. In this time, the action program should disable the condition. See the following example.

```
MAIN.TP
1: MONITOR MON1
 :
9: MONITOR END MON1


MON1.Cond
1: WHEN R[1]=1 CALL ACT1


ACT1.TP
1: R[1]=0 <--- disable the condition
2:
3: ( action )
4:
5: MONITOR MON1 <--- restart monitor
```

You cannot execute to the CH program directly.

Use Procedure 10-17 to create a condition handler program.Use Procedure 10-18 to create an action program. Use Procedure 10-19 for an example of how to create a condition handler program. Use Procedure 10-20 to start a condition handler program from a teach pendant program.

Procedure 10-17 Creating a Condition Handler Program

Steps

1. Press SELECT.
2. Press F2, CREATE.
3. Enter the program name (CH program name).
4. To display program header information,
   a. Press F2, DETAIL.
   b. Move the cursor to the sub type and press F4, [CHOICE]. You will see a screen similar to the following.

```
Sub Type
1 None
2 Macro
3 Cond
4
Program Detail
  1 Program Name          [CHK_CELL]
  2 Sub Type:             [        ]
```

   c. Select cond.

   Note If you set the sub type to cond, the system sets the group mask to [*,*,*,*,*] automatically. You cannot change the group mask.

5. When you have finished entering program information, press F2, END.
6. Press F1, [INST]. You will see a list of WHEN instructions. See the following screen for an example.

```
WHEN statement
1 WHEN ...=...        5 WHEN ...>...
2 WHEN ...<>...       6 WHEN ...>=...
3 WHEN ...<...
4 WHEN ...<=...
```

Execution Sequence

The following is the sample of the program monitor.

Condition: DI[1] turns on

Action: DO[1] turns on

Procedure 10-18 Creating an ACTION Program

Steps

1. Press SELECT.
2. Press F2, CREATE.
3. Type the program name (for example ACT).
4. Display program header information to change the group mask.
   a. Press F2, DETAIL.
   b. Change the group mask to **[*,*,*,*,*]**
5. Teach the following instruction.

```
ACT.TP (group mask = [*,*,*,*,*])
1: DO[1]=ON
```

Note The group mask of the action program for the system monitor must be set to [*,*,*,*,*].

Procedure 10-19 Creating a Condition Handler Program (Example)

Steps

1. Press SELECT.
2. Press F2, CREATE.
3. Enter the program name (for example COND1).
4. To display program header information, press F2, DETAIL.
   a. Move the cursor to sub type and press F4, [CHOICE].
   b. Select cond.
   c. Press F2, END.
5. Teach the instruction.

```
COND1.TP(sub type = COND, group mask = [*,*,*,*,*])
1: WHEN DI[1]=ON+,CALL ACT
```

```
COND1
[End]
```

6. Press F1, [INST].

```
WHEN statement
1 WHEN ...=...        5 WHEN ...>...
2 WHEN ...<>...       6 WHEN ...>=...
3 WHEN ...<...        7
4 WHEN ...<=...       8
COND1
[End]
Select item
```

7. Select WHEN ...=....

```
WHEN statement
 1 R[ ]                5 RI[  ]
 2 DO[  ]              6 GO[  ]
 3 DI[  ]              7 GI[  ]
 4 RO[  ]              8 --next page--
COND1
   1:  WHEN =... ...
[End]
Select item
```

8. Select DI[ ] and teach the rest of the instruction. See the following screen for an example.

```
COND1
   1:  WHEN DI[1]=ON+,CALL ACT
[End]
Select item
```

Procedure 10-20 Starting a Condition Handler Program from a Teach Pendant Program

Steps

1. Press SELECT.
2. Press F2, CREATE.
3. Enter the program name (for example MAIN.TP).
4. Teach the instruction. For example:

```
MAIN.TP
1: MONITOR COND1
2: WAIT 10.00(sec)
3: MONITOR END COND1
```

```
MAIN
[End]
```

5. Press F1, [INST]. You will see a screen similar to the following.

```
Instruction
 1 Registers           5 JMP LBL
 2 I O                 6 CALL
 3 IF SELECT           7
 4 WAIT                8 MONITOR/MON. END
MAIN
[End]
Select item
```

6. Select MONITOR/MON. END. You will see a screen similar to the following.

```
MONITOR statement
 1 MONITOR             5
 2 MONITOR END         6
 3                     7
 4                     8
MAIN
[End]
Select item
```

7. Select MONITOR and display the list of CH programs.

```
Cond. PROGRAM list
 1 COND1               5
 2                     6
 3                     7
 4                     8
MAIN
   1:  MONITOR
[End]
Select item
```

8. Teach the following program.

```
MAIN
   1:  MONITOR COND1
   2:  WAIT  10.00(sec)
   3:  MONITOR END COND1
[End]
Select item
```

9. Start the "MAIN" program.

10. If you turn on the DI[1], the DO[1] will turn on while the program executes the second line.

10.12 COORDINATES OFFSET FUNCTION

10.12.1 Using the Coordinates Offset Function

The coordinates offset function changes either the tool coordinate system or the user coordinate system for a range of motion instructions in a program for which teaching has been completed. The function then converts the positional data so that the TCP position does not change due to the shift between the original and changed coordinate systems.

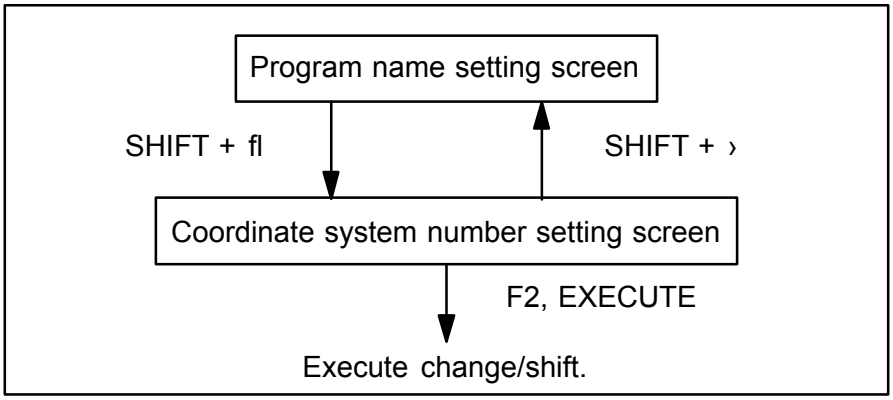
Types of Coordinates Offset

The following two kinds of coordinates offset are available:

- TOOL OFFSET - Changes the **tool** coordinate system number and positional data in a teach pendant program.
- UFRAME OFFSET - Changes the **user** coordinate system number for the positional data in a teach pendant program.

Coordinates offset is executed on the TOOL/UFRAME OFFSET screens (UTILITIES, Tool offset/Frame offset). The screens are switched as shown in Figure 10–25 .

Figure 10–25. Coordinates Offset Screens



Coordinates Offset

The coordinates offset function performs the following:

- Changes the tool coordinate system number or user coordinate system number for the positional data (Cartesian coordinates) in all or a range of motion instructions in an existing program.
- If the positional data is specified with joint coordinates, converts the data according to the shift resulting from the tool or user coordinate system change.
- Inserts the results of the conversion into a new or existing program.
- Executes the same conversion for other programs, if necessary.

Converting the Positional Data

The positional data is converted according to the following rules:

Position and attitude

- Positional data specified with Cartesian coordinates is converted to Cartesian coordinates. Positional data specified with joint coordinates is converted to joint coordinates.
- If the converted joint coordinates fall outside the operating range, the corresponding positional data is assumed to be untaught. For Cartesian coordinates, the converted position is stored as is.

- The positional data in the position registers is not converted.
- For motion instructions that include the incremental motion option, positional data specified with joint coordinates is assumed to be untaught.

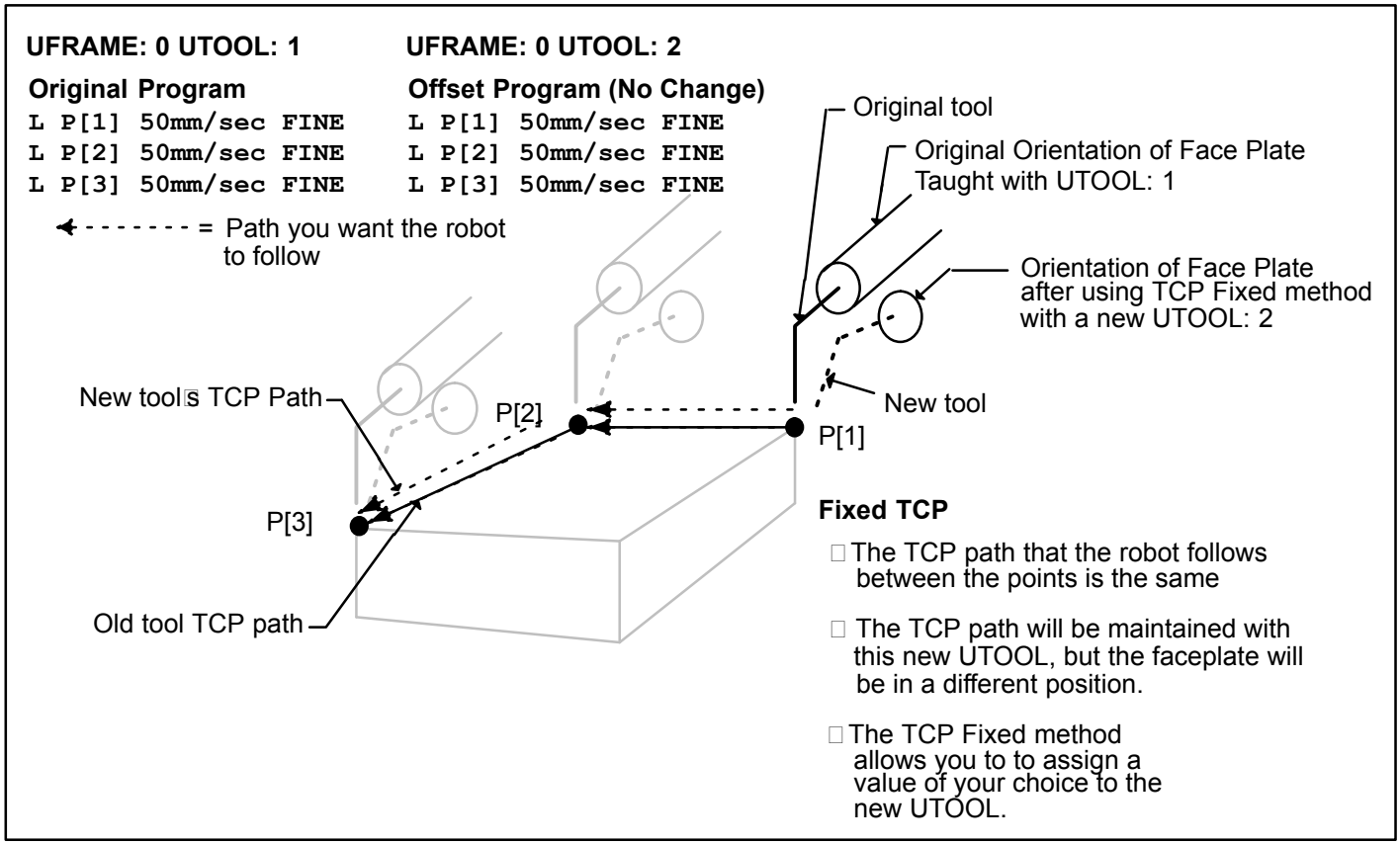
Axis location and rotation speed of positional data specified with Cartesian coordinates

- The same format is used both before and after conversion.
- If the wrist axis is rotated through 180° or more as a result of conversion, the rotation speed for the axis is optimized; a message is then displayed prompting you to select whether to use the optimized rotation speed.

For **UTOOL OFFSET**, you can select either of the following positional data conversion methods:

- **TCP fixed**: This method lets you specify a new TOOL frame number of your choice for use with a new or damaged tool. Programmed positions are not changed. The same TCP path will be maintained with this new UTOOL, but the faceplate position will be different. See Figure 10–26 .

Figure 10–26. TCP Fixed Method



- **Robot fixed**: This method lets you specify the TOOL frame number to use with the current tool. Programmed positions are automatically adjusted to maintain the desired path. The robot's motion does not change. See Figure 10–27 and Figure 10–28 .

Figure 10–27. Robot Fixed Method - Example 1

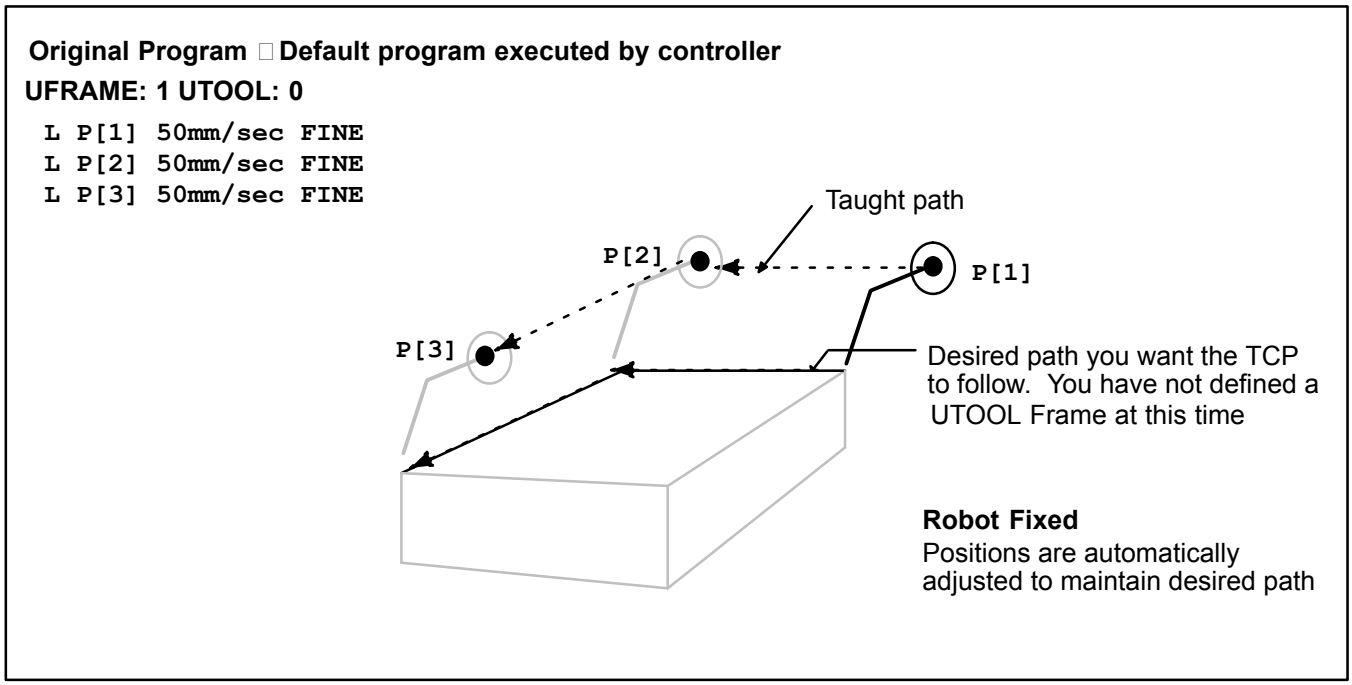
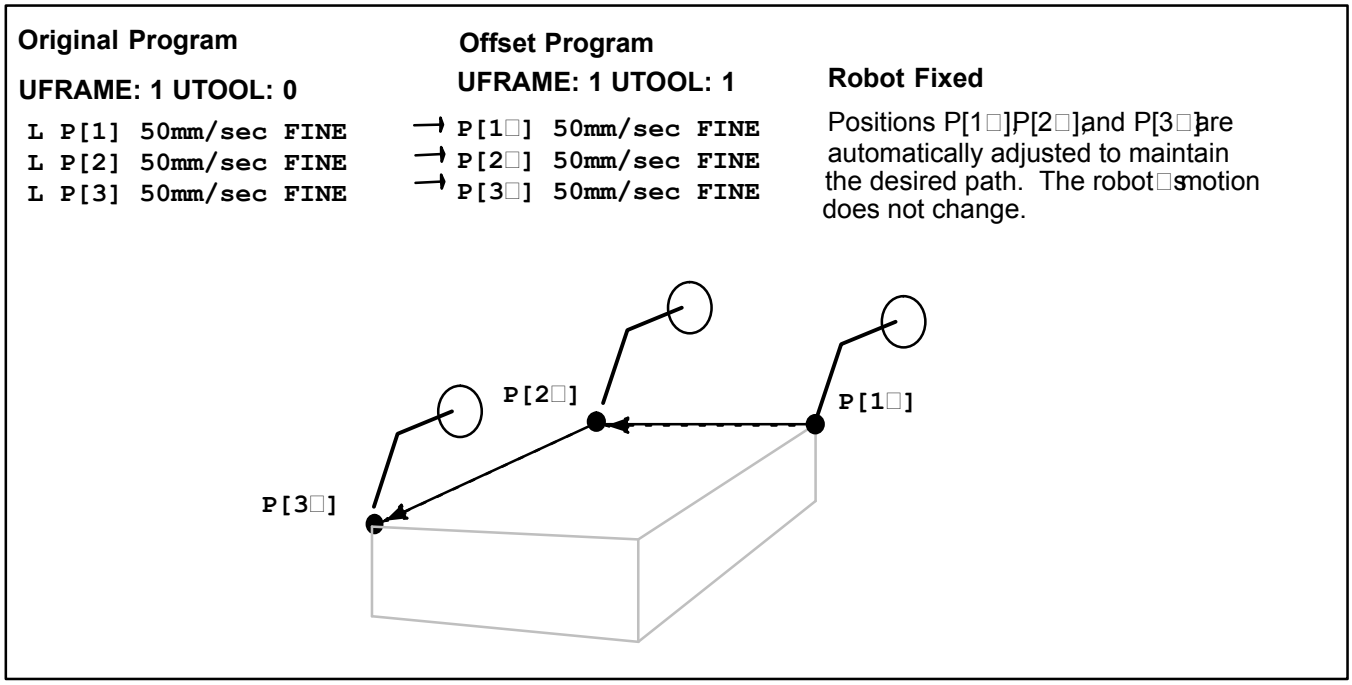


Figure 10–28. Robot Fixed Method - Example 2



For **UFRAME OFFSET**, you can select whether the positional data is to be converted. Refer to Section 10.12.3 .

- **Convert**: The position data is converted so that the TCP position does not change.
- **Not convert**: The position data is not converted even when the coordinate system number is changed.

10.12.2 Tool Frame Offset Function

You can change the tool coordinate system number and positional data from the TOOL OFFSET screen.

Table 10–19 lists and describes the items you set to perform the tool frame offset function. Use Procedure 10-21 to perform the tool frame offset function. Refer to Section 3.1.2.1 for information on setting up a tool frame.

Table 10–19. Tool Offset Screen Items

ITEM	DESCRIPTION
Program Name Setting Screen	
Original Program	This item indicates the name of the program that will be offset or shifted.
Range	This item specifies the portion of the program that will be offset or shifted: • WHOLE - offsets or shifts the entire program • PART - offsets or shifts part of the program
Start line	This item is not used when Range is set to WHOLE.
End line	This item is not used when Range is set to WHOLE.
New Program	This item is the name of the program that results when you offset or shift the Original Program. If you want the resulting offset or shifted program to replace the Original Program, make the New Program name the same as the Original Program name.
Insert line	This item is used only when all of the following conditions exist: • You have not entered a name for the new program, in which case the data conversion will be performed on the currently selected program, or you have entered the name of a program that already exists for the New Program name. • You have selected the Robot Fixed method as the data conversion type. • You have executed the data conversion
Coordinate System Number Setting Screen	
Old UTOOL Number	This item is the number of the UTOOL that was used when the positions in the Original Program were recorded.

Table 10–19. Tool Offset Screen Items (Cont'd)

ITEM	DESCRIPTION
New UTOOL Number	This item is the number of the UTOOL that will be used to offset or shift the program. You must have defined this UTOOL prior to using it.
Convert Type	This item specifies the kind of positional data conversion that will be performed during the offset or shift: • **TCP fixed** - The TCP is maintained during conversion. This means that robot joint positions will change, but Cartesian positions will be fixed. TCP fixed mode can be used, for example, when a damaged hand has been replaced. Specify the tool coordinate system number of the damaged hand for Old UTOOL number and the tool coordinate system number of the replacement hand for New UTOOL number. Then, execute the tool change or shift in TCP fixed mode. The result will be that the TCP of the new tool will move to the originally taught position. • **Robot fixed** - The robot joint positions are maintained during conversion. Robot fixed mode can be used, for example, when a program has been taught using a tool coordinate system other than that for the mounted hand, after which the tool coordinates have been corrected. Specify the tool coordinate system number used when the program was taught for Old UTOOL number and the corrected tool coordinate system number for New UTOOL number. Then, execute the tool change or shift in Robot fixed mode. The program is modified so that the robot moves according to the corrected tool coordinate system, without changing the resultant robot movement.

Procedure 10-21 Executing a Tool Change or Shift

Conditions

- The new UTOOL you want to use has been defined.
- A program is to be shifted. See the following screen for an example program.

```
TEST1
  1: J P[1] 100% FINE
  2: J P[2] 70% CNT50
  3: L P[3] 1000cm/min CNT30
  4: L P[4] 500mm/sec FINE
  5: J P[1] 100% FINE
[End]
```

Steps

1. Press MENUS.
2. Select UTILITIES.
3. Press F1, [TYPE].
4. Select Tool offset. You will see a screen similar to the following.

```
TOOL OFFSET
Program
1   Original Program:         [TEST1  ]
2     Range:                       WHOLE
3     Start  line:(not used)       ***
4     End    line:(not used)       ***
5   New Program:              [TEST2  ]
6     Insert line:(not used)       ***
Use shifted up, down arrows for next page
```

5. Move the cursor to the original program, and press ENTER. Use the appropriate function keys to type the program name, and press ENTER.
6. Move the cursor to the new program, and press ENTER. Use the appropriate function keys to type the program name, and press ENTER.
7. Hold down the SHIFT key and press the down arrow key to display the coordinate system number setting screen. To return to the program name setting screen, hold down the SHIFT key and press the up arrow key.

```
TOOL OFFSET
UTOOL number
1   Old UTOOL number:             1
2   New UTOOL number:             2
3   Convert type          TCP fixed
Use shifted up, down arrows for next page
```

8. Move the cursor to the Old UTOOL number, type the tool frame number, and press ENTER.
9. Move the cursor to the New UTOOL number, type the tool frame number, and press ENTER.
10. **To convert data using the TCP Fixed method**, move the cursor to Convert Type, press F4, [CHOICE], select TCP Fixed, and press ENTER.
    a. **To execute the transformation**, press F2, EXECUTE, and then press F4, YES, to confirm.
    b. **To cancel the transformation**, press F5, NO, as the answer to the confirmation question.

11. **To convert data using the Robot Fixed method**, move the cursor to Convert Type, press F4, [CHOICE], select Robot Fixed, and press ENTER.

a. **To execute the transformation**, press F2, EXECUTE, and then press F4, YES, to confirm.

b. **To cancel the transformation**, press F5, NO, as the answer to the confirmation question.

Note If the "Insert line not set" message is displayed, you are about to perform the data conversion on the original program, or a program that already exists. You will have to enter the number of the line to insert.

12. If the rotation speed has changed (been optimized) as a result of conversion, you are prompted whether to use the new rotation speed. See the following screen for an example.

```
Select P[3]:J5 angle.(deg183)
  183°-177°*uninit*
QUIT>
```

Select the action you want to take:

- **To use the new, optimized rotation speed**, press F1. The label above F1 indicates that angle that corresponds to the optimized rotation.
- **To use the original rotation speed**, press F2. The label above F2 indicates the angle that corresponds to the original rotation speed.
- **To write the data as untaught data**, press F3, *uninit*.
- **To cancel conversion**, press F5, QUIT.

13. To clear all shift settings, press NEXT, > then press F1, CLEAR.

Note After TOOL OFFSET has been executed, the current tool coordinate system number is changed to the newly specified number.

10.12.3 User Frame Offset Function

You can change the user coordinate system number and positional data from the UFRAME OFFSET screen.

Table 10–20 lists and describes the items you set to perform the user frame offset function. Use Procedure 10-22 to perform the user frame offset function. Refer to Section 3.1.2.1 for information on setting up a tool frame.

Table 10–20. User Frame Offset Screen Items

ITEM	DESCRIPTION
Program Name Setting Screen	
Original Program	This item is the name of the program that will be offset or shifted.
Range	This item specifies the portion of the program that will be offset or shifted: • WHOLE - offsets or shifts the entire program • PART - offsets or shifts a part of the program
Start line	This item is not used.
End line	This item is not used.
New Program	This is the name of the program that results when you offset or shift the Original Program. If you want the resulting offset or shifted program to replace the Original Program, make the New Program name the same as the Original Program name.
Insert line	This item is used only when all of the following conditions exist: • You have not entered a name for the new program, in which case the data conversion will be performed on the original program, or you have entered the name of a program that already exists for the New Program name. • You have selected the Robot Fixed method as the data conversion type. • You have executed the data conversion
Coordinate System Number Setting Screen	
Old UTOOL Number	This item is the number of the UTOOL that was used when the positions in the Original Program were recorded.
New UTOOL Number	This item is the number of the UTOOL that will be used to offset or shift the program. You must have defined this UTOOL prior to using it.
Convert Position Data	This item specifies whether to convert the positional data during the user frame offset: • **YES** - Converts the positional data so that the TCP does not change during the offset or shift. • **NO** - Does not convert the positional data when the coordinate system is changed.

Procedure 10-22 Executing a User Coordinate Change or Shift

Conditions

- The new UFRAME you want to use has been defined.
- A program is to be shifted. See the following screen for an example program.

```
TEST1
  1: J P[1] 100% FINE
  2: J P[2] 70% CNT50
  3: L P[3] 1000cm/min CNT30
  4: L P[4] 500mm/sec FINE
  5: J P[1] 100% FINE
[End]
```

Steps

1. Press MENUS.
2. Select UTILITIES.
3. Press the F1, [TYPE].
4. Select Frame offset. You will see a screen similar to the following (Program Name Setting screen).

```
UFRAME OFFSET
Program
1   Original Program:         [TEST1]
2     Range:                       WHOLE
3     Start  line:(not used)       ***
4     End    line:(not used)       ***
5   New Program:              [TEST2]
6     Insert line:(not used)       ***
Use shifted up, down arrows for next page
```

5. Move the cursor to the original program, and press ENTER. Use the appropriate function keys to type the program name, and press ENTER.
6. Move the cursor to the new program, and press ENTER. Use the appropriate function keys to type the program name, and press ENTER.

7. Hold down the SHIFT key and press the down arrow key to display the coordinate system number setting screen. To return to the program name setting screen, hold down the SHIFT key and press the up arrow key. You will see a screen similar to the following.

```
UFRAME OFUFRAME OFFSET
 UFRAME number
1   Old UFRAME number:             1
2   New UFRAME number:             2
3   Convert Position data (Y/N):   YES
Use shifted up, down arrows for next page
```

8. Move the cursor to the Old UFRAME number, type the tool frame number, and press ENTER.
9. Move the cursor to the New UFRAME number, type the tool frame number, and press ENTER.
10. **To execute the UFRAME offset transformation without converting positional data,** move the cursor to Convert Position data (Y/N) and press F5, NO.
    a. **To execute the transformation**, press F2, EXECUTE, and then press F4, YES, to confirm.
    b. **To cancel the transformation**, press F5, NO, as the answer to the confirmation question.
11. **To execute the UFRAME offset transformation with conversion of positional data,** move the cursor to Convert Position data (Y/N) and press F4, YES.
    a. **To execute the transformation** , press F2, EXECUTE, and then press F4, YES, to confirm.
    b. **To cancel the transformation** , press F5, NO, as the answer to the confirmation question.

    Note If the "Insert line not set" message is displayed, you are about to perform the data conversion on the original program, or a program that already exists. You will have to type the number of the line to insert.

12. If the rotation speed has changed (been optimized) as a result of conversion, you are prompted whether to use the new rotation speed. See the following screen for an example.

```
Select P[3]:J5 angle.(deg183)
  183°-177°*uninit*
QUIT>
```

Select the action you want to take:

- **To use the new, optimized rotation speed**, press F1. The label above F1 indicates that angle that corresponds to the optimized rotation.

- **To use the original rotation speed**, press F2. The label above F2 indicates the angle that corresponds to the original rotation speed.
- **To write the data as untaught data**, press F3, *uninit*.
- **To cancel conversion**, press F5, QUIT.

13. To clear all shift settings, press NEXT, > then press F1, CLEAR.

Note After FRAME OFFSET has been executed, the current user coordinate system number is changed to the newly specified number.

10.13 CYCLE TIME LOGGING AND DISPLAY

10.13.1 Overview

The Cycle Time Logging and Display function allows you to track and display teach pendant program cycle times. Cycle times are tracked based on individual cycles and hourly averages. Multiple cycles can be observed to view trends and analyze data.

Typically, the *i*Pendant is used to display the cycle time data graphically. However, raw cycle time data can also be observed.

By default the cycle time is tracked from the start of a teach pendant program until the program ends. However, you can adjust this by adjusting the $CY_CONFIG system variable.

The tracked cycle time is based on multiple categories. See Table 10–21 . Table 10–22 lists the items displayed on the Cycle Time Display screen.

Table 10–21. Tracked Cycle Time Categories

Category	Description
Motion	This indicates the amount of motion time in the cycle.
Process	This indicates the amount of process time including welding time.
Appl	This indicates the amount of application-specific time executed.
SP 1	
SP 2	
Wait	This indicates the amount of time spent waiting for I/O or a data event.
Delay	This indicates the fixed TPP delay time.
KAREL	This indicates the amount of time spent executing KAREL logic.

Table 10–21. Tracked Cycle Time Categories (Cont'd)

Category	Description
Logic	This indicates the amount of time spent executing general logic.
HOMIO	For Spot Welding programs, this indicates the time to complete the housekeeping routine in preparation for the next cycle.
IDLE	For Spot Welding programs, this indicates the time after the HOME position is reached until the next cycle start is issued.
Fault	Fault time is tracked but it is not considered part of the total cycle time. Fault time will not be recorded unless the cycle is resumed. If the cycle is aborted, fault time will not be generated.

Table 10–22. Cycle Time Tracking Screen Items

Item	Description
Program	This item indicates the current program that is being tracked if cycle mode is selected.
Mode	This item indicates whether the cycles are being tracked per cycle or per hour. One, ten, or 100 cycles can be displayed. It can also display one, ten, or 100 hours.
Date	This item indicates the date of the currently highlighted cycle or hour.
Time	This item indicates the time the currently highlighted cycle was executed.
Range (TOTAL)	This item is the total number of items being tracked as *n/m* where *n* is the currently tracked cycle or hour and *m* is the total number of cycles or hours.
Samples	This item indicates the number of hourly samples used in the currently displayed average of hours.
[PROG]	This function key allows you to change the name of the program for which cycle time information is displayed.
[ZOOM]	This function key allows you to change the displayed cycle time information. You can select to display one hour, ten hours, a hundred hours, one cycle, ten cycles, or a hundred cycles.
[CATGRY]	This function key allows you to choose which set of cycle time information is displayed. You can choose Total which displays the total of all of the categories and the cycle time for each individual category. Or, you can choose an individual category to be displayed. When an individual category is displayed, depending on the currently selected mode, the category name will be displayed on the top right corner, and cycle time information for that category only will be displayed. For example, if the Motion category is displayed in ten cycle mode, a screen similar to the following will be displayed. You can choose to display the following categories: Samples, Motion, Process, Appl, Wait, Delay, KAREL, Logic, HOMIO, Idle, and Fault.

Table 10–22. Cycle Time Tracking Screen Items (Cont'd)

Item	Description
[PAGE]	This function key allows you to display cycle time data in a variety of formats. To display cycle time data on the *i* Pendant, Cycle Graphics should be displayed. Other formats include Cycle Text, and Average Text. Line Text is only available in Text mode. If you select hourly data and cycle text, the .DG files will be displayed. CYCAVE.DG displays the hourly data, and CYCLES.DG displays the cycle data. These files can also be displayed using the Web Server on the *i*Pendant.
[UPDATE]	This function key allows you to load data or resize the allocated area to store data. If you want to change the size of the allocated area, 1. Press FCTN and select SAVE to save the current data. 2. Change the system variable to the adjusted size. Change CY_CONFIGNUMCYCLES for cycle by cycle data, and CY_CONFIGNUMAVERAGES for hourly data. These values designate the number of cycles or averages which are kept. By default 100 cycles and 100 averages are kept. This uses SRAM memory so the higher the number the more SRAM that will be consumed. 3. Display the cycle time display again, press >, NEXT, and then press F3, [UPDATE]. 4. Select either Load Cycle Data, or Clear Cycle Data.
RECORD	This function key allows cycle data to be recorded to the teach pendant program area. This provides reference cycle information displayed as a blue line on the screen. The blue line will be displayed when any category is selected. It will not be displayed when Total is selected. This function key is only displayed in one cycle mode as long as the program currently displayed on the Cycle Time Data screen still exists on the controller.

10.13.2 Modes

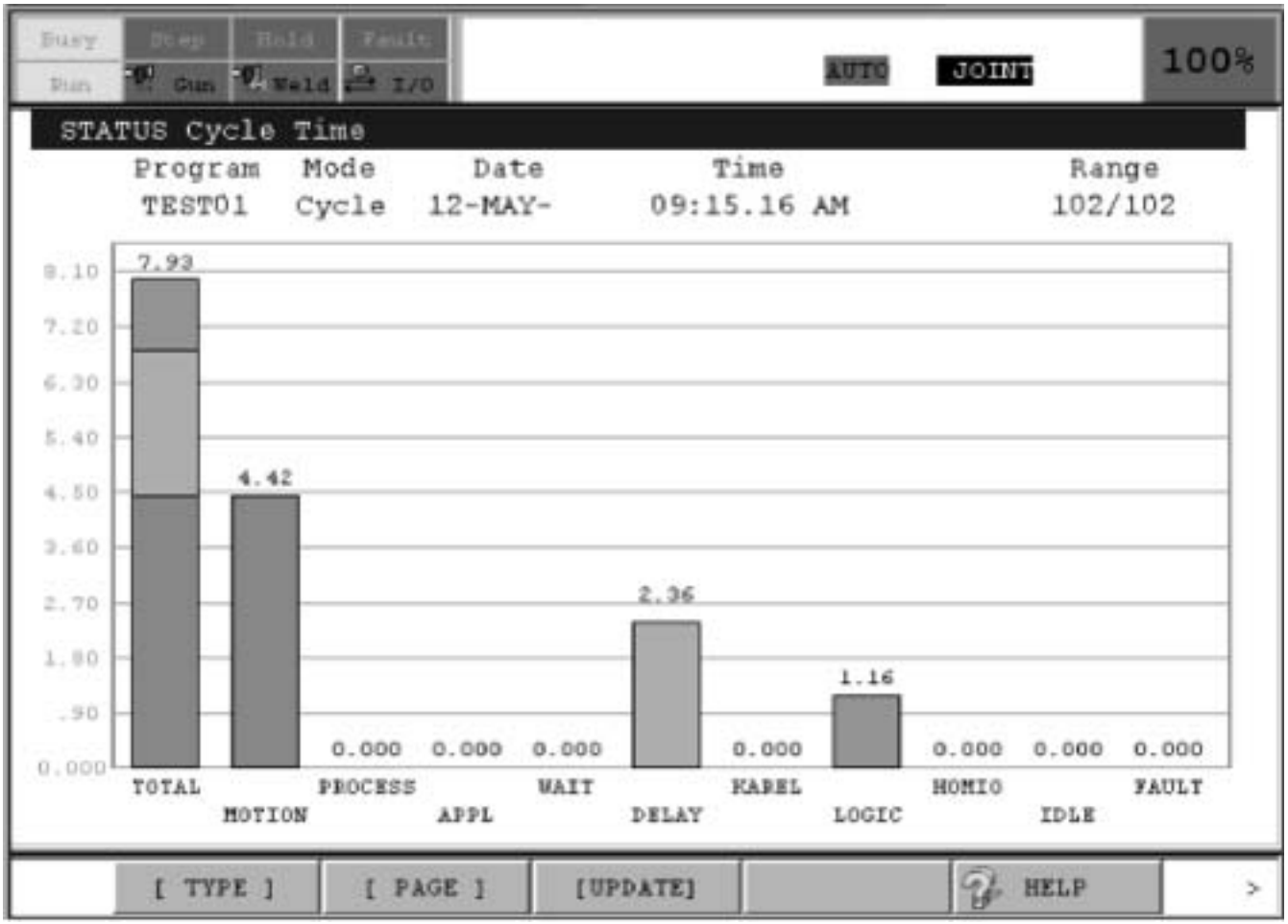
Cycle time data can be displayed by cycles, by hours, or line by line.

10.13.2.1 Cycles

If you display cycle time data by cycle, you can display data for one cycle, ten cycles, or a hundred cycles at a time.

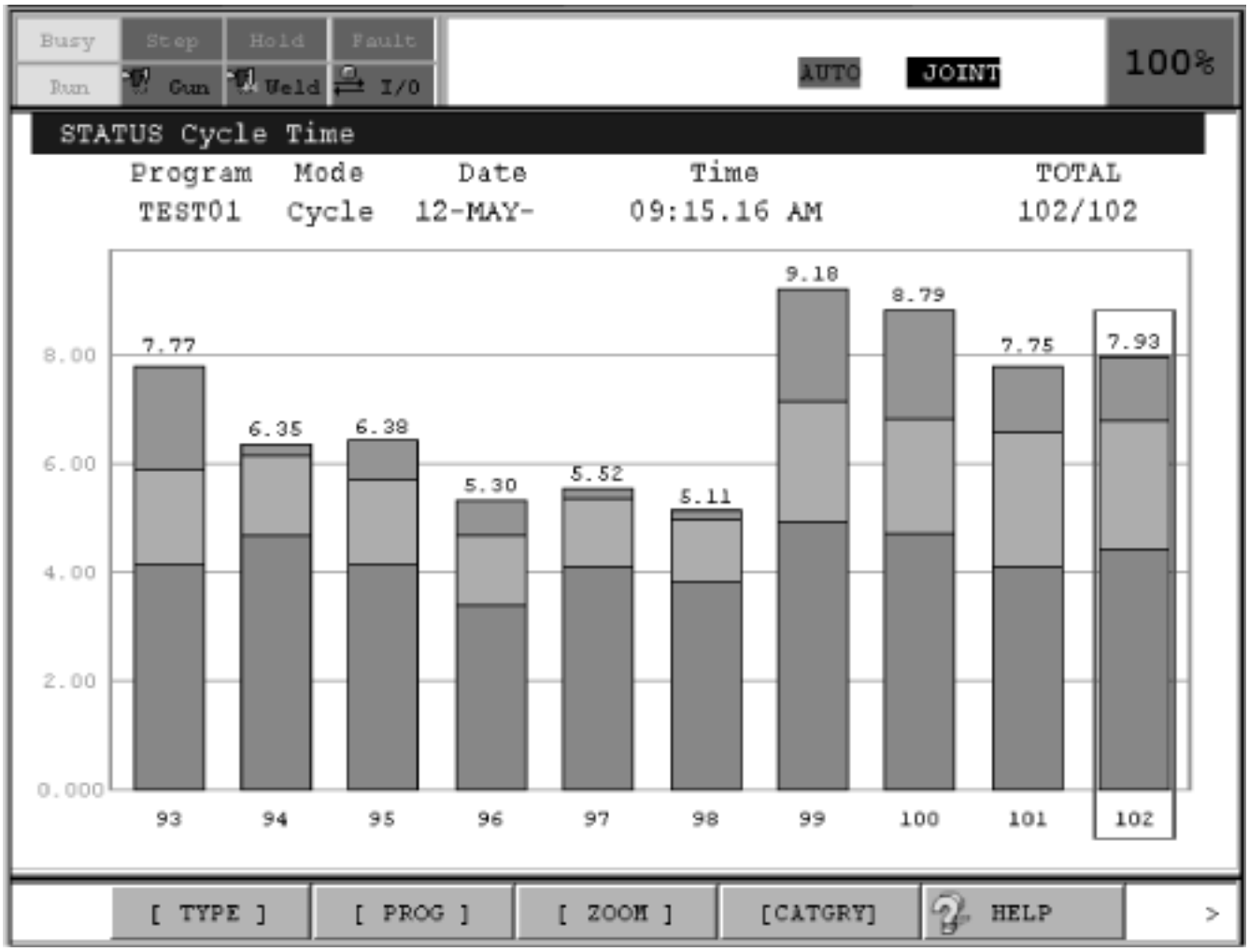
The following screen displays the total cycle time for the current cycle in One Cycle mode. It also displays the total cycle time broken down by category. Each category is represented by a different color.

Figure 10–29. Total Cycle Time in One Cycle Mode



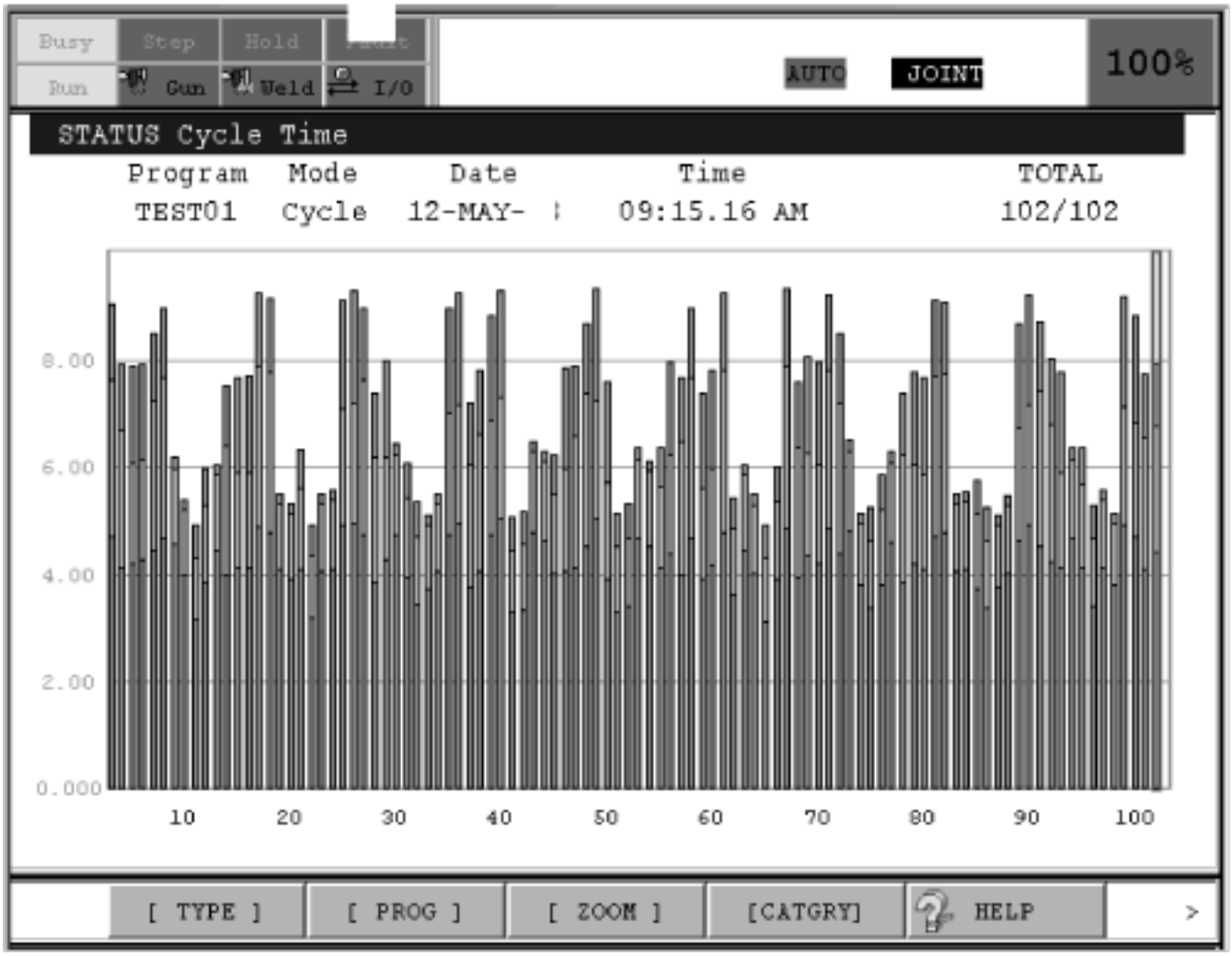
The following screen displays Ten Cycle Mode.

Figure 10–30. Ten Cycle Mode



The following screen displays Hundred Cycle Mode.

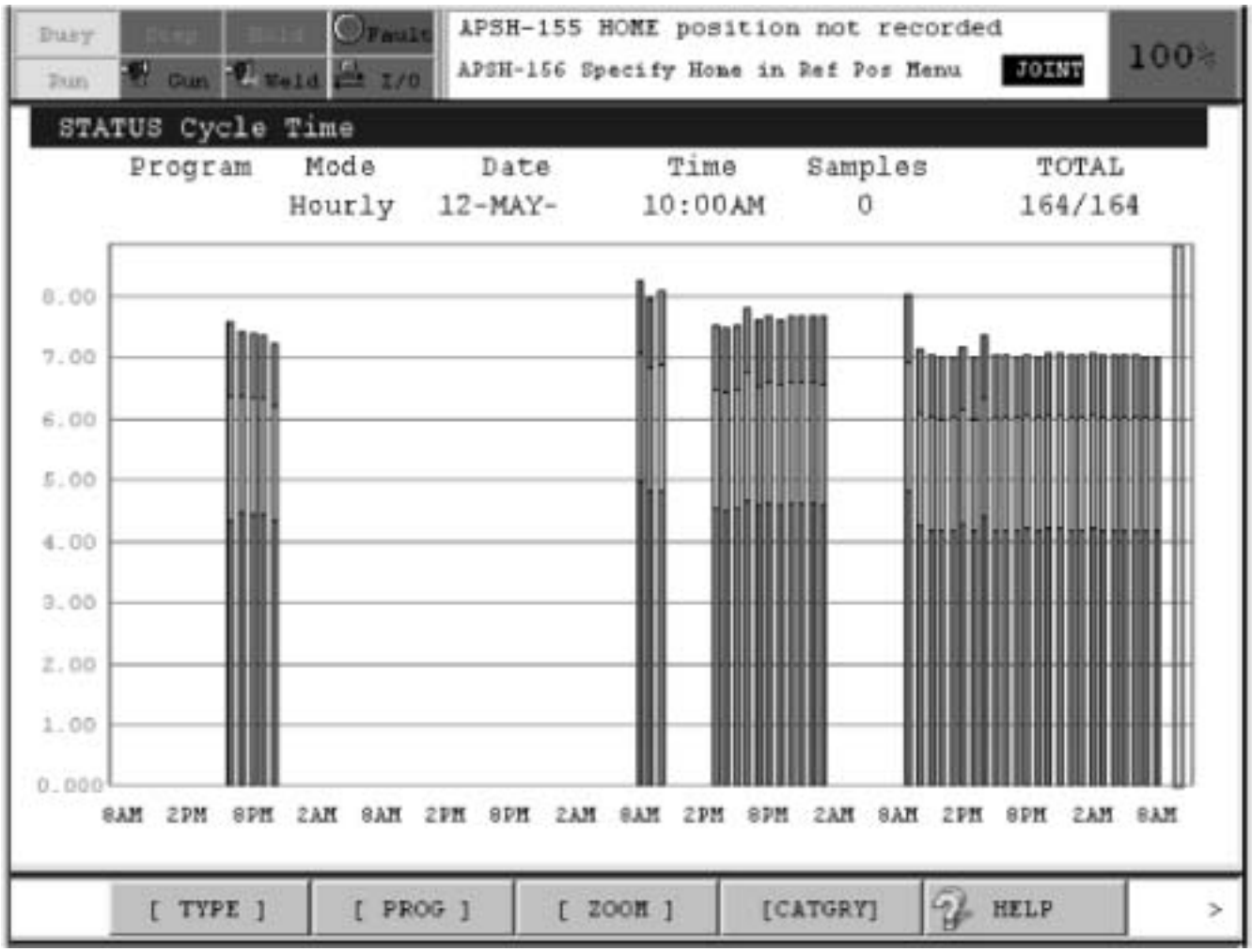
Figure 10–31. Hundred Cycle Mode



10.13.2.2 Hours

If the current mode is by hour, you can display cycle time data for one hour, ten hours, or one hundred hours. Hourly mode is only available if hourly data exists. See Figure 10–32 for an example of hourly mode.

Figure 10–32. One Hundred Hour Mode



10.13.2.3 Line by Line Mode

If the $lineenable=TRUE and $numlines>0, line-by-line cycle time tracking will be enabled. This allows you to track a defined number of lines in the current program. This information is stored in cycline.dg on DRAM and is lost when the controller is turned off. This file can only be viewed using the Web Server or by selecting Line Text from the [PAGE] function key.

10.13.3 Tracking Cycle Time

Procedure 10-23 Tracking Cycle Time on the *i*Pendant

Conditions

- You are using an *i*Pendant to display the cycle time data graphically.
- The program you want to track has been executed at least one time.

1. Press MENUS.
2. Select STATUS.
3. Press F1, [TYPE].
4. Select Cycle Time.
5. Set the name of the program for which you want cycle time data displayed. By default, ALL programs are selected. To select a single program to display, press F2, [PROG], and select the name of the program.
6. **To change the cycle time display between cycles and hours** , press F3, [ZOOM] and select the kind of information you want displayed. After you have made a display selection (cycles or hours) you can quickly change the display by using the arrow keys. The up and down arrow keys move from one cycle or hour to ten cycles or hours, then a hundred cycles or hours. The left and right arrow keys can move forward or backward through the cycles or hours. To move forward or backward quickly, press SHIFT and the left or right arrow key. If you are in ten cycle mode it will 5. If you are in ten cycle mode, it will move the display by 30.
7. If you want to change the way the cycle time information is displayed, press F2, [PAGE], and select one of the following:

   - *Cycle graphics* which displays the cycle time information graphically on the *i*Pendant.
   - *Cycle text* which displays the raw cycle text information. The cycle information is stored in CYCLES.DG.
   - *Average text* which displays the average raw cycle text information. The average text information is stored in CYCAVE.DG.
   - *Line text* which displays the line by line raw text information. This information is stored in CYCLINE.DG.

   Note All of this information can be displayed from the web server.

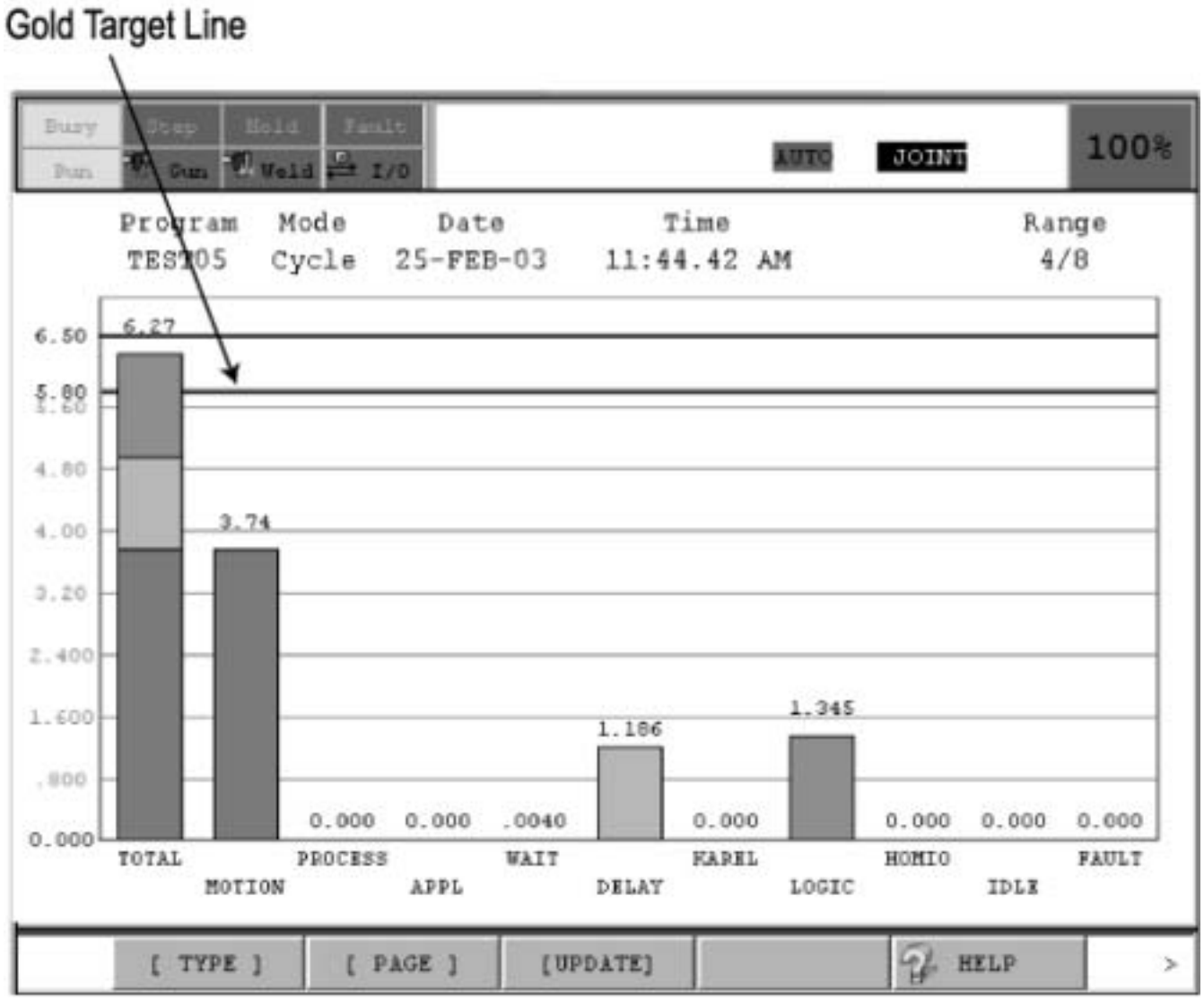
8. **To display cycle time information for a specific program category**, press F4, [CATGRY], and then select the category of information you want displayed.
9. **To change whether the cycle time information is displayed graphically or as raw text**, press >, NEXT, press F2, [PAGE], and then select the kind of display you want.
10. **To clear the current cycle or hourly data**, press >, NEXT, and then press F3, [UPDATE]. Select Clear Cycle Data which clears all cycle information.

Note You might want to clear the current cycle or hourly data if you are done tracking it and you want to start over, or if you want to load historical data and do not want to compare it with what currently exists on the controller.

11. **To load preexisting cycle data**, press >, NEXT, press F3, [UPDATE], then select Load Cycle Data.
12. **To begin collecting hourly data**, press >, NEXT, press F3, [UPDATE], then select Collect Hourly Data.
13. **To enable line by line cycle time tracking**, you must set the following system variables: $CY_CONFIG.$LINEENABLE=TRUE and $NUMLINES=number of lines to track. After the system variables are set, you must update the line by line information.
14. If you want to set a target cycle time which will be displayed as a horizontal gold bar when one cycle is displayed,

    - Press SELECT and select the current program.
    - Press >, NEXT, and then press F2, DETAIL.
    - Press F3, NEXT until Target Cycle Time(seconds) is displayed.
    - Press ENTER and type a new target cycle time.
    - When you have finished, press F1, END, and then press ENTER.
    - Press MENUS, select STATUS, press F1, [TYPE] and select Cycle Time if it is not already displayed.
    - Press >, NEXT, and then press F2, [PROG] to select the current program.
    - Press F3, [ZOOM], and select One Cycle. You will see a screen similar to the following.

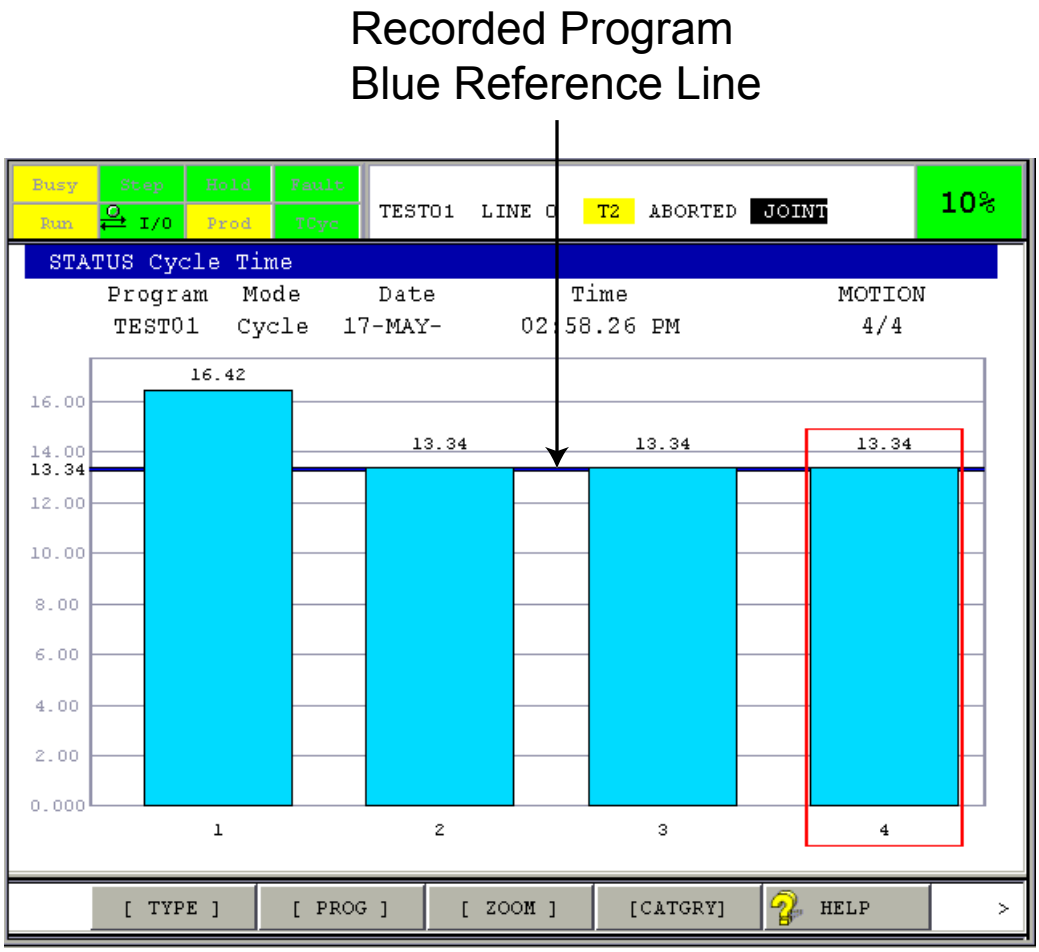
Figure 10–33. Target Cycle Time



15. **To record the current program as a reference program**, press F4, RECORD. This recorded program will be displayed as a blue line on the screen and can then be used as a reference. F4, RECORD will only be displayed during cycle mode when an individual category is selected. See Figure 10–34 .

Note If you have already recorded a reference program, you will be prompted to answer whether or not you want to overwrite the previous reference program.

Figure 10–34. Reference Program Recorded Display



16. **To save the cycle time information**, press FCTN, and select SAVE. This information can then be used if a system upgrade is performed. Cycle time information is saved to PGCYCCYC.TXT and Line information is saved to PGCYCHR.TXT

Note You can also use Backup under the FILE menu to save this information.

10.14 DATA MONITOR

10.14.1 Overview

Data Monitor is a tool for improving process quality. You can use it to monitor and record important process parameters. It can alert you to a parameter going out of limit and it can record data for use in a quality record.

Data Monitor operates much like a strip chart recorder or a data acquisition system. To use the Data Monitor feature you make selections in two or three teach pendant screens (depending what application you are using). Specifically,

- Enable or disable any specific features in the Data Monitor Utility screen
- Select items to monitor (such as arc current feedback or actual flow rate) with one of the Data Monitor schedules
- If you are using ArcTool or DispenseTool, add Sample Start[schedule number] and Sample End instructions to your TP program to control when monitoring occurs
- If you are using PaintTool, enable Automatic Sampling (in production mode) or Asynchronous Sampling in the Diagnostics setup screen.

  Note If you are using PaintTool, Data monitor also supports the addition of Sample Start [schedule number] and Sample End instructions in your teach pendant programs to control when monitoring occurs. It is recommended that you use this only for test purposes. You should remove the corresponding instructions from the teach pendant programs before putting your system into production mode.

You can monitor up to six items at once with a Data Monitor schedule. The maximum sampling frequency is 250 Hz. You can specify separate frequencies for limit checking and for recording. As the items specified in the schedule are recorded, the following data is also collected: time, date, distance, program name and line number. You can choose the items you want to monitor from the Data Monitor Schedule screen.

As data is recorded, it can be formatted as a report and sent to a file.

A short example report is shown in Report Example . The data is tab-delimited for importing into a spreadsheet application.

Report Example

```
DATA MONITOR REPORT
                      Number Tick Time  Program Line Voltage [Volts]
Wire feed[IPM]
                         1    48   .192      TEST 2           0.000
0.000
                         2    98   .392      TEST 3          20.000
200.000
                         3   148   .592      TEST 3          20.000
200.000
                         4   198   .792      TEST 3          20.000
200.000
                         5   248   .992      TEST 3          20.000
200.000
                         6   298  1.992      TEST 3          20.000
200.000
```

```
                          7   348  1.392      TEST 4           0.000
0.000
```

Definitions

This section contains definitions of terms you should know to use Data Monitor.

Item - A specific data element to be monitored. For example, an I/O signal, like WO[2] (for ArcTool) or DO[2] (for PaintTool or DispenseTool) or AI[2]. Data Monitor can monitor the following kinds of items:

- System variables (Real or Integer only)
- KAREL program variables (Real or Integer only)
- I/O ports (digital and analog)
- Registers (numeric only and ArcTool and DispenseTool specific)

Schedule -A set of parameters that define how to monitor specific items and where to save recorded data.

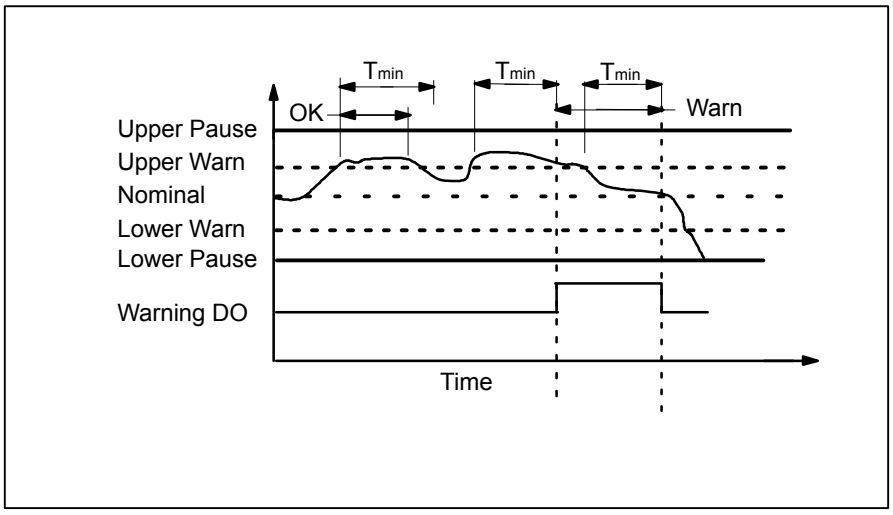
Trigger - A condition that must be met to begin or end monitoring.

Limit - A defined high or low value for a monitored item.

Monitoring Limits

The monitoring option can check each sampled data item against upper and lower boundary limits. If the sampled data value is past a limit for a specified time period an alarm occurs. You must specify two upper and two lower limits for each monitored data item. See Figure 10–35 .

Figure 10–35. Process Limits



One limit is of a WARN severity. If the first limit is crossed for a specified time a digital output is set to ON, but normal operation continues. If the monitored process signal returns to an acceptable value for a specified time or if sampling ends, the digital output is turned OFF. The second limit is more severe and has a PAUSE severity. If the second limit is crossed for the specified time, a pause error is posted immediately and the same digital output is set to ON. The digital output is set OFF if:

Note For PaintTool, it is recommended that you leave PAUSE limits DISABLED.

- The teach pendant program is resumed.
- The teach pendant program is aborted.
- The system is RESET.

Asynchronous Control

If you are using PaintTool, then you have the ability to control the Data Monitor sub-system *asynchronously*. Asynchronous monitoring means that a monitoring session can be started and stopped without modifying teach pendant programs with instructions to start and stop.

Note If you are using PaintTool, while you are using Data Monitor in asynchronous mode, **do not** enable the "program name" and "line number" data monitor setup items.

10.14.2 Data Monitor Setup

You must set up Data Monitor before you can use it. Table 10–23 lists and describes the items found on the Data Monitor SETUP screen.

Table 10–23. Data Monitor SETUP Screen Menu Items

ITEM	DESCRIPTION
Data Monitor Operation Default: Enabled	This item enables and disables operation of the Data Monitor function.
Recording Default: Enabled	This item turns data recording ON and OFF.
Filing Default: Enabled	This item turns report filing ON and OFF.
Pause on File Errors Default: Disabled	This item controls the severity of certain file errors. If this item is disabled, WARNING severity errors are posted and execution of the program being monitored will continue. If it is enabled, PAUSE severity errors are posted and execution of the program being monitored stops.

Table 10–23. Data Monitor SETUP Screen Menu Items (Cont'd)

ITEM	DESCRIPTION
Warning Limits Default:Enabled - ArcTool and DispenseTool only Default:Disabled - PaintTool only	This item turns Warning Limits ON and OFF.
Pause Limits Default: Enabled - ArcTool and DispenseTool only Default: Disabled - PaintTool only	This item turns Pause Limits ON and OFF. This option should always be disabled for PaintTool.
Limit Error Output Default: DO[0]	This item defines the port type and port number for the limit output. This digital output is turned ON when a limit error is detected.
Sample Buffer Size (for ArcTool and DispenseTool only) Default:10 Min:1 Max:20 **Sample Buffer Size (for PaintTool only)** Default:20 Min:1 Max:20	This item specifies the size of the sample buffer. If you are sampling at high speed (>125 Hz) you should increase the buffer size to 20. A power cycle is necessary for this change to take effect.
Record Buffer Size (for ArcTool and DispenseTool only) Default:10 Min:1 Max:99 **Record Buffer Size (for PaintTool only)** Default:99 Min:1 Max:99	This item specifies the size of the record buffer. If you are recording high data rate information or more than two channels you should increase the record buffer size to 99. A power cycle is necessary for this change to take effect.
Setup Default: Disabled	This item enables or disables printing of data monitor setup information in the report header. See Figure 10–36 .
Items Default: Disabled	This item enables or disables printing of information in the report header about each of the items you want to monitor.
Schedule Default: Disabled	This item enables or disables printing of Schedule information in the report header.

Table 10–23. Data Monitor SETUP Screen Menu Items (Cont'd)

ITEM	DESCRIPTION
Triggers Default: Disabled	This item enables or disables printing of Trigger information in the report header.
Program Name - PaintTool only Default: Disabled	This item enables or disables printing the program name column in the Data Monitor report. See Figure 10–36 . Disable this when you start a session in Asynchronous mode or Automatic Sampling mode.
Line Number - PaintTool only Default: Disabled	This item enables or disables printing the line number column in the Data Monitor report. See Figure 10–36 . Disable this when you start a session in Asynchronous mode or Automatic Sampling mode.
Date Default: Disabled	This item enables or disables printing the date and time of day column in the Data Monitor report. See Figure 10–36 .
Tick + time Default: Disabled - ArcTool and DispenseTool only Default: Enabled - PaintTool only	This item enables or disables printing the Tick and Time column in the Data Monitor report. See Figure 10–36 .
Event Default: Disabled	This item enables or disables printing the event column in the Data Monitor report. See Figure 10–36 .
Distance Default: Enabled - ArcTool and DispenseTool only Default: Disabled - PaintTool only	This item enables or disables printing the distance column in the Data Monitor report. See Figure 10–36 . **Note** : If you are using PaintTool, this item is not applicable when your system works in Asynchronous mode or Automatic Sampling mode.

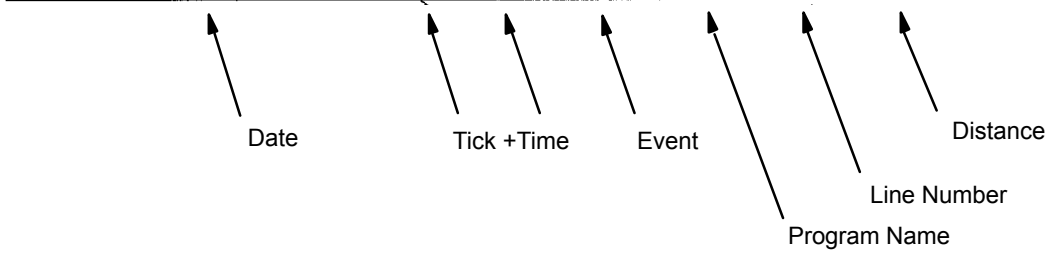
Table 10–24. Data Monitor ITM DETAIL Screen Items

ITEM	DESCRIPTION
Item type	This field indicates that the current item is one of the following: Integer, Real, I/O, Register.
Item sub type	This field is the sub type of the current item.
Port or register number	This field is the port or register number for the current item.
Program name	This field is the program name of the current item.
Var	This field is the system variable.
Des	This field is a description of the current item.
Units	This field indicates the units of measurement for the current item.
Slope	This field is the slope of the current item.
Intercept	This field is the intercept of the current item.

Use Procedure 10-24 to set up Data Monitor

Figure 10–36. Sample Report

Number	Date	Tick	Time	Event	Program	Line	Distance	Voltage (Command) [Volts]
1	18-AUG-98 10:41:46	0	0	1	TEST	2	0	0
2	18-AUG-98 10:41:46	62	0.248	1	TEST	2	0	0
3	18-AUG-98 10:41:46	124	0.496	1	TEST	2	0	0
4	18-AUG-98 10:41:46	186	0.744	1	TEST	2	0	0
5	18-AUG-98 10:41:46	248	0.992	1	TEST	3	0	0
6	18-AUG-98 10:41:46	310	1.24	1	TEST	3	0	0
7	18-AUG-98 10:41:46	372	1.488	1	TEST	3	0	0
8	18-AUG-98 10:41:48	434	1.736	1	TEST	3	0	0
9	18-AUG-98 10:41:48	496	1.984	1	TEST	3	0	0
10	18-AUG-98 10:41:48	558	2.232	1	TEST	3	0	0
11	18-AUG-98 10:41:50	620	2.48	1	TEST	3	0	0



Procedure 10-24 Setting Up Data Monitor

Conditions

- Data Monitor is installed on your controller.

Steps

1. Press MENUS.
2. Select UTILITIES.
3. Press F1, [TYPE].
4. Select Data Monitor. Depending on the application you are using, you will see a screen similar to the following.

```
UTILITIES DMON SET
1 Data Monitor Operation:        ENABLED
2 Recording:                     ENABLED
3 Filing:                        ENABLED
4 Pause on file errors:          DISABLED
5 Warning limits:                ENABLED
6 Pause limits                   ENABLED
7 Limit error output:            RO[  1]
8 Sample buffer size:            10 samples
9 Record buffer size:            10 samples
  ITEM DESCRIPTION                 ITEM 1
10 Voltage (Command)                    1
REPORT TABLE CONTENTS
11 Setup:                        ENABLED
12 Items:                        ENABLED
13 Schedule:                     ENABLED
14 Triggers                      ENABLED
REPORT TABLE CONTENTS
15 Pause limits:                 ENABLED
16 Line number:                  ENABLED
17 Date:                         DISABLED
18 Tick + time:                  ENABLED
19 Event:                        DISABLED
20 Distance                      ENABLED
```

5. Select each item on the menu and set it as desired.

6. **Edit Data Monitor Items.** Data Monitor allows you to define 20 items to monitor. These items are initialized for you, but you can edit them to suit your needs. If you want to edit Data Monitor items, move the cursor to item 10. When the cursor is on item 10, the DETAIL, [CHOICE], and HELP function keys will be available. [CHOICE] allows you to choose an item from a list. DETAIL allows you to edit that item. Item 10 has two columns. The right column contains an item number from 1 to 20. The left column contains the corresponding item description. Depending on the application you are using, you will see a screen similar to the following.

```
ITEM DESCRIPTION               ITEM NUM
10 Voltage (Command)                  1
```

7. Press F4, [CHOICE]. Depending on the application you are using, you will see the following screen.

```
UTILITIES DMON SET
\1\Voltage\(Command\5 Register 1
 2 Wire feed (Command 6 Arc detect status
 3 Voltage (Feedbac 7 Item 7
 4 Current (Feedbac 8 -- NEXT --
UTILITIES DMON SET
    ITEM DESCRIPTION              ITEM NUM
10 Voltage (Command)                     1
REPORT HEADER CONTENTS
```

8. Select an item from the list by number or by moving the cursor and pressing ENTER.

9. Press the F3, DETAIL function key to edit the selected item. In this example, 2 Wire feed has been selected. Depending on the application you are using, you will see a screen similar to the following.

```
UTILITIES DMON SET
    Item number:                 2/20
  1 Item type:                   REAL
    Item sub type:               **
    Port or register number:     **
  2 Program name: [      *SYSTEM*]
  3 Var: [      $awepor[1].$wfs_cmd]
  4 Des: [      Wire feed (Command)]
  5 Units               [       IPM]
  6 Slope:                   0.00
  7 Intercept:               0.00
```

10. This is the screen you use to edit an item. Press F2, ITEM, to select a different item by number. Not all of the menu items are available for all item types. If they are not available the item is not numbered, you cannot move the cursor to it, and it displays as ***. To set the Item type, move the cursor to line 1 and press F4, [CHOICE]. Depending on the application you are using, you will see a screen similar to the following.

```
1  Integer
2  Real
3  I/O
4  Register
UTILITIES DMON ITM
   1  Item type:               REAL
      Item sub type:              0
   2  Program name:  [      *SYSTEM*]
   3  Var    [ $awepor[1].$wfs_cmd]
```

11. To change the Item type, move the cursor to Item type, and press ENTER.

12. When you are finished editing this item you can press F2, ITEM, to select a different item by number, or press F3, EXIT, to return to the SETUP screen.

10.14.3 Data Monitor Schedule

You choose the items you want to monitor in a Data Monitor schedule.

You can also specify:

- Sampling frequencies
- WARN and STOP limits
- Report file naming details
- Start and Stop triggers

Reports

Reports are created automatically when you set the Reporting item on the Data Monitor schedule screen to ENABLED. You can specify a file name and a file device on the Data Monitor schedule screen for the Data Monitor report.

Table 10–25. Data Monitor Schedule Menu Items

ITEM	DESCRIPTION
Schedule Comment	You can add a comment to each Data Monitor schedule.
File device	This item allows you to specify the name of the device to be used when you write a report. You can choose from FLPY:, PRN:, FR:, MC:, CONS:, or RD:.

Table 10–25. Data Monitor Schedule Menu Items (Cont'd)

ITEM	DESCRIPTION
File name Default: Blank	This item allows you to specify the name of the file to be used for a report. A .DT file extension is always used. If you leave this item blank, and Reporting is enabled, the saved data file will be named "SAMPL". A directory name can be included with the file name. A file name of A\SAMPL results in one file named SAMPL.DT or multipleSAMPLxxx.DT files in the directory named A. Multiple data files are created if the file name index is greater than zero. If the file name includes a directory path on the file device, the directory must exist or an error will be posted and the file will not be saved.
File name index Default: 0 Min: 0 Max: 999	This item allows you to specify an index number to be appended to the file name when a report is generated. If this item is non-zero, each time a report is generated this index is incremented. For example if the file name is SAMPL then successive reports will be named SAMPL001.DT, SAMPL002.DT, and so forth. If the file name specified is more than 5 characters long it is truncated to 5 characters before the file index number is added. This limits the file name length to 8 characters total and permits numbering from SAMPL001.DT to SAMPL999.DT. When numbering reaches the limit, 999 in this case, the file index is reset to 1. If the file name is less than 5 characters long more file index numbers can be used. For example: SAMP0001.DT to SAMP9999.DT. Note the root directory of a file device may be limited to only 512 files. A subdirectory on a device may permit more files, possibly 65535. A directory name can be included with the file name. For example a file name of A\SAMPL will result in files named SAMPLxxx.DT in the directory named A.
File size - ArcTool and DispenseTool only Default: 0 Min:0 Max:99999	This item specifies the amount of memory in KB you expect to use on the file device. During execution of Sample Start[n], the device is checked for this amount of free memory. If it is not available, an error is posted. If you specify 0 or less than 5KB as the file size, the system will verify that there is at least 5KB of space available on the device you are saving data to.

Table 10–25. Data Monitor Schedule Menu Items (Cont'd)

ITEM	DESCRIPTION
File size - PaintTool only Default: 0 Min:0 Max:99999	This item specifies the amount of memory in KB you expect to use on the file device. During execution of Sample Start[n], the device is checked for this amount of free memory. If it is not available, an error is posted. If you specify 0 as the file size, the system only checks that there is at least one available block on the media. This should always be set to zero.
Sampling Default: 250 Min: 0 Max: 250	This item specifies the sampling frequency. • Request - This is the sampling frequency you specify. • Actual - This is the actual sampling frequency that Data Monitor will use. Since there are only certain frequencies available, it might be greater than or less than the sampling frequency you specify.
Monitoring Default: 10 Min: 0 Max: 250	This item specifies the monitoring frequency. • Request - This is the monitoring frequency you specify • Actual - This is the actual monitoring frequency that Data Monitor will use. Since there are only certain frequencies available, it might be greater than or less than the monitoring frequency you specify.
Recording Default: 1 Min: 0 Max: 250	This item specifies the recording frequency. • Request - This is the monitoring frequency you specify • Actual - This is the actual monitoring frequency that Data Monitor will use. Since there are only certain frequencies available, it might be greater than or less than the monitoring frequency you specify.

Table 10–25. Data Monitor Schedule Menu Items (Cont'd)

ITEM	DESCRIPTION
Record mode Default: CONTINUOUS	This item specifies the recording mode. • ONE BUFFER - Data will be recorded until the record data buffer is full • CONTINUOUS - The record buffer is re-used when full.
Number of items - ArcTool and DispenseTool only Default: 1 Min: 1 Max: 6 **PaintTool only** Default: Min: 1 Max:	This item specifies how many items are monitored by this particular schedule.

Use Procedure 10-25 to set up and edit a Data Monitor schedule

Procedure 10-25 Setting Up and Editing a Data Monitor Schedule

Conditions

- You have installed the Data Monitor option on your controller.

Steps

1. Press MENUS.
2. Select UTILITIES.
3. Press F1, [TYPE].
4. Select Data Monitor Schedules. Depending on the application you are using, you will see a screen similar to the following.

```
UTILITIES DMON SCH
 Sched: 1/5 [Weld cmd + fbk]
 1 Schedule: 1         [Weld\cmd\+\fbk\\]
 2 File device:                   [MC:]
 3 File name:               [           ]
 4 File name index:               1
 5 File size:                     0 KB
   FREQUENCY        REQUEST      ACTUAL
 6 Sampling:        250.00       250.00 Hz
 7 Monitoring:      125.00       125.00 Hz
 8 Recording:        10.00        10.00 Hz
 9 Record mode:                 CONTINUOUS
10 Number of items to monitor: 5
   ITEM DESCRIPTION            ITEM NUM
11 Voltage (Command)                   1
12 Wire Feed (Command)                 2
13 Current (Feedback)                  3
14 Voltage (Feedback)                  4
15 Fast Clock                          5
16 Start item:    2 > 22.5     ENABLED
17 Stop item:     3 < 200.0    ENABLED
```

5. There are five Data Monitor schedules. The top line of the Schedule screen displays the current schedule number and its comment. To select a different schedule, press F2, SCHEDULE, and enter the number of the schedule you want to modify.

6. To enter or modify the schedule comment, move the cursor to menu item 1 and press ENTER.

7. To specify the File device, move the cursor to line 2 and press F4, [CHOICE].

8. To specify the data monitor report file name, move the cursor to menu item 3 and press ENTER.

9. If you want to generate multiple report files with an index number in the file name, specify the starting index number on line 4. If you do not want to create a new indexed file each time this schedule is used, enter 0 on line 4.

10. Move the cursor to line 5, and specify the maximum report size, if desired, or leave it set to 0.

11. You specify the Sampling, Monitoring, and Recording frequencies on lines 6, 7, and 8 of the Data Monitor schedule screen. There are two values shown for these three items, the Requested Frequency that you specify, and the Actual Frequency that will be used as the Sampling, Monitoring, or Recording Frequency. When you enter the desired frequency in the Requested column, the Actual column will update with the closest available frequency.

Note The Monitoring and Recording Frequencies must be fractions of the Actual Sampling frequency. If you have an Actual Sampling frequency of 125 Hz, the maximum Monitoring and Recording frequency can only be 125 Hz. If you modify the Sampling frequency, the Actual frequency might change for all three frequencies.

12. To modify the Record mode, move the cursor to line 9 on the Data Monitor schedule screen and press F4, [CHOICE].

13. Each Data Monitor schedule can monitor up to five items simultaneously. You can specify the number of items to monitor on line 10 of the Data Monitor schedule screen.

14. Specify the items you want to monitor on lines 11 through 15 of the Data Monitor schedule screen. Press F4, [CHOICE], and select the item you want from the list of items displayed. You can also specify an item by number in the Item Num column.

15. **To specify WARN and STOP limits**, move the cursor to each item you have specified on lines 11 to 15 and press F3, LIMITS.

16. Move the cursor to one of monitored items.

17. Press F3, LIMITS. You will see a screen that contains details on how specifically to monitor the item you selected. Depending on the application you are using, you will see a screen similar to the following.

```
DATA MONITOR
TEST          LINE0
Schedule:   [Sample example        ]
 Item: 2
    Des:[Wire feed (Command) ]
    Var:[$awepor[1].$wfs.cmd ]
 1 Nominal value:         0.00 Volts
 2 Warning limit:         0.00 Volts
 3 Pause limit:           0.00 Volts
 4 Time before error:     0 seconds
```

18. Select each item and set it as desired.

19. **When you finished setting items**, press F3, EXIT, to display the previous screen.

20. You can specify the **Start and Stop trigger items** and conditions using the fields in menu items 16 and 17 on the Data Monitor schedule screen.

10.14.4 Data Monitor Programming

You can use the following teach pendant instructions to start and end data monitoring:

- Sample Start[]
- Sample End

Note If you are using PaintTool, do not use these instructions when you use the Automatic Sampling or Asynchronous Control features.

Sample Start[]Sample End

The Start instruction has a schedule number as an input parameter.

Note You cannot start multiple monitoring processes at one time. You must end a monitoring session with a Sample End before executing another Sample Start.

- To **start** a monitoring process, include the Sample Start[] instruction in a teach pendant program.
- To **stop** a monitoring process, include the Sample End instruction in a teach pendant program.

See Example of Using Sample Start[] and Sample End in a Teach Pendant Program for an example of how to use these instructions in a teach pendant program.

Example of Using Sample Start[] and Sample End in a Teach Pendant Program

```
1:  Sample Start[1]
2:J P[1] 40% FINE
 :  Arc Start[1]
3:L P[2] 20.0inch/min FINE
 :  Arc End[1]
4:  Sample End
[End]
```

The Sample Start and Sample End instructions are located in the Data Monitor category of the teach pendant editor INST menu.

10.15 DIAGNOSTIC VIDEO MONITORING

10.15.1 Overview

Diagnostic Video Monitoring (DVM) enables you to record pictures related to robot events using the robot's vision hardware. These pictures are accumulated into a video loop inside the controller. A copy of this video loop is logged to permanent memory when a significant event occurs. You can then view the DVM log at a later time to help understand the conditions leading up to the event.

DVM is configurable. You can control when pictures are taken, and can set the size of the video loop. You can also identify the event(s) that cause a video loop to be stored, and can enable/disable the feature as required.

Note DVM is available as a stand-alone option. It is compatible with other *i*RVision options on R-30iA robots. However, when loaded with any other *i*RVision option, the amount of TEMP memory available to DVM is significantly reduced.

Diagnostic Video Monitoring can be either *time based* or *event based*. If DVM is time based, you can set up a schedule to perform a snap on a fixed time interval. If DVM is event based, you can set up a schedule to perform a snap based on a specified event.

Refer to Table 10–26 for information on setting up Time Based schedules. Refer to Table 10–27 for information on setting up an event based schedule. Use Procedure 10-26 to set up a DVM schedule.

10.15.2 DVM Setup

Table 10–26. Setting Up DVM Time Based Schedules

ITEM	DESCRIPTION
TimeBased Schedules	Time based schedules perform a picture snap on a fixed time interval. They perform logging based on specified events.
Name	Each schedule is identified by a 1 to 12 character name. To modify the name, move the cursor to it, press ENTER, and type the name.
State	A schedule can be in one of four states – ENABLED, DISABLED, POWER UP or PENDING. Changing any setting on a schedule causes the state to become DISABLED. Press F4, ENABLE to enable the schedule or F5, DISABLE to disable it. A schedule with the state of ENABLED, POWER UP or PENDING will be enabled automatically on power-up. If the schedule cannot be automatically enabled at power-up, then an alarm will be posted and the state set to DISABLED. You must fix the problem and manually enable it again.
Camera port	Cameras are identified by port number. To change the assignment, cursor to this field and enter a new number between 1 and 8. When the cursor is on this field, a picture from the selected camera is periodically sent to the STATUS *i*RVision display. You can use this feature to aim and focus the camera. Put the *i*Pendant display into Double screen mode. Show the DVM schedule DETAIL screen on one pane, and the STATUS *i*RVision screen on the other.
Exposure Time	This value controls the exposure setting of the camera when DVM snaps a picture. When exposure time is changed a new picture is taken and sent to the STATUS *i*RVision display.
Picture Snap Control	When a time based schedule is enabled, DVM snaps pictures on a fixed time interval using the robot's vision hardware. A logging event also triggers a picture snap to ensure that you have a picture synchronized with the event that caused the log to be written. These pictures are accumulated into a video loop in the controller's TEMP memory. The size of the video loop is determined by the rate that the pictures are being snapped and the total amount of time the loop is required to cover. After enough pictures have been recorded to fill the video loop, the next snapped picture replaces the oldest one. The next three parameters control time based Picture Snap process.

Table 10–26. Setting Up DVM Time Based Schedules (Cont'd)

ITEM	DESCRIPTION
Interval	The rate that pictures are snapped is controlled by this setting. It is the time, in seconds, between picture snaps. The minimum time is 0.1 sec.
Loop Coverage Before	This value defines the requirement that the video loop be large enough to maintain pictures snapped this number of seconds before the log event.
Loop Coverage After	If this value is non-zero, pictures continue to be added to the video loop after the log event for this many seconds, then the video loop is logged. The size of the video loop is must be large enough to contain pictures being snapped at the interval rate before and after the log event.
Logging Control	Logging is the process of writing the video loop and associated data to non-volatile memory. It is triggered by any one of the events that you specify in the DVM schedule.
Log Path	The log path is the path to a folder in which DVM logs are to be stored. It must be on a non-volatile storage device. Supported devices include: • MC: • UD1: • FR: • Cx: (FTP Client) When logging is triggered, a new folder under Log Path is created and populated with a number of files. See Section 10.15.4.3 for details on content and format of the files written.
Events	You can specify from one to five independent events to trigger the log process. The events are numbered from 1 to 5. You must specify at least one event. Events are defined by type and number.

Table 10–26. Setting Up DVM Time Based Schedules (Cont'd)

ITEM	DESCRIPTION
Event Type	Pressing F4, [CHOICE] presents the following types of events: • **None** – Don't use this event • **XX On** – I/O point transition to On • **XX Off** – I/O point transition to Off • **FFFF Alarm** - Alarms with FFFF facility codes A selection for every type of discrete I/O point currently configured on your robot is represented in the [CHOICE] list. The XX shown above will be replaced by one the following: DI, DO, FL, RI, RO, SI, SO, UI, UO, WI, WO, WSI, WSO. The list contains a selection for every alarm facility that is possible on your controller.
Event Number	The meaning of event number is dependent on the event type. • When the event type is None, the event number is ignored. • For I/O event types, Number is the I/O signal number of the type specified. • For alarm types, Number is the specific alarm number. -1 indicates that any alarm of the specified facility code will trigger the event.

Table 10–27. Event Based Schedules

ITEM	DESCRIPTION
Event Based Schedules	Event based schedules perform a picture snap when specified events occur. They also perform logging based on specified events.
Name	Each schedule is identified by a 1 to 12 character name. To modify the name, move the cursor to it, press ENTER, and type the name.
State	A schedule can be in one of four states – ENABLED, DISABLED, POWER UP or PENDING. Changing any setting on a schedule causes the state to become DISABLED. Press F4, ENABLE to enable the schedule or F5, DISABLE to disable it. A schedule with the state of ENABLED, POWER UP or PENDING will be enabled automatically on power-up. If the schedule cannot be automatically enabled at power-up, then an alarm will be posted and the state set to DISABLED. You must fix the problem and manually enable it again.

Table 10–27. Event Based Schedules (Cont'd)

ITEM	DESCRIPTION
Camera port	Cameras are identified by port number. To change the assignment, cursor to this field and enter a new number between 1 and 8. When the cursor is on this field, a picture from the selected camera is periodically sent to the STATUS *i*RVision display. You can use this feature to aim and focus the camera. Put the *i*Pendant display into Double screen mode. Show the DVM schedule DETAIL screen on one pane and the STATUS *i*RVision screen on the other.
Exposure Time	This value controls the exposure setting of the camera when DVM snaps a picture. When exposure time is changed a new picture is taken and sent to the STATUS *i*RVision display.
Picture Snap Control	When an event based schedule is enabled, DVM snaps a picture each time a specified robot event occurs. A logging event (see below) also triggers a picture snap to ensure that you have a picture synchronized with the event that caused the log to be written. These pictures are accumulated into a video loop in the controller's TEMP memory.
Events	You can specify from one to five independent events for snapping a picture. The events are numbered from 1 to 5. Events are defined by type and number.
Event Type	Pressing F4 [CHOICE] presents the following types of events • **None** – Don't use this event • **XX On** – I/O point transition to On • **XX Off** – I/O point transition to Off • **FFFF Alarm** - Alarms with FFFF facility codes A selection for every type of discrete I/O point currently configured on your robot will be represented in the [CHOICE] list. The XX shown above will be replaced by one or more of the following: DI, DO, FL, RI, RO, SI, SO, UI, UO, WI, WO, WSI, WSO. The list will also contain a selection for every alarm facility that is possible on your controller.
Event Number	The meaning of event number is dependent on the event type. • When the event type is **None**, the event number is ignored. • For I/O event types, Number is the I/O signal number of the type specified. • For alarm types, Number is the specific alarm number. • -1 indicates that any alarm of the specified facility code will trigger the event.
Logging Control	Logging is the process of writing the video loop and associated data to non-volatile memory. It is triggered by any one of the events that you specify in the DVM schedule.

Table 10–27. Event Based Schedules (Cont'd)

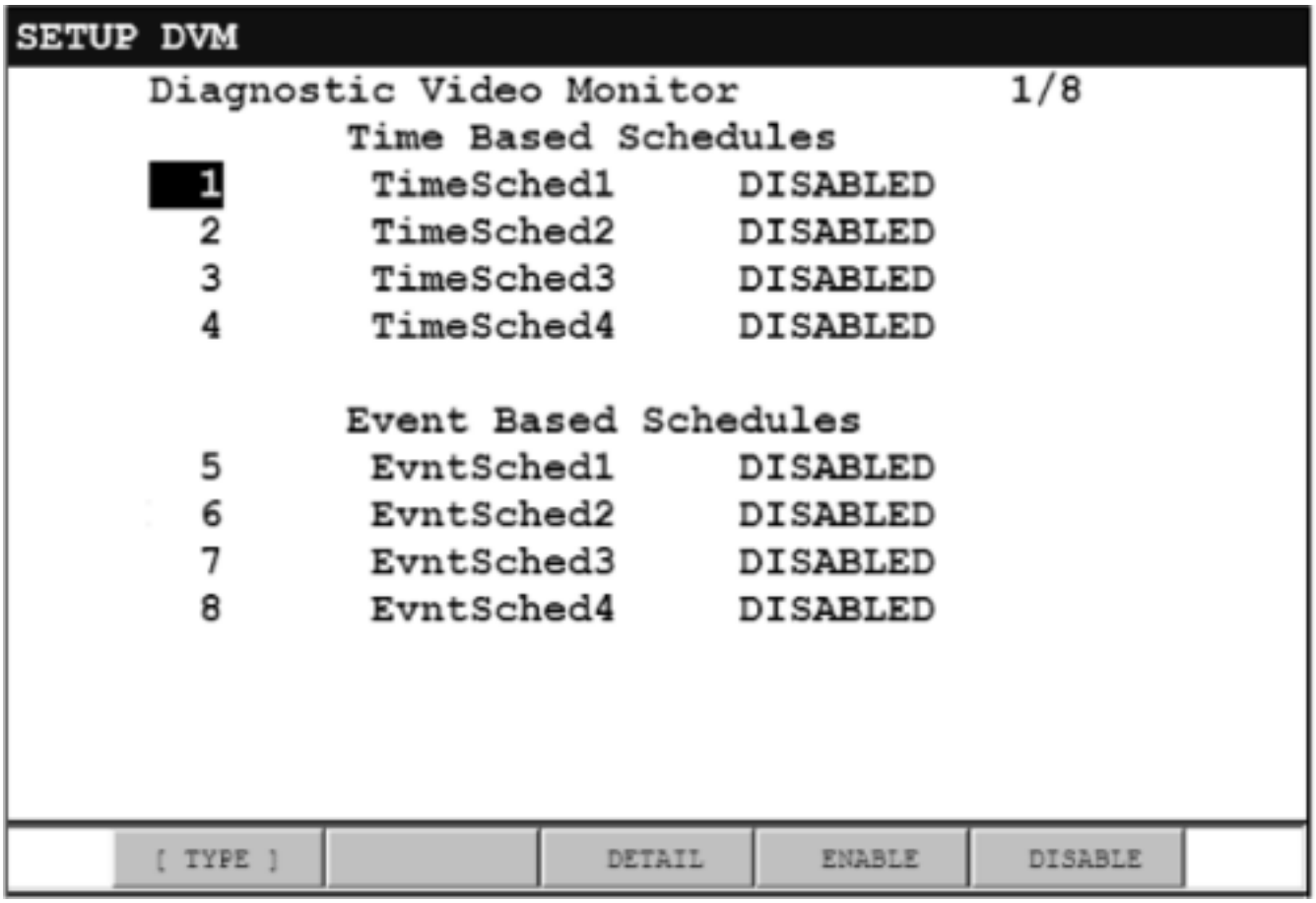
ITEM	DESCRIPTION
Number of Pictures to Log	This is a maximum number of pictures to log. If fewer than this number has been snapped, then only that number will be included in the log. After enough pictures has been recorded to fill the video loop, the next snapped picture replaces the oldest one. This value defines how much controller TEMP memory must be set aside for holding pictures. If the number is too large, the alarm "CVIS-295 There is not TEMP memory to run DVM at this time." will be displayed when you try to enable the schedule.
Log Path	The log path is the path to a folder in which DVM logs are to be stored. It must be on a non-volatile storage device. Supported devices include: • MC: • UD1: • FR: • Cx: (FTP Client) When logging is triggered, a new folder under Log Path is created and populated with a number of files. See Section 10.15.4.3 for details on content and format of the files written.
Events	You can specify from one to five independent events to trigger the log process. The events are numbered from 1 to 5. You must specify at least one event. Events are defined by type and number.
Event Type	Pressing F4, [CHOICE] presents the following types of events: • None – Don't use this event • XX On – I/O point transition to On • XX Off – I/O point transition to Off • FFFF Alarm - Alarms with FFFF facility codes A selection for every type of discrete I/O point currently configured on your robot represented in the [CHOICE] list. The XX shown above will be replaced by one the following: DI, DO, FL, RI, RO, SI, SO, UI, UO, WI, WO, WSI, WSO. The list contain a selection for every alarm facility that is possible on your controller.
Event Number	The meaning of event number is dependent on the event type. • When the event type is None, the event number is ignored. • For I/O event types, Number is the I/O signal number of the type specified. • For alarm types, Number is the specific alarm number. -1 indicates that any alarm of the specified facility code will trigger the event.

Procedure 10-26 Setting Up DVM Schedules

Refer to Table 10–26 . Refer to Table 10–27 .

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE], and select Diag Video Mon from the list.
4. You will see a screen similar to the following.

Figure 10–37. DVM Schedule List Screen



5. You can configure up to eight schedules. Select the schedule you want by highlighting the number in the left column.
6. If you want to edit a Time Based schedule, move the cursor to the schedule you want to edit and press F3, DETAIL. Set each item as desired.
7. If you want to edit an Event Based schedule, move the cursor to the schedule you want to edit and press F3, DETAIL. Set each item as desired.

8. Press F4, ENABLE to enable the schedule or F5, DISABLE to disable it. When F4 is pressed, the settings of the schedule are checked to ensure that they are valid for controlling DVM. If so, the state is set to ENABLED and monitoring is immediately commenced.

- When a schedule's state is ENABLED, the robot is currently using its settings to do diagnostic video monitoring. Only one schedule can be enabled at a time.
- DISABLED schedules are ignored and need not be properly configured.
- A state of POWER UP indicates that it will be ENABLED automatically the next time the controller is powered up.
- The PENDING state is rare. A schedule will be in the PENDING state for a short time if a currently enabled schedule takes some time to be disabled before the newly selected schedule can be enabled.

9. To display the main DVM screen again, press F3, LIST or press PREV.

10.15.3 Controlling DVM Execution

You control which DVM schedule is enabled using the DVM SETUP screens. Each displays the state of the schedule(s). The function keys F4 and F5 are labeled ENABLE and DISABLE respectively. Press F4 ENABLE to enable the schedule. If one is currently enabled when you press F4 on a different schedule, the current one is disabled before the new one is enabled. When F4 is pressed, the settings of the schedule are checked to ensure that they are valid for controlling DVM. If so, the state is set to ENABLED and monitoring is immediately commenced.

If parameter validation fails and the schedule cannot be enabled, the reason is displayed as an alarm and the schedule remains DISABLED. A typical case is that there is not enough TEMP memory to support the requested picture capture criteria. If TEMP memory is too fragmented to accommodate the requirements for the video loop, the state will be set to POWER UP. The schedule will be enabled when you cycle power on the controller. A schedule with the state of ENABLED, POWER UP or PENDING will be enabled automatically on power-up. If the schedule cannot be automatically enabled at power-up, then an alarm will be posted and the state will be set to DISABLED. You must fix the problem and manually enable it as described above.

DVM uses system variables to provide advanced configuration and operation control. The following system variables can be modified to provide some advanced user features that are not presented using the teach pendant SETUP screens.

- $DVM_RUN_TIM.$ENABLE_NEW is an INTEGER for enabling schedule. Setting it to a non-zero value causes the current schedule to be disabled (if any) and the schedule indicated by the number to be enabled. DVM will set this value back to zero when it enables the schedule. Time schedules are numbered 1 through 4. Event schedules are numbered -1 through -4.

- $DVM_RUN_TIM.$DISABLE_ALL is a BOOLEAN that when set to TRUE disables the current schedule (if any). DVM sets it back to FALSE when the schedule is disabled.

10.15.4 Displaying DVM and DVM Logs

10.15.4.1 Overview

The camera image can be viewed on the*i*Pendant during setup and run-time.

To view a DVM log which displays a recorded video loop, you must use a personal computer.

10.15.4.2 Displaying DVM

Each time DVM snaps a picture, the picture is displayed on the STATUS *i*RVision screen.

Procedure 10-27 Displaying DVM

1. Press MENUS.
2. Select STATUS.
3. Select Vision.

10.15.4.3 DVM Logs

A DVM log is a copy of the video loop that is being captured. The log is stored as a folder of picture files plus a data file. A new log folder is created under the log path that you specified when the logging event is triggered. It is populated with the pictures from the video loop and the data file.

The log folder is named MMMDD_HHMMSS where MMM = a 3 character abbreviation of the month, DD = day of the month (1-31), HH = hour (0–23), MM=minute (0–59), and SS = seconds (0–58) (2 second resolution).

Pictures are saved in the newly created DVM log folder in compressed Portable Network Graphics (.png) file format. The picture file names are pic_NNN.png where NNN = 001 to the number of pictures in the video loop. pic_001 is the oldest one.

One additional data file named dvm_log.dat is also written to the log folder. It is a text file using windows style line termination (cr-lf). dvm_log.dat has the following format:

Table 10–28. Log File Data

Line #	Content	Description
1	VersionID	This is a version number that identifies the format of this file. This will allow file format changes to support future enhancements. VersionID = 1.0 for the format described here.
2	SchedName	This is the name of the schedule.
3	Date, Time, TimeStamp, LogEvent	This is information about the event that triggered the log. • Date – formatted as YYYY/MO/DD where: YYYY = year MO = month (01-12) DD = day (01-31) • Time – formatted as HH:MM:SS where: HH = hour (00-24) MM = minute (00-59) SS = second (00-58) • TimeStamp - the value of $FAST_CLOCK when the snap was triggered. • LogEvent - The event that triggered the log to be stored. This is a string. If it is an I/O event the string will have the form “DI[9] On”. Alarm events will contain the text of the alarm.
4	NumPic	This is the number of pictures saved in this log
5 through NumPic + 4	FileName, Date, Time, TimeStamp, SnapEvent	This is information about each picture. • FileName - “pic_NNN.png” where NNN= the picture number. • Date, Time and TimeStamp - formatted as above. • SnapEvent - the event that triggered the picture to be snapped. It is a string formatted like LogEvent above. If this picture was snapped using rate criteria, the format is “Interval 0.5 sec.”

10.15.4.4 Log Viewer

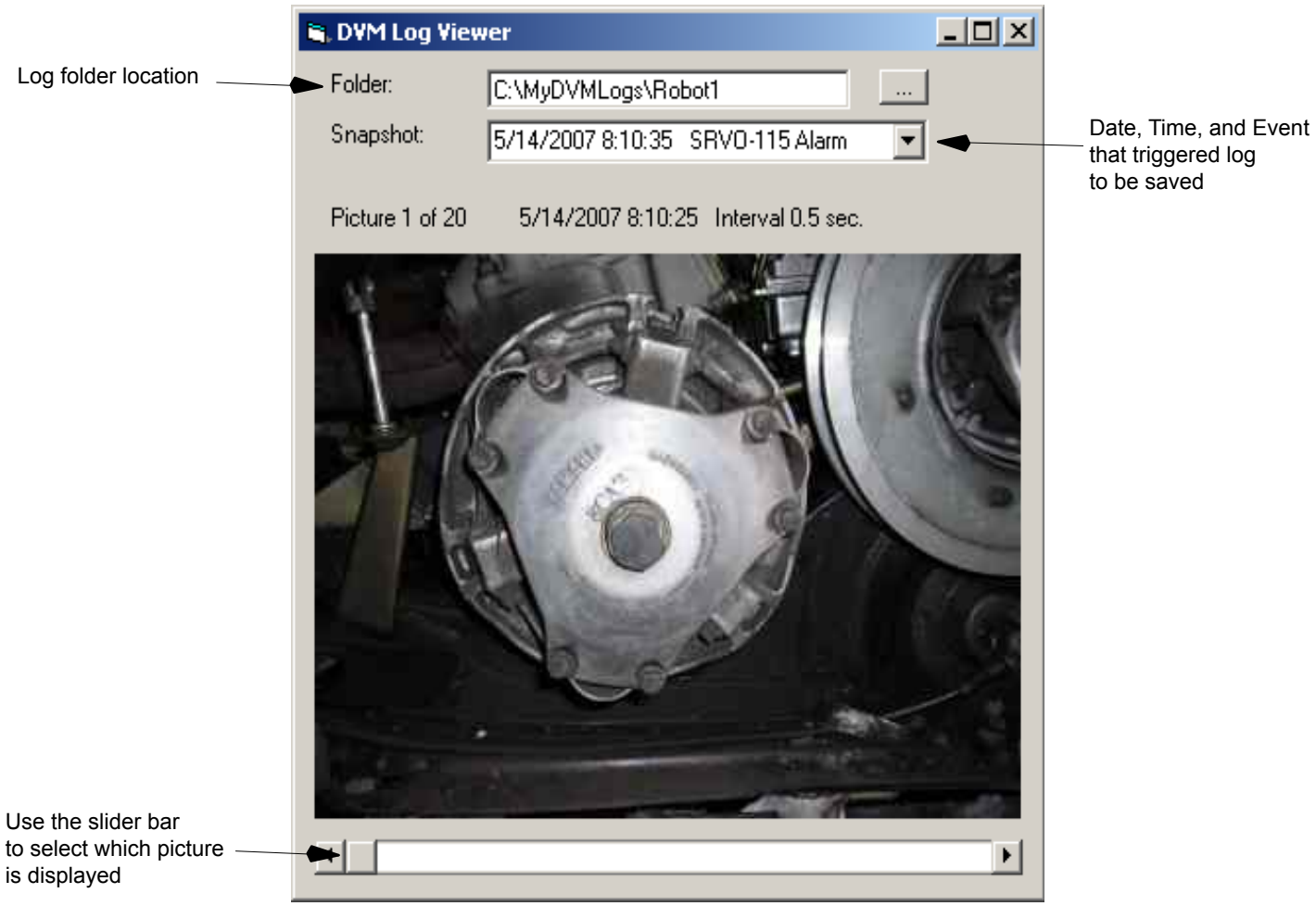
You can use standard PC tools to view the files in the DVM log folder. However, FANUC supplies a DVM Log Viewer that greatly improves your ability to analyze the log.

You can download the FANUC Robotics Diagnostic Video Monitor Log Viewer at www.fanucrobotics.com. You need to register a username and password in order to download

software. After you have registered, you can download the software by going to Support/Downloads, and select Software downloads. Then select Download PC Software, and choose DVM Viewer.

See Figure 10–38 for an example DVM Log Viewer.

Figure 10–38. Log Viewer



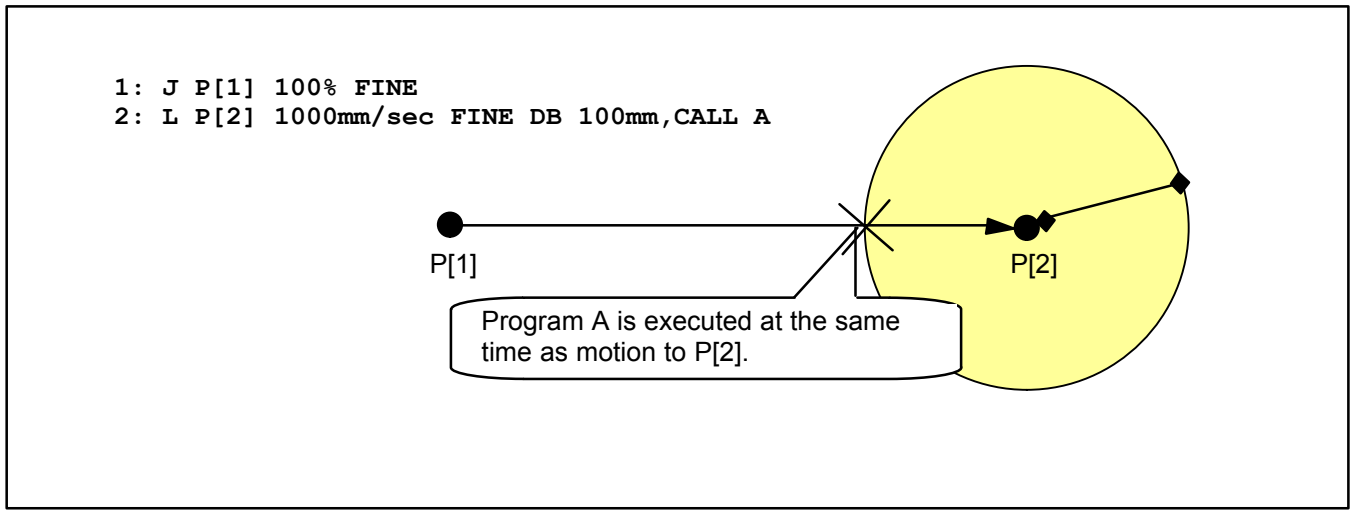
The DVM Log Viewer searches for logs in the selected folder and displays them in a drop-down list ordered chronologically. See Figure 10–38 . The logs are identified by date, time and event that triggered the log to be saved. Pictures in the log are displayed in the viewer below the status information. The picture number is shown along with the date, time and event that triggered the capture. If the pictures were being captured at a specified interval (rate), then the period is displayed. Use the slider bar at the bottom to select which picture is displayed.

10.16 DISTANCE BEFORE

10.16.1 Overview

The Distance Before (DB) function calls a program or outputs a signal when the Tool Center Point (TCP) goes into a region that is within a specified distance from the destination point. The program call and signal output are done in parallel with the motion execution of the main program. See Figure 10–39 for an example the execution timing of Distance Before.

Figure 10–39. Execution Timing of Distance Before



Note Before you can use Distance Before, the following system variable must be set:

$SCR_GRP[1].$M_POS_ENB = TRUE

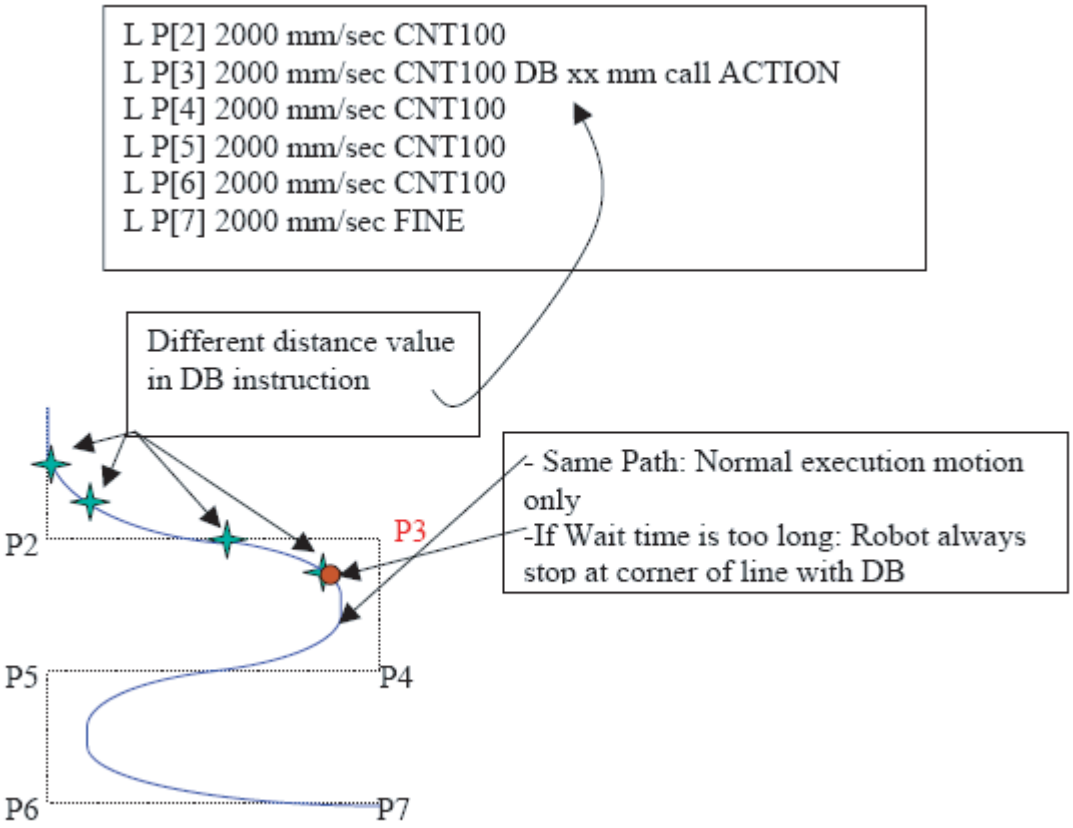
10.16.2 Specification

Refer to Table 10–29 for the specifications of the Distance Before function.

With the DB instruction, the robot TCP path varies with the value of $distbf_ver (see system variables section for details).

When $distbf_ver = 4, the system tries to maintain constant path. That is, the robot TCP path can be maintained the same regardless of the distance value in the DB instruction, regardless of the time in the WAIT statement in DB action, and regardless of the speed override, as shown in the following figure:

Figure 10–40. Normal Execution when $distbf_ver = 4



Warning

During Hold, E-Stop, and Resume, the robot TCP will not maintain the same path, no matter what the value of $distbf_ver!

Table 10–29. Distance Before Specification

Item	Specification	Limitation
Distance value	0 - 999.9 mm	The distance value and the actual execution timing are different. The difference depends on the speed of the TCP.

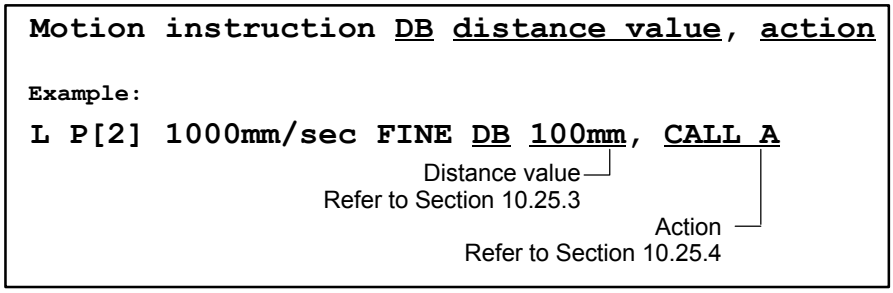
Table 10–29. Distance Before Specification (Cont'd)

Item	Specification	Limitation
Trigger condition	The TCP goes into a region that is within a specified distance from the destination point. Refer to Section 10.16.4 .	
Available instructions	• Signal output (such as DO[1] = ON) • CALL program	The program to be called cannot use a motion group. Only logic instructions are available.

10.16.3 Program Instruction

Distance Before is a motion option called DB. You can use Distance Before only as a motion option within a motion instruction. See Figure 10–41 for the motion instruction that includes the Distance Before motion option.

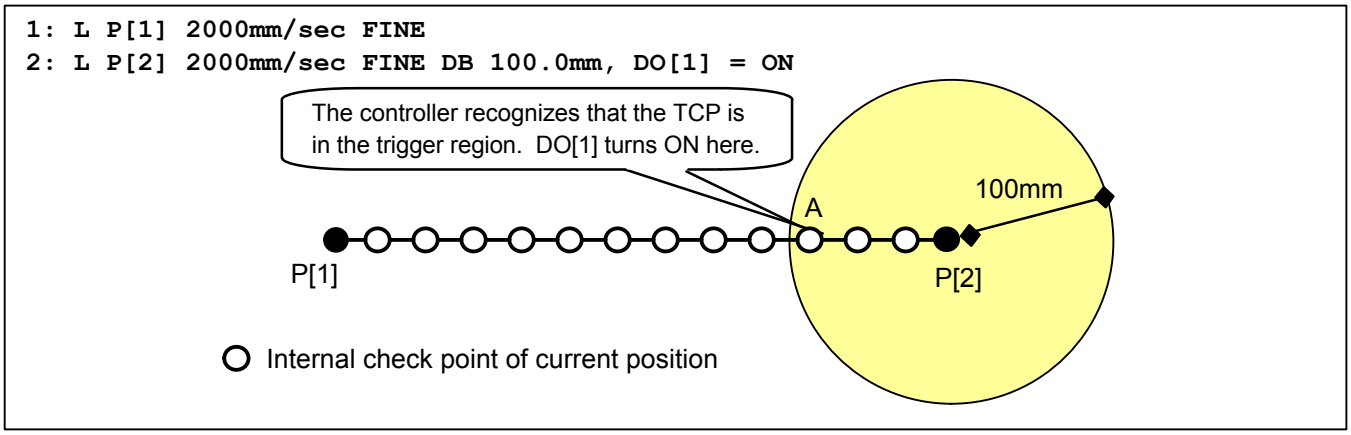
Figure 10–41. Distance Before Motion Option, DB



10.16.4 Distance Value

Distance Before executes an action when the TCP goes into a spherical region whose center is the destination position. The distance value specifies the radius of this sphere. The distance value is defined in millimeters, with a range of 0 to 999.9 millimeters. The spherical region whose radius is the distance value is called the *trigger region*. See Figure 10–42 .

Figure 10–42. Checking Whether the TCP Goes into the Trigger Region



Internally, the controller calculates the current position to judge whether the TCP is in the trigger region. The action is executed when this calculated position is in the trigger region.

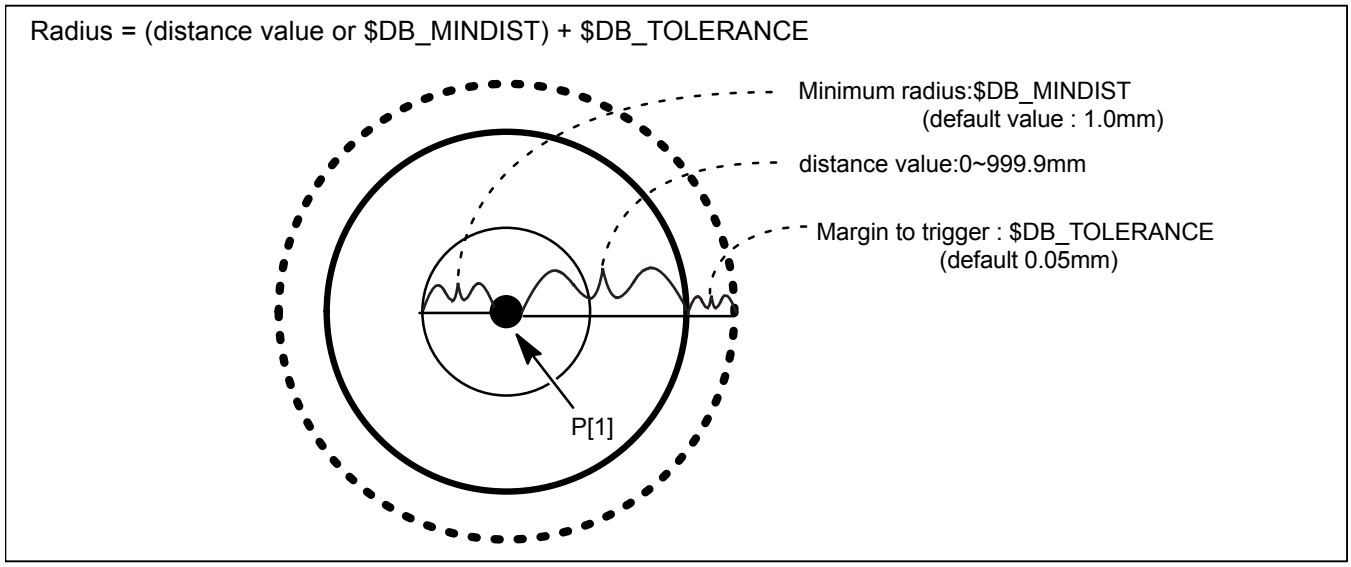
Caution

Execution timing of the action is determined by distance (in millimeters). Because the judgement to trigger is done cyclically by calculating the distance between the current position and destination position, the actual execution timing can be different than the distance value.

Radius of Trigger Region

The radius of the trigger region is defined as shown in Figure 10–43 .

Figure 10–43. Size of Trigger Region



If the distance value is less than $DB_MINDIST, $DB_MINDIST is used as the distance value. For example, suppose the following motion instruction is taught with $DB_MINDIST = 1.0:

```
L P[1] 2000mm/sec FINE DB 0.0mm DO[1]=ON
```

In this case, the controller interprets it as DB 1.0 mm. Then, $DB_TOLERANCE (which has a default value of 0.05 mm) is added to determine the radius of the trigger region. In this case, the trigger region radius of the trigger region is 1.0 + 0.05 or 1.05mm when default system variable values are used.

10.16.5 Action

The action specifies what is done when the TCP goes into the trigger region. Distance Before supports the following program instructions for the action:

- CALL program instruction
- Signal output instructions

Distance Before CALL Program

When a CALL program instruction is the action, the specified program is executed when the condition is triggered. The program to be called cannot use motion groups. You must change the group mask to [*,*,*,*,*] in the program header information screen. You can use parameters when you call a program, as follows:

```
L P[2] 1000mm/sec FINE DB 100mm, CALL A
```

```
(1,2)
```

Distance Before Signal Output

You can use the signal output instructions listed in Table 10–30 as Distance Before actions. You can use only one signal output for each Distance Before motion option.

Table 10–30. Distance Before Signal Output Instructions

Signal Instruction	Value
DO[ ] RO[ ]	ON OFF R[ ] pulse
GO[ ] AO[ ]	Constant R[ ] AR[ ]

You can also output a signal by calling a program that uses signal output instructions. However, to output only one signal with one Distance Before motion option, it is better to use the direct signal output method for ease of teach pendant program maintenance.

10.16.6 Changing the Trigger Condition

The action of a Distance Before motion option is executed when the controller recognizes that the TCP is in the trigger region. However, in some cases, the controller does not recognize that the TCP is in the trigger region. In these cases, the condition for the action to be executed, called the *DB condition*, is changed by the system variable $DB_CONDTYP, as shown in Table 10–31 .

Table 10–31. $DB_CONDTYP

$DB_CONDTYP	DB Condition	When Alarm is Posted
0	TCP is in the trigger region. This is called a *region trigger*.	*going away* or *penetration*
1 (default value)	*region trigger* + *going away*	*penetration*
2	*region trigger* + *penetration*	*going away*

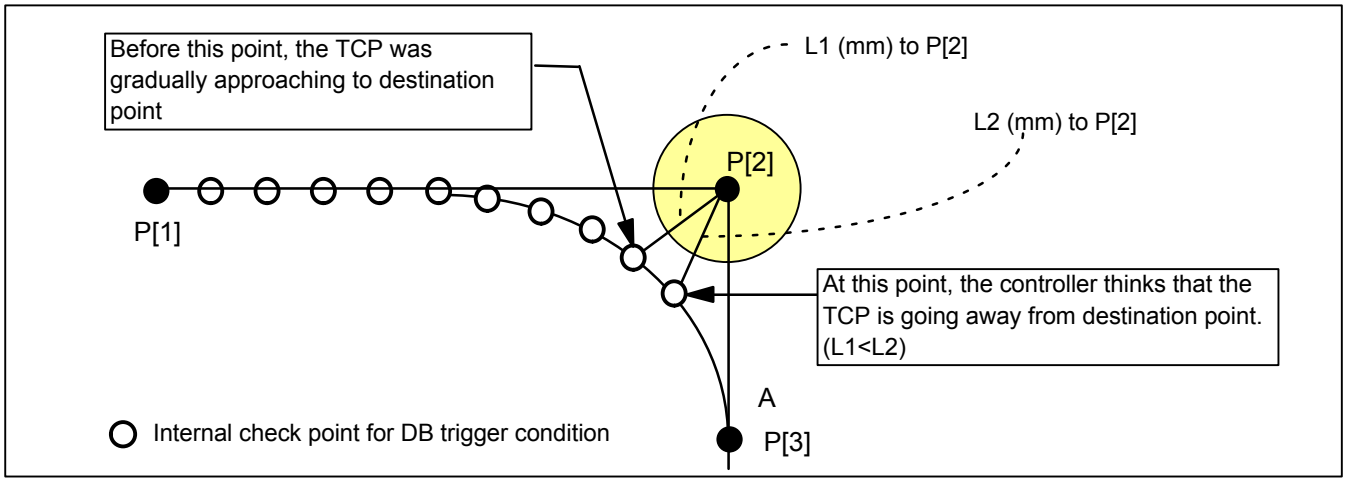
The *going away* and *penetration* cases are described below.

Distance Before executes the action when the DB condition is satisfied. Otherwise, an alarm is posted. There are two alarms for not-triggered Distance Before: INTP-293 and INTP-295. $DBCONDTRIG decides which alarm is posted. Refer to Section 10.16.7 for more information.

Going Away Case

If the termination type is CNT and the distance value is small, the TCP might not go into the trigger region. See Figure 10–44 .

Figure 10–44. TCP Does not Go into Trigger Region

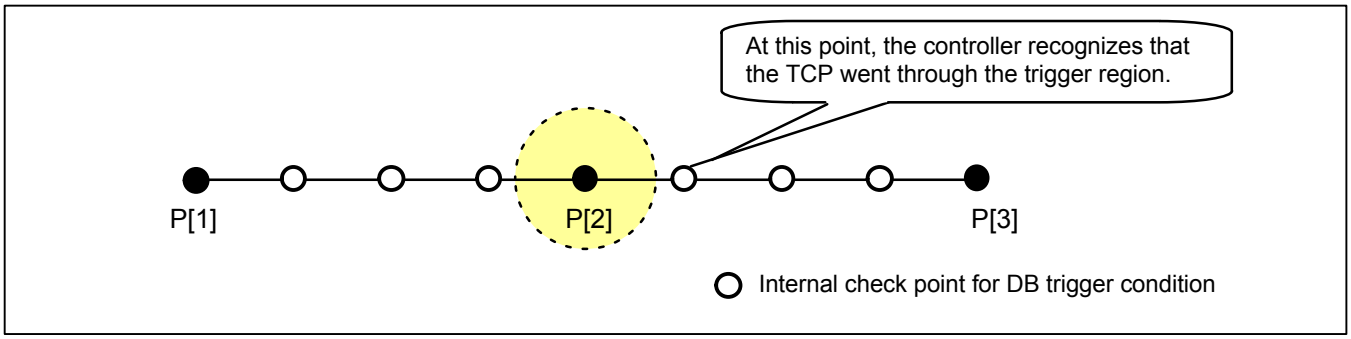


In Figure 10–44 , the TCP does not go into the trigger region. The TCP starts to go away from the destination point (P[2]). The controller cyclically judges if the TCP is going away from the destination point in addition to the *region trigger*. The controller recognizes that the TCP is going away when the calculated distance between the current position and destination position is greater than the previous one by more than $DB_AWAY_TRIG millimeters. This case is referred to as *going away*.

Penetration Case

This function cyclically checks whether the DB condition is triggered. Because of this cyclical check, CNT motion with high speed might cause the controller to omit cyclical checking in a small trigger region. See Figure 10–45 .

Figure 10–45. Penetration



In this case, the TCP moves too fast for the controller to check the DB condition in a small trigger region. Because cyclical checking is done outside of the trigger region, the fact that the TCP is in the trigger region is not recognized by the controller. This case is referred to as *penetration*.

To handle cases like the one shown in Figure 10–45 , Distance Before checks whether the TCP went through the trigger region. If the trajectory of the TCP penetrated the trigger region (penetration), the action is executed by the default configuration. However, in this case, the execution of the action is done after the TCP passed the destination point.

Motion with FINE termination type does not cause triggering by penetration. If you want to execute the DB action when the case is going away, set $DB_CONDTYP to 2 (the default value is 1).

10.16.7 Alarms Posted When Distance Before is Not Triggered

Distance Before posts an alarm if the condition is not triggered. The alarm posted depends on the value of $DBCONDTRIG, as shown in Table 10–32 .

Table 10–32. Distance Before Alarms

$DBCONDTRIG	Alarm to be Posted
0 (default value)	INTP-295 WARN (Program name, line number)DB condition was not triggered.
1	INTP-293 PAUSE.L (Program name, line number)DB condition was not triggered.

By default, INTP-295 is posted. Because the severity of this alarm is WARN, program execution does not stop. If you want to hold the program when a condition is not triggered, set $DBCONDTRIG to 1. INTP-293 will be posted when the condition is not triggered. The program will be held and the robot will decelerate to a stop.

10.16.8 Single Step Execution

If the Distance Before CALL program is executed using single step, the program is held at the time the action program is called. The rest of the motion instruction is executed at the next step execution that executes the action program step-by-step.

Single step execution of a motion instruction with a DB signal output action is the same as a motion instruction without DB, except the signal output is executed.

10.16.9 Hold and Resume

When a motion instruction with Distance Before is held and resumed, the execution timing of the next line and action is controlled by $DISBF_TTS. Refer to Table 10–33 for more details.

- **Execution timing of the next line** - In the default configuration, the execution of the next line does not wait until the rest of the motion instruction with the DB option completes.
- **Execution timing of the action** - Execution timing depends on the action, as follows:
  - — A signal output action is executed when the DB condition is triggered.
  - — The execution timing of a CALL program action is decided by DISTBF_TTS. When DISTBF_TTS is set to its default value (1), execution of the action program does not wait for the completion of the motion instruction.

Table 10–33. DISTBF_TTS for a CALL Program Action

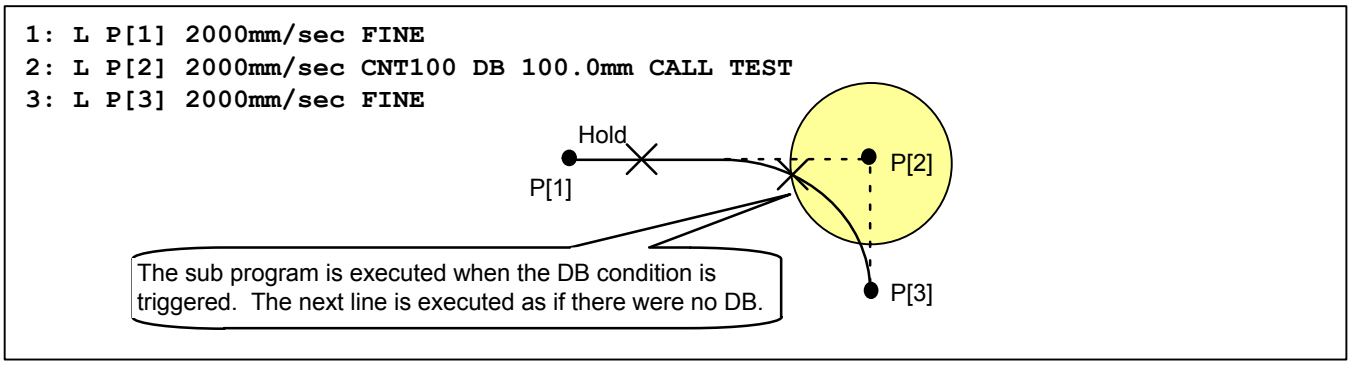
DISTBF_TTS	Execution Timing of Action Program	Execution Timing ofNext Line
0 (default value)	When the DB condition is triggered (Figure 10–46)	Normal
1	After the completion of the held motion instruction (Figure 10–47)	After the completion of the held motion instruction
2	When the hold is during the execution of the action program, the execution is the same a DISTBF_TTS = 1. Otherwise, the execution is the same as DISTBF_TTS = 0. (Figure 10–50)	The same as the action program execution timing

DISTBF_TTS = 0

When DISTBF_TTS = 0, the action program is executed just after it is called. The execution of line 3 does not wait for the completion of the motion in line 2.

Figure 10–46 shows an example of execution when **DISTBF_TTS = 0** .

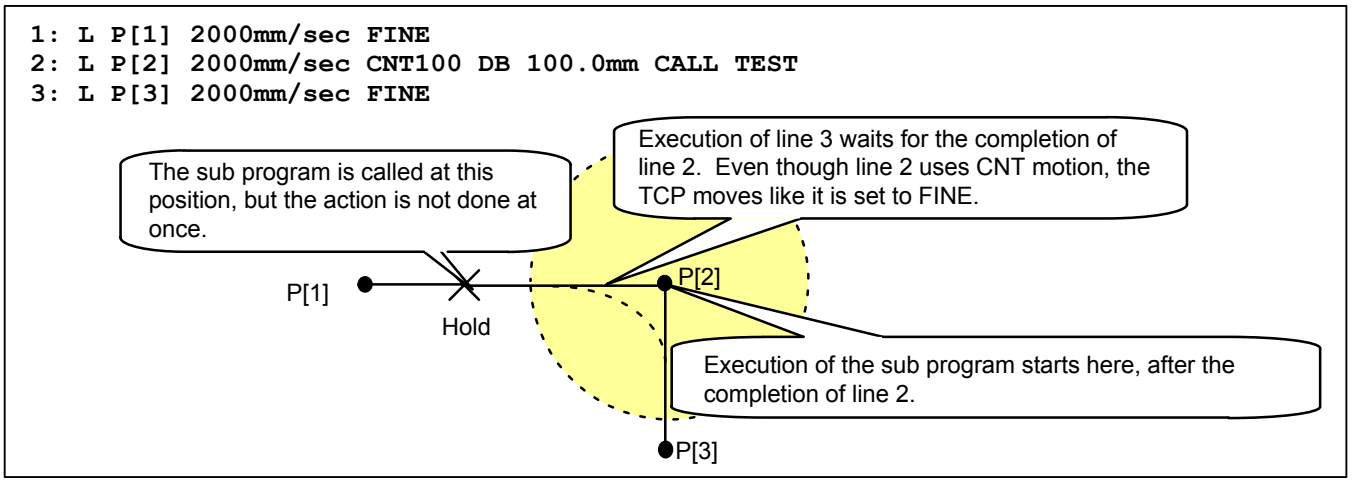
Figure 10–46. DISTBF_TTS = 0



DISTBF_TTS = 1

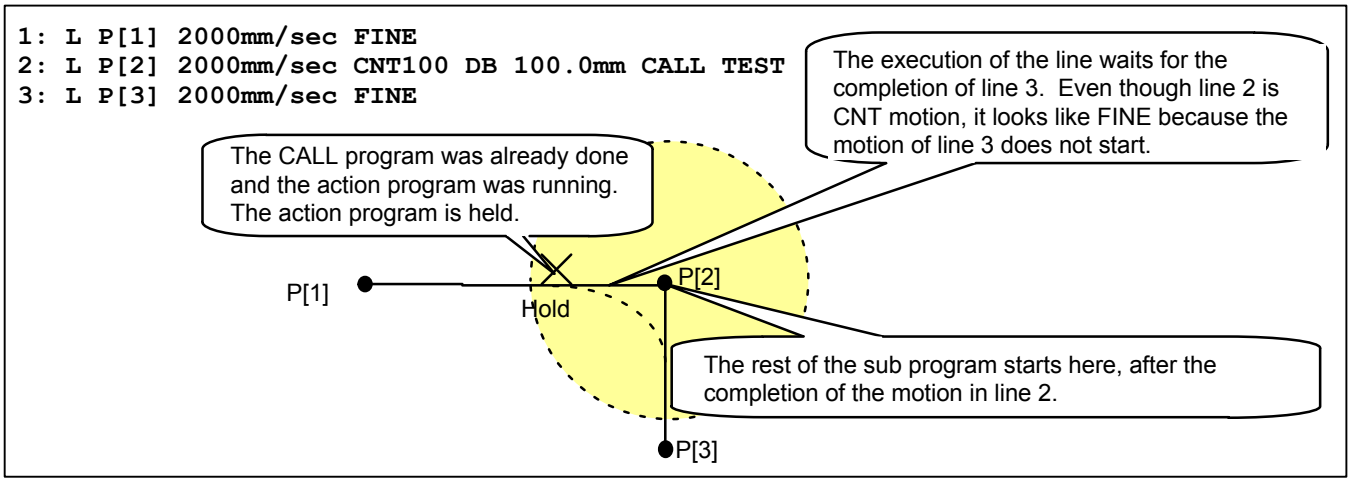
Hold before DB condition is triggered - The execution of an action program is done as described in Figure 10–47 . The CALL program is executed when the condition is triggered. However, the action program does not progress until P[2], where the motion of line 2 completes. Because the execution of line 3 also waits for the completion of line 2, the motion of line 2 looks like motion with a FINE termination type.

Figure 10–47. Execution Timing of an Action Program



Hold during the execution of an action program - When a hold occurs during the execution of an action program, the action program is held as well as the motion itself. After resuming, the execution of the rest of the action program does not start until line 2 completes. See Figure 10–48 .

Figure 10–48. Hold During the Execution of an Action Program

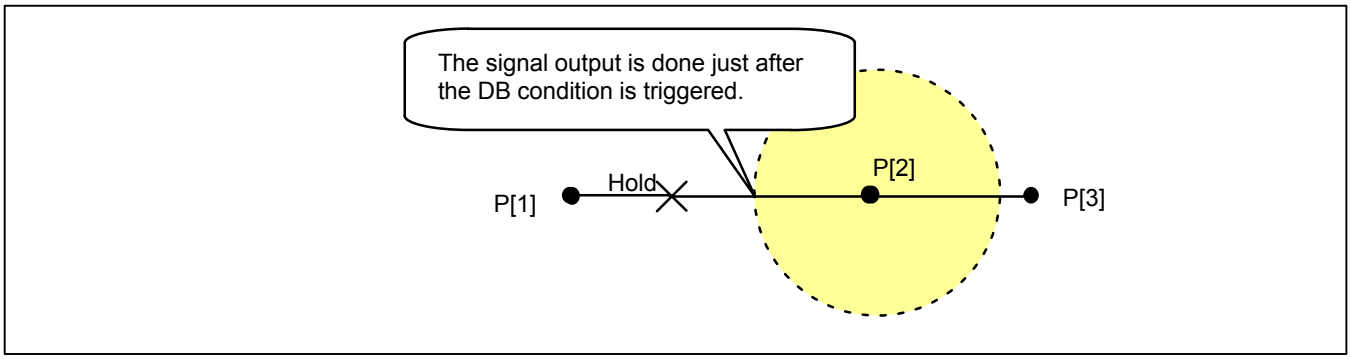


Holding after the completion of an action program — makes the next instruction wait for the completion of line 2.

Hold after the completion of line 2 but before the trigger of DB condition — In this case, if the DB condition triggers after resuming, execution of the action program is done just after the DB condition is triggered.

For a signal output action - a hold does not affect execution timing. However, the execution of the next line waits for the completion of the current line (line 2), as it does in the case of a CALL program action. See the example in Figure 10–49 .

Figure 10–49. Signal Output Timing After Resume

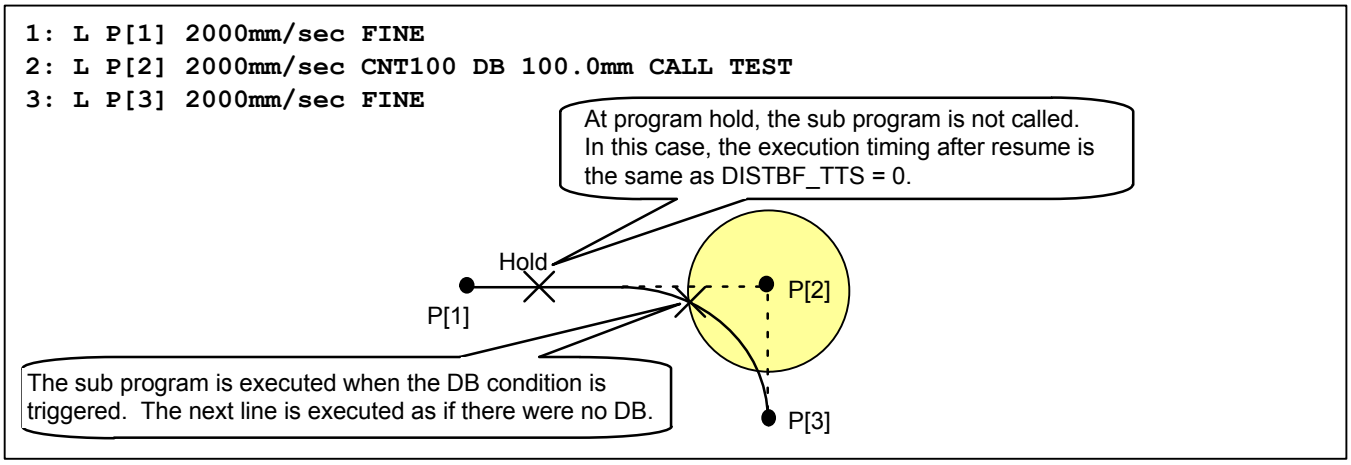


DISTBF_TTS = 2.

In Figure 10–50 , the TCP is not in the trigger region when the program is held, so the action program is not called yet. In this case, the execution timing of the action program and the next line is the same as DISTBF_TTS = 0.

If the program is held during the execution of the action program, the execution timing of the action program and the next line is the same as when DISTBF_TTS = 1.

Figure 10–50. DISTBF_TTS = 2



10.16.10 Resume After Jogging

If you hold a motion instruction that has the DB motion option, jog the robot, and then resume the program, the execution timing depends on the TCP position at the instant the program is resumed. If the TCP is in the trigger region when you resume the program, the action is executed just after resuming. Otherwise, the action is executed when the DB condition is triggered by motion after the program resumes.

Suppose that the following program is executed and held on line 2. In this case the DB condition was not triggered.

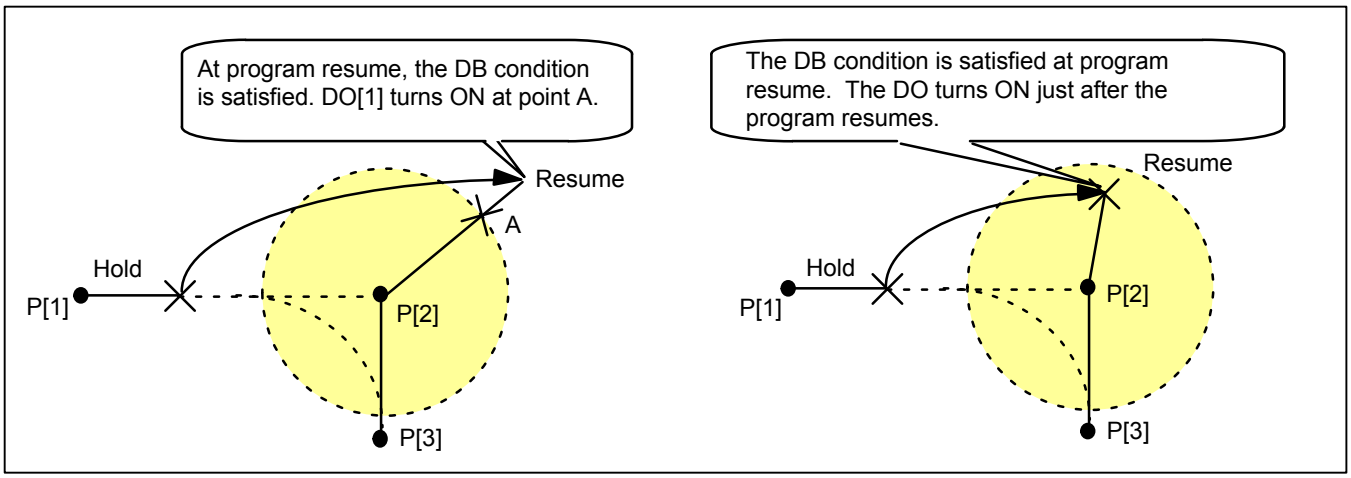
```
1: L P[1] 2000mm/sec FINE

2: L P[2] 2000mm/sec CNT100 DB 100.0mm DO[1]=ON

3: L P[3] 2000mm/sec FINE
```

If the TCP is distant from P[2] enough not to trigger (more than 100mm away), DO[1] turns ON when the DB condition is triggered by motion after resuming, at point A in the left diagram in Figure 10–51 . If the TCP is in the trigger region when you resume the program, DO[1] turns ON just after resuming, as shown in the right diagram in Figure 10–51 .

Figure 10–51. Resume After Jogging



10.16.11 Power Failure Recovery

If power is turned off during action program execution and power failure recovery is enabled, resuming after power failure recovery executes the rest of the action program. In this case, the action program is executed from the point where the TCP was at the power failure. Execution timing is different from the usual execution timing.

10.16.12 Adding the Distance Before Motion Option (Procedure)

Use Procedure 10-28 to add the Distance Before motion option to a motion instruction.

Procedure 10-28 Adding the Distance Before Motion Option to a Motion Instruction

Steps

1. Select the teach pendant program you want to edit.
2. Press EDIT.
3. Insert a motion instruction or move the cursor to an existing motion instruction.
4. Move the cursor to the motion option area of the motion instruction to which you want to add the Distance Before motion option. See the following screen for an example.

```
PNS0001
  1: J P[1] 100% FINE
[END]
```

5. Press F4, [CHOICE]. You will see a screen similar to the following.

```
Motion Modify
 1 TIME BEFORE           5
 2 TIME AFTER            6
 3 DISTANCE BEFORE       7
 4                       8
PNS0001
  1: J P[1] 100% FINE
[END]
Select item
```

6. Select DISTANCE BEFORE.

DB is added to the program. See the following screen for an example.

```
PNS0001
  1: J P[1] 100% FINE
   :   DB [0.0] mm ...
[END]
Enter Value
```

7. Type the distance value and press ENTER. You will see a screen similar to the following.

```
TIME statement
 1 CALL program              5 GO[ ]=...
 2 CALL program( )           6 AO[ ]=...
 3 DO[ ]=...                 7
 4 RO[ ]=...                 8
PNS0001
  1: J P[1] 100% FINE
   :   DB 100.0mm
[END]
 Select item
```

8. Select the action you want.

   - For a CALL program, go to Step 9 .
   - For signal output, go to Step 10 .

9. **To insert a CALL Program action:**

   a. Select CALL program or CALL program (for a program with parameters). You will see a screen similar to the following.

```
PROGRAM list
 1 HANDOPEN              5
 2 HANDCLOS              6
 3                       7
 4                       8
```

   b. Select the program you want to call.

   c. **If you want to add parameters to the program call** , select the appropriate information to add the parameters to the program call as you would to any program call.

10. **To insert a signal output action:**

   a. Select the output signal you want: DO, RO, GO, or AO.

   b. Press the appropriate function keys to enter the output signal number.

11. **To find a program or output signal that is used by the DB motion option** ,

   a. Press NEXT, >, until F5, [EDCMD], is displayed.

   b. Press F5, [EDCMD].

   c. Select Find.

   d. Select the appropriate item:

      - For a program CALL, select CALL and then select CALL Program.

      - For an output signal, select I/O and then select the appropriate signal.

12. **To replace Distance Before with Time Before/After** ,

   a. NEXT, > until F5, [EDCMD], is displayed.

b. Press F5, [EDCMD].

c. Select Replace.

d. Select the appropriate items to find and replace as desired.

10.16.13 Cautions and Limitations

When you use the Distance Before option, be aware of the following cautions and limitations:

- Distance Before cannot be used with TIME BEFORE/AFTER.
- More than six motion instructions with Distance Before cannot be processed at the same time.
- Robot TCP path may change during Hold, ESTOP and then resume (i.e. no constant path feature in those cases).



Caution

In the case of Hold or Estop, then resume around a motion line with the Distance Before statement, the path will deviate. It is important to verify that the path is all right.

- Distance Before calculates the distance between the current position and the destination point cyclically. Because the trigger condition is judged by this cyclical checking, the actual execution of the timing of the action is different from the distance value. The action might be executed inside of the trigger region. This means that the point where the instruction is executed is closer than the distance value. The degree of difference depends on the speed of the robot. The slower the TCP moves, the more accurate the execution timing.
- Distance Before cannot be recovered by power failure recovery if it is attached to a motion instruction with CNT termination type and power turns off when the motion is about to complete.
- Distance Before cannot be used with INC, SKIP, and QSKIP in a motion instruction.
- Multiple group motion is not supported.
- Robots that do not have Cartesian coordination cannot use Distance Before.
- Positional data in matrix form is not supported.
- Integrated axes are not supported.
- Distance Before is not supported on an F-200i.
- If $DISTBF_VER=2, CJP is enabled, and the TCP does not go into the trigger region, the execution of the program might stop.
- If the program ends before the DB condition triggers, the execution of the instruction is not processed, even if the DB condition is triggered after program execution completes.

10.16.14 System Variables

Refer to Table 10–34 for a listing and description of related system variables.

Table 10–34. Distance Before System Variables

VARIABLE	DESCRIPTION

$DISTBF_VER
default: 1
	This system variable sets the execution timing of the line that is just after the motion instruction that contains the DB motion option. The values are as follows:
	• 1-Execution of the next line will start without waiting for the completion of the DB action;
	• 2-Execution of the next line will not start until the completion of the DB action; For example,
	`1 L P[1] 2000 mm/sec FINE`
	`2 L P[2] 2000 mm/sec CNT100 DB 1.0 mm DO[1]=ON`
	`3 L P[3] 2000 mm/sec FINE`
	with $distbf_ver = 2, the execution of line 3 does not start until DO[1] is turned ON. With $distbf_ver=1, line 3 is executed as if there were no DB motion option.
	• 3-The system will automatically determine the behavior based on DB action routine itself:
	— If DB action routine is not too long and does not contain WAIT statement, then execution of the next line will start without waiting for the completion of the DB action (the same as when $distbf_ver = 1);
	— If DB action routine is lengthy or has a WAIT statement, execution of the next line will not start until the completion of the DB action (the same as when $distbf_ver=2);
	Note With $distbf_ver = 1, 2, or 3, no constant path feature can be obtained. That is, the TCP path of a robot varies with the distance value in the DB instruction and the time in a WAIT statement of the action (if the action has a WAIT statement);
	• 4-During normal execution (not Hold,E-stop/resume case), the system will try to maintain constant path feature: the same TCP path regardless of the distance value in a DB instruction, regardless of the time in a WAIT statement in action and regardless of speed override;

$DB_AWAYTRIG default: 0.08 mm	Distance Before calculates the distance between the current position and destination cyclically. The controller recognizes that the TCP is *going away* from the destination point if this calculated distance is greater than the previous value by $DB_AWAYTRIG millimeters. Refer to Section 10.16.6 .

Table 10–34. Distance Before System Variables (Cont'd)

VARIABLE	DESCRIPTION
$DB_TOLERANCE default: 0.05 mm	The radius of the trigger region is (distance value + $DB_TOLERANCE). If the distance value is less than $DB_MINDIST, the radius is ($DB_MINDIST + $DB_TOLERANCE). Refer to Section 10.16.6 .
$DB_CONDTYP default: 2	This system variable defines the DB trigger condition. The values are as follows: • 0 - The trigger condition is when the TCP goes into a region that is within the distance value; this is known as *region trigger* . • 1 - The trigger condition is also *going away* , in addition to region trigger. • 2 - The trigger condition is also *penetration* , in addition to region trigger. Refer to Section 10.16.6 .
$DBCONDTRIG default: 0	This system variable decides which alarm is posted when a DB condition is not triggered. The values are as follows: • 0 - INTP-295 WARN (program name,line number) DB condition was not triggered • 1 - INTP - 293 PAUSE.L (program name,line number) DB condition was not triggered Refer to Section 10.16.7 .
$DB_MINDIST default: 1.0 mm	This system variable is the internal minimum value of the distance value. If the distance value is smaller than this value by $DB_MINDIST or more, $DB_MINDIST is used as the distance value instead of the distance value you defined. Refer to Section 10.16.4 .
DISTBF_TTS	This system variable specifies the execution timing of the DB action when a motion instruction with Distance Before is held. Refer to Section 10.16.9 .

10.16.15 Error Codes

Refer to the *System Software Error Code Manual* for information about errors or to Appendix A , "Error Codes and Recovery," for recovery procedures.

10.17 EASY NORMAL UTILITY (formerly known as Auto Normal Utility)

10.17.1 Overview

The Easy Normal utility is a software option that is used to align the end-of-arm tooling of the robot perpendicular, or normal, to a surface, such as a workpiece. This option assumes that a height or range sensor is available on the end of the robot arm.

FANUC Robotics has successfully used the Easy Normal utility in laser cutting applications in which a laser height sensor is integral with the cutting tool on the end of the robot arm. This cutting tool with the sensor can be purchased from FANUC Robotics. You can also use this option with your own range or height sensors. Be sure to consider carefully the mounting and usage of the height sensor so you can use the Easy Normal utility effectively. Contact FANUC Robotics for further assistance.

The Easy Normal utility provides a setup screen in which you can enter various search parameters to tune the utility to your particular application. The search parameters are used with a proprietary algorithm to move the robot in a particular manner relative to the surface, such as the workpiece. At the end of these motions, the robot will re-orient the end of the robot arm to be normal to the surface. The jog frame of the robot is also set up auto so that when you jog the robot in the x and y directions, the end of the robot arm maintains its perpendicular orientation relative to the surface of the workpiece.

Without the Easy Normal utility, it would be time consuming to define an accurate relative position and orientation. However, in applications such as laser cutting of shapes, the tool location and orientation are critical for cut quality.

The Easy Normal utility does the following:

- Simplifies the teaching of robot positions
- Reduces teaching time
- Improves the accuracy of taught positions relative to the work surface

To use the Easy Normal utility option, you do the following:

1. Set up Easy Normal - Section 10.17.3
2. Execute Easy Normal to perform the search and yield the appropriate positions and frames for accurate teaching - Section 10.17.4

10.17.2 Using the Easy Normal Utility

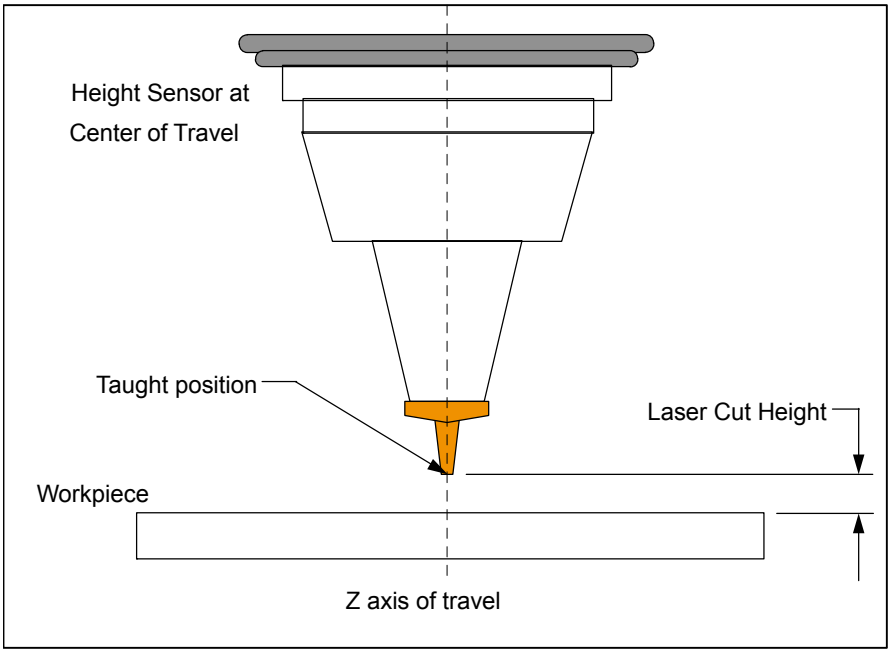
This section contains information on the following topics, which you must know in order to use the Easy Normal utility effectively. The descriptions in this section use a laser height sensor as an example. You can use your own height sensor in a similar manner.

- Laser height sensor
- Search motion
- Shape frames

Laser Height Sensor

The laser cutting tool has an integrated height sensor. The laser height sensor uses an inductive proximity sensor to locate, within a certain degree of accuracy, a surface within the sensor range. During the searching utility, the height sensor will be locked to its center of travel. The final position will be at a known distance (called the laser cut height) above the workpiece controlled by the laser controller. See Figure 10–52 .

Figure 10–52. Laser Height Sensor

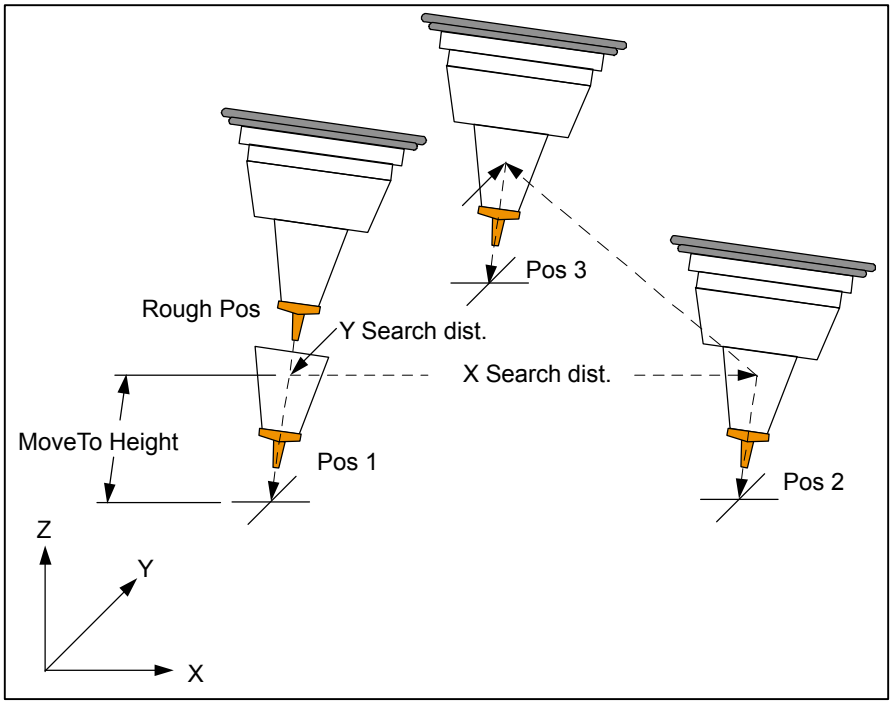


Search Motion

The Easy Normal utility moves the robot, in a specified manner, from a rough position toward an object in the TOOL Z direction. The robot stops searching and moving when the sensor detects an

object. A three-point search algorithm is used to locate the object and to determine its orientation and distance relative to the robot. See Figure 10–53 .

Figure 10–53. Search Motion

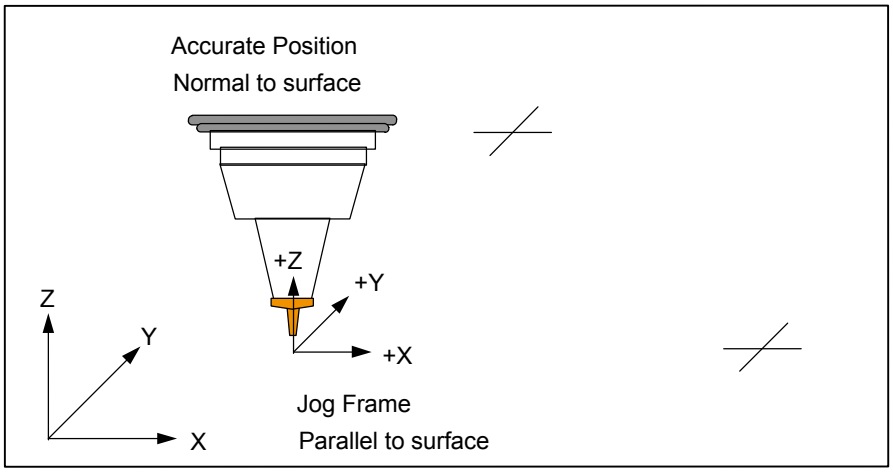


Three **search sizes** are available: small, medium, and large. A larger search on a flat surface will produce a more accurate normalized position. However, smaller search sizes will allow you to perform a search within a small area where tight teaching tolerances are required.

After a search has been completed, the robot will be located at a new position that is normal (or perpendicular) to the surface where a teach pendant position can be recorded or touched up. The cutting torch will be located at a known distance above the workpiece, determined by the laser controller.

A **jog frame** is now set, where the X-Y plane is parallel to the surface of the *shape frame* . This reduces the amount of teaching time by allowing you to jog the robot parallel to the surface and record additional positions located along the same surface. See Figure 10–54 . Refer to the next section ("Shape Frames") for more information on shape frames.

Figure 10–54. Positions and Frames Defined After the Search



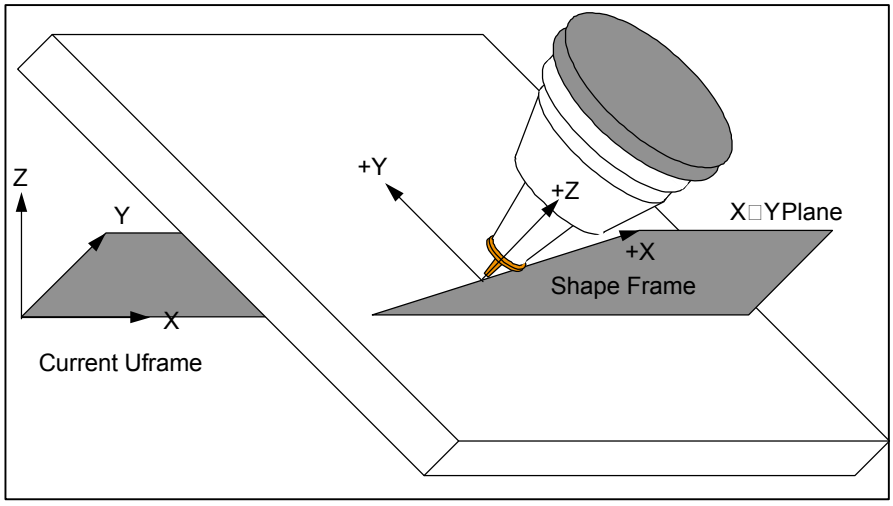
Shape Frames

A shape frame is calculated from a shape center position and current uframe:

- The shape frame **X-direction** is parallel to the object surface where the current UFRAME X-Y plane intersects the object at the shape center position.
- The shape frame **Y-direction** is also parallel to the object surface but perpendicular to the X-direction.
- The shape frame **Z-direction** is *into* the tool along the TOOL Z vector.

When you define the UFRAME (or shape frame), use the six-point method. Also, teach the Z-direction *into* the tool, as shown in Figure 10–55 .

Figure 10–55. Shape Frames



10.17.3 Setup

You set up the Easy Normal utility by setting items on the SETUP Easy Normal screen. Table 10–35 lists and describes the items you must set. Use Procedure 10-29 to set up Easy Normal.

Table 10–35. Easy Normal Setup Items and Operations

ITEM	DESCRIPTION
Sensor Input	This item allows you to specify the number of the digital input that the robot will receive when the sensor locates an object.
Search Speed default: 20 mm/sec min: 1 mm/sec max: 20 mm/sec	This item allows you to specify the speed the robot will use to search toward the object. In general, slower search speeds result in more accurate results.
MoveTo Speed default: 200 mm/sec min: 1 mm/sec max: 200 mm/sec	This item allows you to specify the travel speed the robot will use while moving between search positions.
Search Distance default: 20 mm min: 1 mm max: 50 mm	This item allows you to specify the maximum distance the robot will search toward an object before the robot ends the search and posts an error message.

Table 10–35. Easy Normal Setup Items and Operations (Cont'd)

ITEM	DESCRIPTION
MoveTo Height default: 10 mm min: 1 mm max: 50 mm	This item allows you to specify the height above an object at which the robot will travel to and from search positions.
Current Search Size default: SMALL values: SMALL, MEDIUM, LARGE	This item allows you to specify the currently selected search size in the x-y plane: • **SMALL** - x = 10 mm; y = 10 mm • **MEDIUM** - x = 15 mm; y = 15 mm • **LARGE** - x = 20 mm; y = 20 mm In general, larger search areas result in more accurate normal positions
Small X default: 10 mm Small Y default: 10 mm Medium X default: 15 mm Medium Y default: 15 mm Large X default: 20 mm Large Y default: 20 mm min: 10 mm max: 200 mm	These items allow you to define the x and y distances between search positions for each kind of search size: small, medium, and large.
SMALL	Press this key to define the x and y distances between search positions for the search size of small.
MED	Press this key to define the x and y distances between search positions for the search size of medium.
LARGE	Press this key to define the x and y distances between search positions for the search size of large.

Procedure 10-29 Setting Up Easy Normal

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].

4. Select Easy Normal. You will see a screen similar to the following.

```
SETUP Easy Normal
  1  Sensor input:              DI[13]
  2  Search speed:               20 mm/sec
  3  MoveTo speed:              200 mm/sec
  4  Search distance:            10 mm/sec
  5  MoveTo height:              20 mm
  6  Current search size:        SMALL
     [SEARCH]           X           Y
  7  Small:          10.000       10.000 mm
  8  Medium:         15.000       15.000 mm
  9  Large:          20.000       20.000 mm
```

5. Move the cursor to each item and set as desired.

10.17.4 Execution

The Easy Normal utility uses a teach pendant macro to call a KAREL routine to execute the search. Use Procedure 10-30 to execute the Easy Normal utility.

The teach pendant macro is named EASY NORMAL. the EASY NORMAL macro executes the teach pendant program named NORMAL.TP. The NORMAL.TP teach pendant program contains one instruction, "CALL FINORM." FINORM is a KAREL program that contains all of the search motions.

Procedure 10-30 Executing Easy Normal to Perform a Search

Steps

1. Hold the teach pendant and continuously press the DEADMAN switch on the back of the teach pendant.

2. Turn the teach pendant ON/OFF switch to the ON position.



Warning

In the next step, the robot will move. To stop the robot immediately any time during jogging, release the DEADMAN switch or press the EMERGENCY STOP button. Otherwise, you could injure personnel or damage equipment.

3. Press and hold the SHIFT key and press the MAN FCTNS key. After the robot starts moving, you can release the MAN FCTNS key but must **continue holding down** the SHIFT key.

Note If you release the SHIFT key while the robot is executing the search, the Easy Normal search sequence will be aborted. If this happens, jog the robot or step back to the desired center position and repeat Step 1 through Step 3 .

When the Easy Normal utility has completed the search sequence, a jog frame is set to jog the robot with x and y directions parallel to the workpiece along the shape frame.

10.18 ENHANCED MIRROR IMAGE (OPTION)

The enhanced mirror image option allows you to translate an entire teach pendant program or portion of a teach pendant program to mirror-image the original programmed points. This option can be used to teach symmetrical parts easily. The enhanced mirror image option provides additional functionality so that existing frames such as user frames can be used directly with any specified XY, YZ, or XZ plane. Also, additional orientation shifting is provided for different application requirements.

Note For PaintTool and paint robots, this is almost always accomplished using left-hand and right-hand robots. Use of this option is not recommended.

Mirror imaging of a program can be accomplished either as:

- mirror image using existing frames such as World frame or user frames. No teaching required.
- mirror image using flexible mirror planes such as X-Y, Y-Z, or X-Z from existing frames for better selection
- parallel mirror image with mirrored orientation
- parallel and rotational mirror image with mirrored orientation
- mirror image orientation shifting can be selected based on application requirements.

Refer to Table 10–36 for descriptions of the Mirror Image Shift screen items.

Table 10–36. ENHANCED MIRROR IMAGE Screen Items

ITEM	DESCRIPTION
Original Program	This item is the name of the program that you want to shift using mirror image. To select a program, press ENTER, type the name of the program, and press ENTER again.
Range	This item indicates whether the whole program or a portion will be shifted using mirror image.

Table 10–36. ENHANCED MIRROR IMAGE Screen Items (Cont'd)

Start line	This item is the first line in the program that will be shifted using mirror image.
End line	This item is the last line in the program that will be shifted using mirror image.
New Program	This item is the name of the program that results when you offset or shift the Original Program. If you want the resulting offset or shifted program to replace the Original Program, make the New Program name the same as the Original Program name.
Insert line	This item indicates the location in an existing program where new shifted position information will be inserted.
Position data (used only for "Teach" method)	This item is the current x, y, and z position information.
Image Shift Methods	• Teach is a method that requires the user to enter position data to define the mirror plane. • Frame is a method that allows the user to choose existing frames.
Orientation Shift	• Mirror is a mirror image shifting method that mirrors the orientation as well as the positions. • Fixture is a mirror image shifting method that only mirrors the approach vector of the orientation but leaves the other two vectors intact with flexible tool orientation control.
Mirror Frame:	World, User Frame, or Jog Frame can be selected as the mirror frame.
Frame No:	When User frame or Jog frame is selected, this indicates which frame number will be used.
Mirror Plane:	The XY, YZ, or XZ plane of the selected mirror frame can be used as mirror plane.
Rotation (used only for "Teach" method)	This item, if set to ON, indicates that you want to rotate three position about an axis. If set to OFF, you can shift a single source position to a single destination position.

Table 10–36. ENHANCED MIRROR IMAGE Screen Items (Cont'd)

Source position (used only for "Teach" method)	This item is the original position that you will be shifting using mirror image. If Rotation is set to ON, there will be three source positions.
Destination position (used only for "Teach" method)	This item is the destination position to which you will be shifting using mirror image. If Rotation is set to ON, there will be three destination positions.

Parallel Mirror Image

A parallel mirror image mirrors the program about a mirror plane without an offset or a rotation. See Figure 10–56 and Figure 10–57 .

Figure 10–56. Parallel Mirror Image with Mirror Plane in Center of Robot

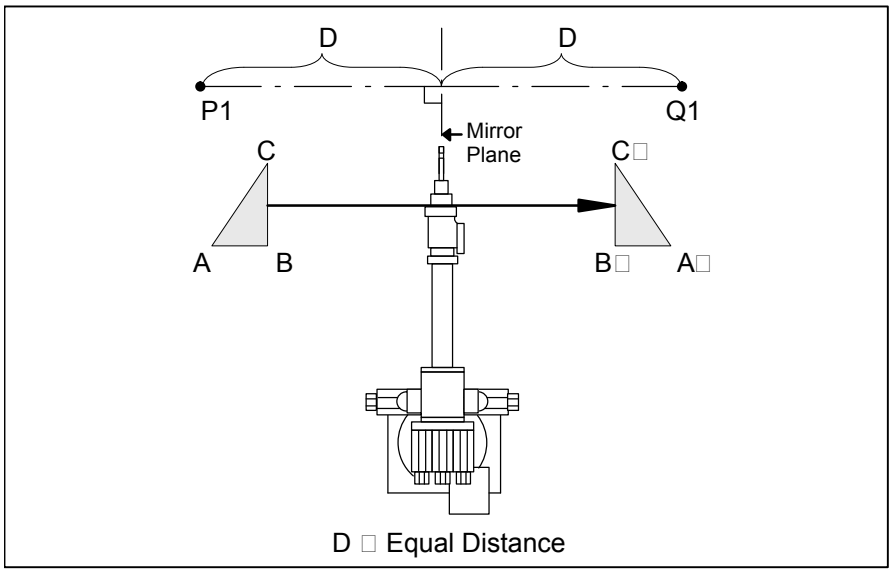
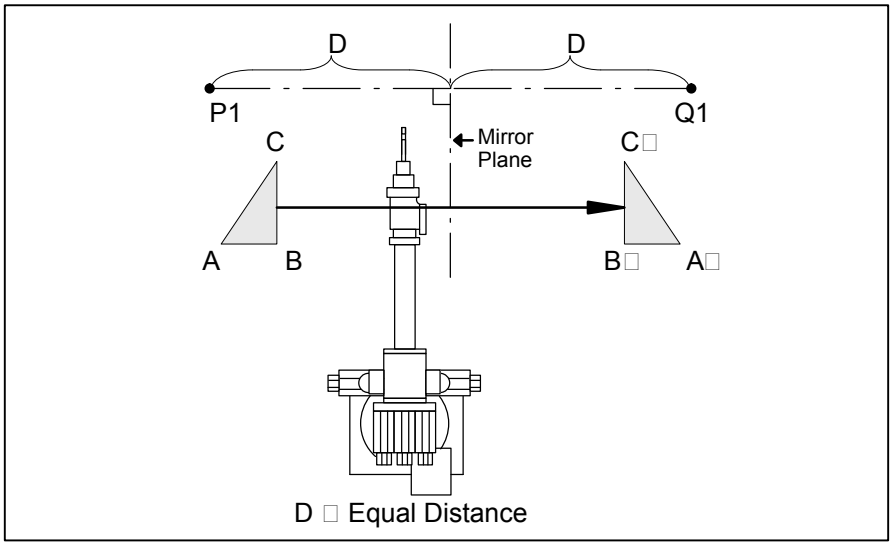


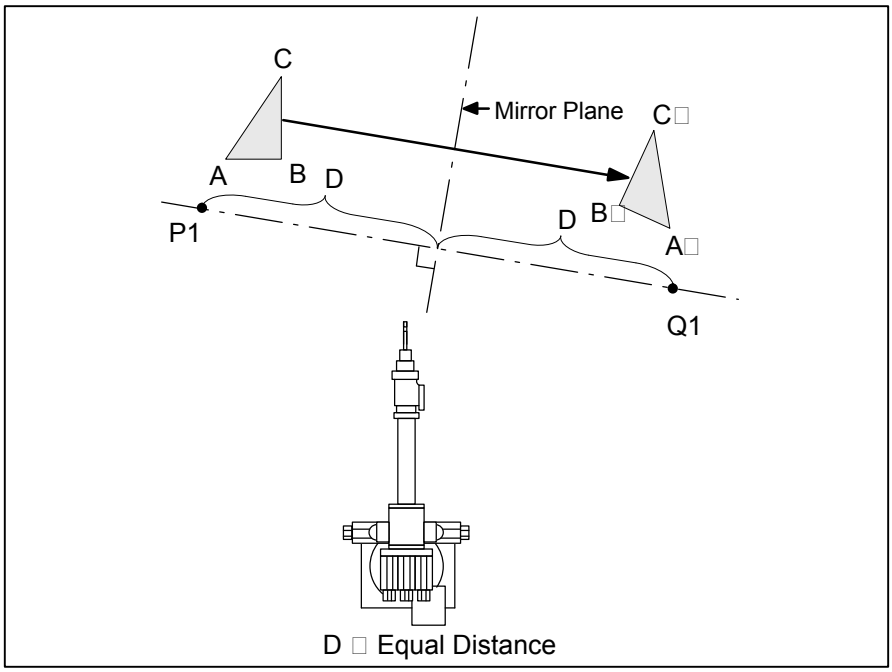
Figure 10–57. Parallel Mirror Image with Mirror Plane Offset from Center of Robot



Caution

To be sure the parallel mirror image works correctly, you must have an exact TCP. If you do not, the resulting mirror image program will contain an offset value. See Figure 10–58 .

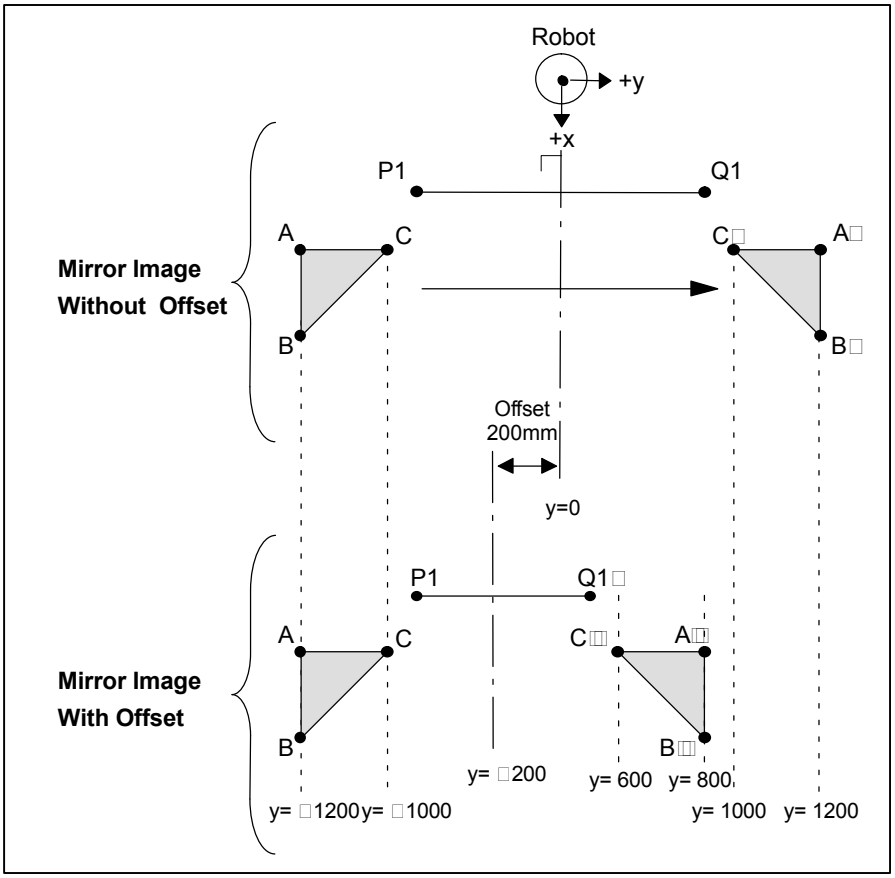
Figure 10–58. Parallel Mirror Image with Offset



Mirror Image Example

Figure 10–59 displays a mirror image of A, B, and C to A', B', and C' when P1 and Q1 are taught positions. Figure 10–59 also displays a mirror image with an offset when P1 and Q1 are taught but Q1 is taught at an offset of 200mm. In this case, the result is A'', B'', and C''.

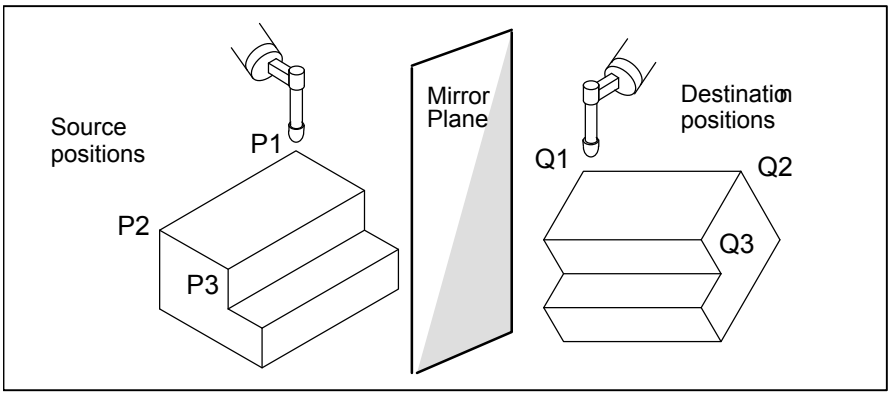
Figure 10–59. Positional Mirror Image



Rotational Mirror Image

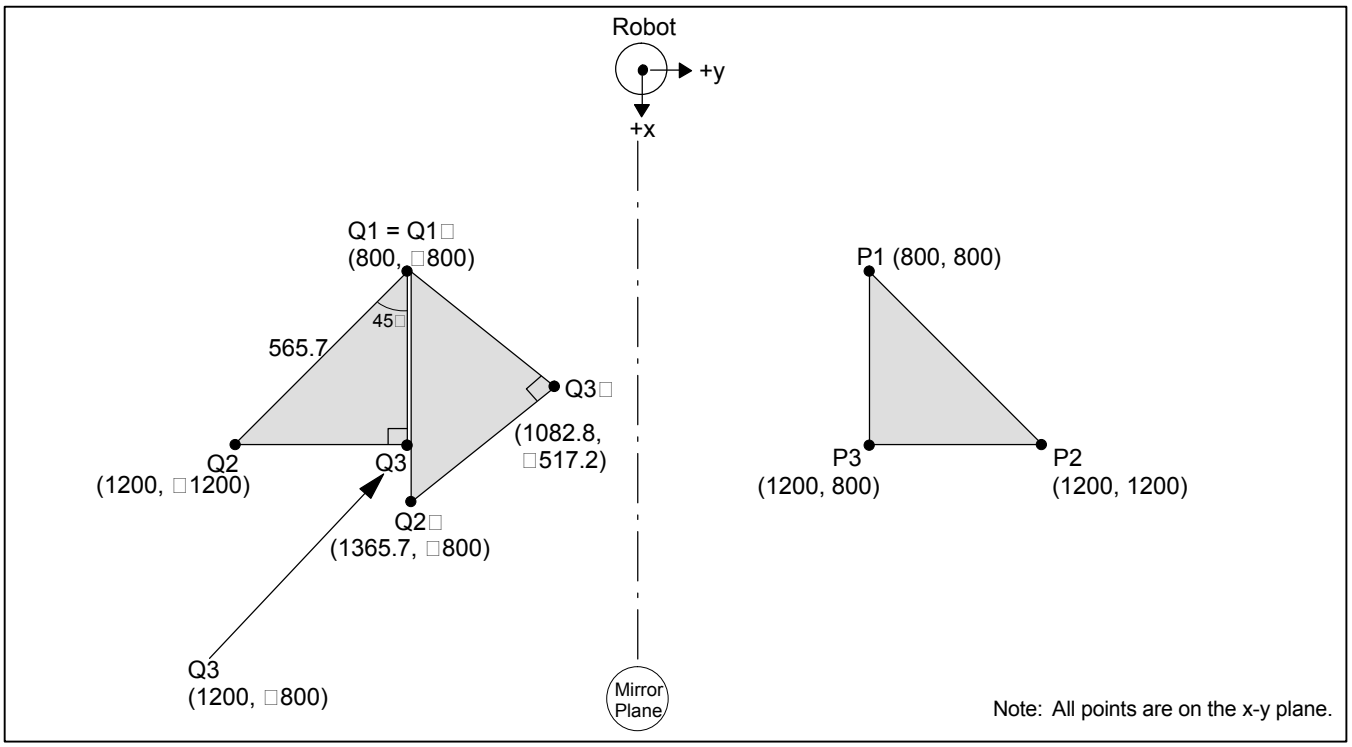
A parallel mirror image mirrors the program about a mirror plane first, and then the mirrored program is rotated about a center of rotation. Orientation of the part to be imaged is rotated about one or more of its axes relative to the mirror plane. See Figure 10–60 .

Figure 10–60. Rotational Mirror Image



In Figure 10–61 the positions, P1, P2 and P3 are mirrored about the mirror plane as Q1, Q2 and Q3. These positions are then rotated 45°about Q1 and stored as Q1', Q2' and Q3'.

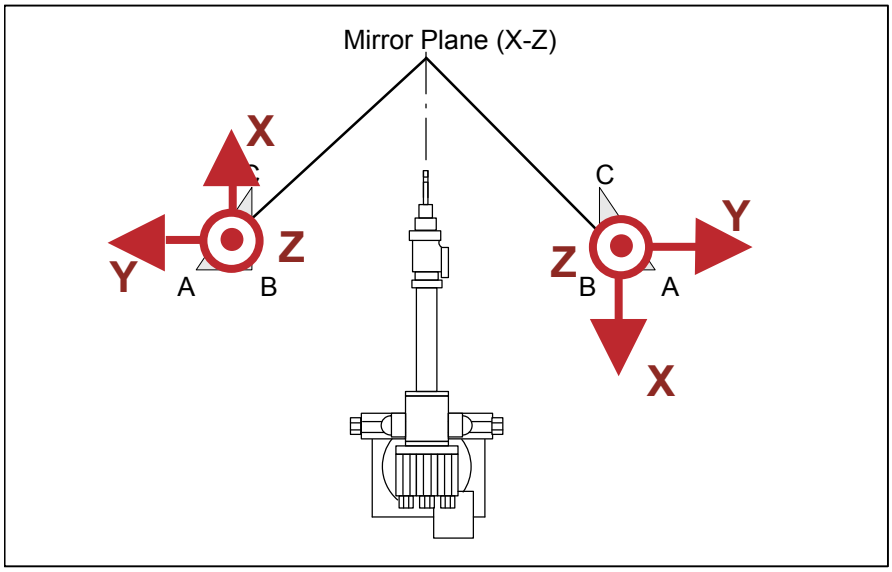
Figure 10–61. Rotational Mirror Image



Mirror Image using Existing Frames and Mirror Planes, with Controlled Orientation

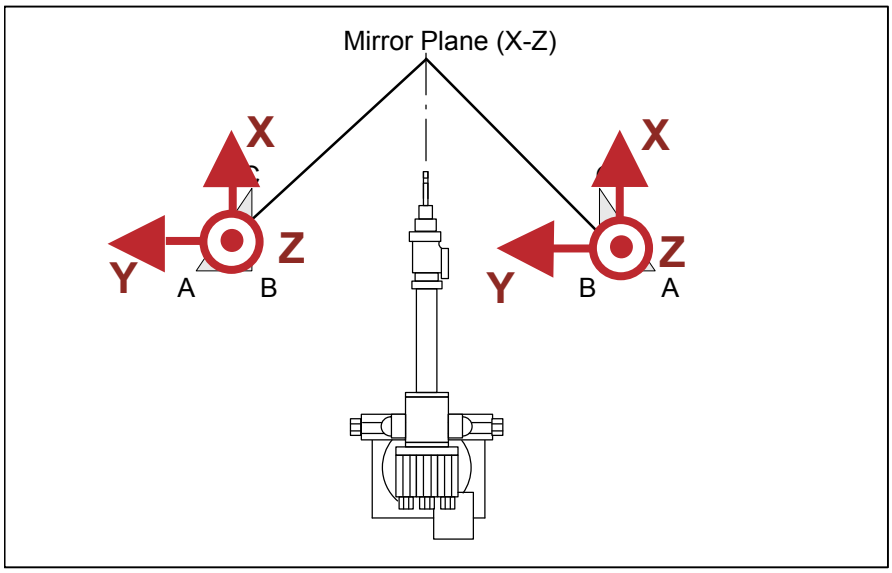
The standard mirror image utility mirrors orientation as well as positions, as shown in Figure 10–62 .

Figure 10–62. Mirror Image Shift with Orientation Mirrored (Mirror Method)



The enhanced mirror image option also provides a simple way to use existing frames, such as World, User, or Jog Frame, with defined mirror planes, such as X-Y, Y-Z, or X-Z. Also, to control the orientation as well as the positions in the mirrored program, the Orientation Mirror method can be selected as "Mirror" as shown in Figure 10–62 , or "Fixture", as shown in Figure 10–63 .

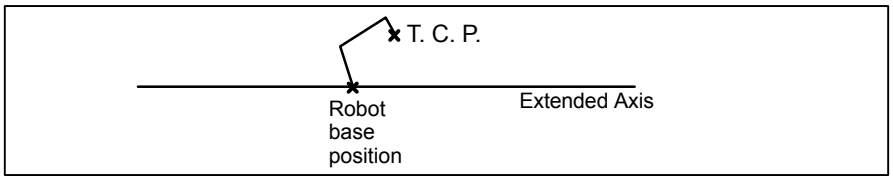
Figure 10–63. Enhanced Mirror Image with Orientation Controlled (Fixture Method)



Mirror Image of Extended Axes

Extended (Ext) axes determines how the mirror image function will translate the program when you are using extended axes. Figure 10–64 displays a key for use with Figure 10–65 through Figure 10–67 .

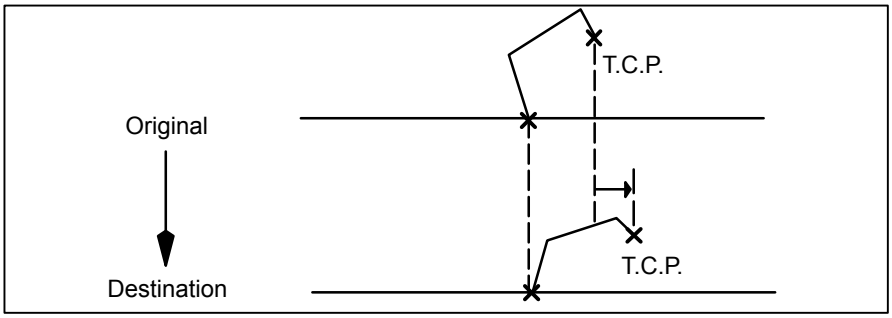
Figure 10–64. Mirror Image Key



Possible extended axes configurations are:

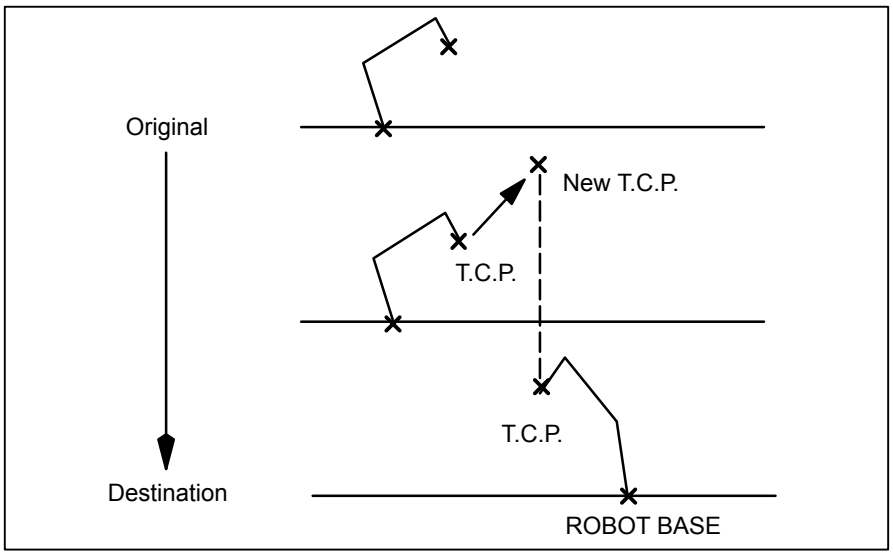
- **Robot axes only** - allows you to mirror the axes of the robot without mirroring any non-integrated extended axes such as a positioning table. The shift is calculated using the change in the robot tool center point (TCP).

Figure 10–65. Example of Robot Axes Only Mirror Image



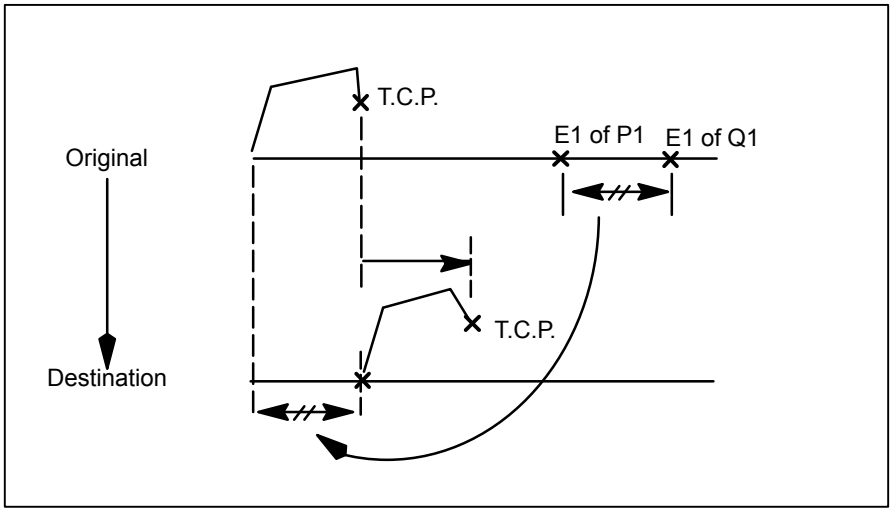
- **Ext integrated** - allows you to mirror the axes of the robot and any integrated axes. The amount of mirror image for the robot and the extended axes is calculated using the change in the TCP. See Figure 10–66 .

Figure 10–66. Example of Extended Axes Integrated Mirror Image



- **With ext axes** - allows you to mirror positions for robot axes and any extended axes in your system. The amount of mirror image for the robot is calculated by using the change in the TCP. The amount of shift for the extended axes is calculated using the center of the difference between an original position (P1) and a new position (Q1) as the point where the mirror image occurs. See Figure 10–67 .

Figure 10–67. Example of With Extended Axes Mirror Image



- **Ext axes only** - Ext axes only is not available for a mirror shift.
- **Replace Ext axes** - Replace Ext axes is not available for a mirror shift.

Procedure 10-31 Using Enhanced Mirror Image Shift

Conditions

- The program you want to mirror has been created and contains recorded positions.
- All robot joint axes are at zero degrees.
- If you are using PaintTool applications and are using mirror image to transfer positions between right and left hand robots, be sure you are performing the mirror image on the **destination** robot and not the source robot. Therefore, you must first copy the program from the source robot to the destination robot before you perform the mirror image.

Steps

1. Press MENUS.
2. Select UTILITIES.
3. Press F1, [TYPE].
4. Select Mirror Image Shif. You will see the MIRROR IMAGE SHIFT screen.

```
MIRROR IMAGE SHIFT
 Settings
   Original Program:
 1
 2   Range:                          WHOLE
 3   Start line: (not used)           ****
 4   End line: (not used)             ****
 5 New Program:              [           ]
 6  Insert line: (not used)           ****
 7 Image Shift Method:               TEACH
 8 Orientation Shift:                MIRROR
 9 EXT axes:             Robot axes only
```

5. Move the cursor to 1 under Original Program. If the program you want to mirror is not selected, press F4, [CHOICE]. Select the program from the list and press ENTER.

   Note The last program selected using the SELECT menu will automatically be named as the original program.

6. Move the cursor to Range. Use the function keys to select whether you want to mirror all or part of the original program.

   - **To mirror the whole program, press F5, WHOLE.** You will see a screen similar to the following.

```
MIRROR IMAGE SHIFT
 Settings
   Original Program:
 1 STYLE37
 2   Range:                          WHOLE
 3   Start line: (not used)          *****
 4   End line: (not used)            *****
 5 New Program:              [            ]
 6  Insert line: (not used)          *****
 7 Image Shift Method:               TEACH
 8 Orientation Shift:                MIRROR
 9 EXT axes:             Robot axes only
```

Note EXT axes will be displayed only if you are using mirror image for extended axes.

- **To shift part of the program, press F4, PART.** You will see a screen similar to the following.

```
MIRROR IMAGE SHIFT
 Settings
   Original Program:
 1 STYLE37
 2   Range:                          PART
 3   Start line:                        0
 4   End line:                          0
 5 New Program:         [                  ]
 6  Insert line: (not used)          *****
 7 Image Shift Method:               TEACH
 8 Orientation Shift:                MIRROR
 9 EXT axes:             Robot axes only
```

7. **If you selected to mirror PART of a program,** select the lines of the program that you want to mirror:

   - Move the cursor to Start line and type the starting line number. Press ENTER.
   - Move the cursor to End line and type the ending line number. Press ENTER.

8. Move the cursor to New Program and press ENTER. Use the appropriate function keys to type the name of the new program and press ENTER. This is the program to which you will be mirroring the positions.

Note You can mirror the positions of an entire program or portion of a program from within a program or from one program to another. See Figure 10–68 and Figure 10–69 .

Figure 10–68. Mirroring an Entire Program

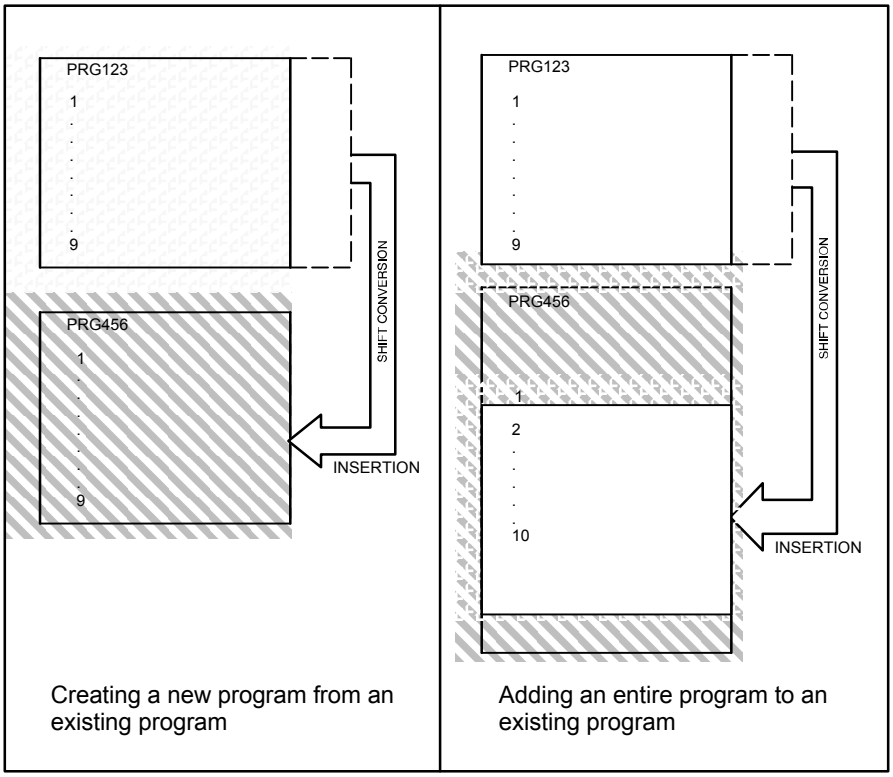
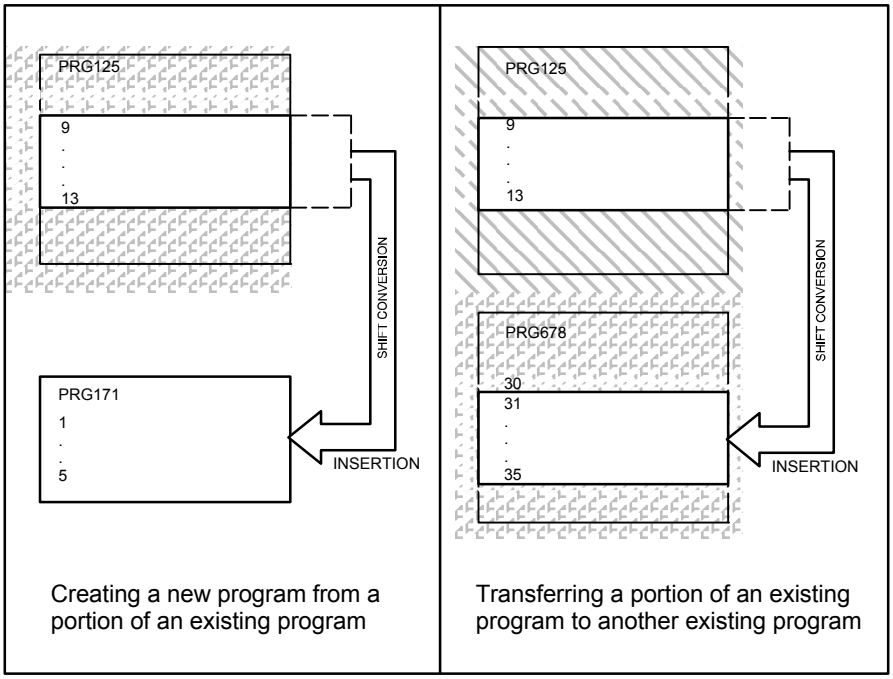


Figure 10–69. Mirroring a Portion of a Program



Note The New Program can be the same as the original program, an already existing program, or a program that does not exist.

- **If you are inserting lines into the original program,** type in the name of the program and the corresponding line numbers.
- **If you are inserting lines into an existing program,** move the cursor to Insert line and type the line number at which you want to insert the shifted information.

9. Move the cursor to Image Shift Method.

- **To mirror the image using existing User, Jog, or World frames** and well-defined mirror frames such as Z-Y, Y-Z or X-Z, press F4, FRAME. Refer to Figure 10–63 .
- **To mirror the image by teaching points to define the mirror plane**, press F5, TEACH. This method also mirrors the orientation. Refer to Figure 10–62 .

10. Move the cursor to Orientation Shift.

- **To control the orientation shift** in the mirrored program, press F4, FIXTURE.
- **To mirror the orientation** of the original program, press F5, MIRROR.

11. Press the down arrow key.

- **If you selected TEACH** for the Image Shift Method, you will see a Shift amount/Teach screen similar to the following:

```
MIRROR IMAGE SHIFT
  Shift amount/Teach
  Position data
  X :********  Y :********  Z :********
  1 Rotation:                        OFF
  2 Source position:        P1:
  3 Destination position:   Q1:
```

Continue to Step 12 .

- **If you selected FRAME** for the Image Shift Method, you will see a Shift Frame Settings screen similar to the following:

```
MIRROR IMAGE SHIFT
  Shift Frame Settings

1 Mirror Frame:               WORLD
2   Frame No. :               *****
3 Mirror Plane:                  XY
```

Continue to step Step 15 .

12. For the TEACH method, move the cursor to Rotation and press F4, ON. You will see a screen similar to the following:

```
MIRROR IMAGE SHIFT
Shift amount/Teach
  Position data
  X :********  Y :********  Z :********
  1 Rotation:                        ON
  2 Source position:        P1:
  3                         P2:
  4                         P3:
  5 Destination position:   Q1:
  6                         Q2:
  7                         Q3:
```

HINTS: P1 and Q1 should not be part of the program to be mirrored. They should be new positions that are located at an equal distance from the mirror plane. (See Figure 10–56 .) Use the following guidelines to teach these positions:

a. Jog the robot to zero degrees by matching the witness marks on the robot (if they are available), or by displaying the POSITION screen.

b. Jog the robot:

- Set the jog coordinate system to WORLD.
- Jog the robot in +Y by a known distance, 200 mm for example.

OR

- Set the jog coordinate system to JOINT.
- Jog the robot a known angle, 20°, for example.

c. Record this position as P1.

d. Jog the robot back to zero.

e. Jog the robot in the opposite direction by the exact same distance from the mirror plane as you jogged the robot in Step 10b .

f. Record this position as Q1. See Figure 10–56 .

13. If you do not want to rotate the positions,

a. Press F5, OFF. You will see a screen similar to the following.

```
MIRROR IMAGE SHIFT
Shift amount/Teach
  Position data
  X :********  Y :********  Z :********
  1 Rotation:                      OFF
  2 Source position:        P1:
  3 Destination position:   Q1:
```

b. Move the cursor to Source position (see Figure 10–56). You will see a screen similar to the following.

```
MIRROR IMAGE SHIFT
Shift amount/Teach
  Position data
  X :********  Y :********  Z :********
  1 Rotation:                      OFF
  2 Source position:        P1:
  3 Destination position:   Q1:
```

c. Move the robot to the source position (P1) and either record or specify the position:

- **To record a position,** jog the robot to the position you want, press and hold in the SHIFT key and press F5, RECORD.
- **To specify a previously recorded position or position register,** press F4, REFER. Type the number of a previously defined position or position register, and press ENTER.

```
Select reference position type
                P[ ]   PR[ ]
```

- **To clear a position setting** , press NEXT, >, and then press F1, CLEAR. At the prompt "Clear all data," press F4, YES, to confirm.

d. Move the cursor to Destination position (see Figure 10–101). You will see a screen similar to the following.

```
MIRROR IMAGE SHIFT
Shift amount/Teach
  Position data
  X :********  Y :********  Z :********
  1 Rotation:                      OFF
  2 Source position:         P1:    P[1]
  3 Destination position:    Q1:
```

- **To record a position,** jog the robot to the destination position (Q1). Press and hold in the SHIFT key and press F5, RECORD.
- **To specify a previously recorded position register,** press F4, REFER. Type the number of the position register and press ENTER.

```
Input position register number:
```

- **To clear a position setting** , press NEXT, >, and then press F1, CLEAR. At the prompt "Clear all data," press F4, YES, to confirm.

e. Record or specify the destination position.

14. If you want to rotate the positions,

a. Press F4, ON. You will see a screen similar to the following.

```
MIRROR IMAGE SHIFT
Shift amount/Teach
  Position data
  X :********  Y :********  Z :********
  1 Rotation:                        ON
  2 Source position         P1:
  3                         P2:
  4                         P3:
  5 Destination position    Q1:
  6                         Q2:
  7                         Q3:
```

b. Move the cursor to Source position (see Figure 10–56). You will see a screen similar to the following.

```
MIRROR IMAGE SHIFT
Shift amount/Teach
  Position data
  X :********  Y :********  Z :********
  1 Rotation:                        ON
  2 Source position         P1:
  3                         P2:
  4                         P3:
  5 Destination position    Q1:
  6                         Q2:
  7                         Q3:
```

c. Move the robot to the first source position (P1) and either record or specify the position:

- **To record a position,** jog the robot to the position you want, press and hold in the SHIFT key and press F5, RECORD.
- **To specify a previously recorded position or position register,** press F4, REFER. Select the position or position register.

```
Select reference position type
               P[ ]   PR[ ]
```

- **To clear a position setting** , press NEXT, >, and then press F1, CLEAR. At the prompt "Clear all data," press F4, YES, to confirm.

d. Record or specify all the source positions.

e. Move the cursor to Destination position (see Figure 10–101). You will see a screen similar to the following.

```
MIRROR IMAGE SHIFT
Shift amount/Teach
  Position data
  X :********  Y :********  Z :********
  1 Rotation:                      ON
  2 Source position         P1:    P[1]
  3                         P2:    P[2]
  4                         P3:    P[3]
  5 Destination position    Q1:
  6                         Q2:
  7                         Q3:
```

- **To record a position,** jog the robot to the destination position (Q1). Press and hold in the SHIFT key and press F5, RECORD.
- **To specify a previously recorded position register,** press F4, REFER. Type the number of the position register and press ENTER.

```
Input position register number:
```

- **To clear a position setting** , press NEXT, >, and then press F1, CLEAR. At the prompt "Clear all data," press F4, YES, to confirm.

f. Record or specify all the destination positions.

Note Pressing F2, EXECUTE, will cause the positions you have selected to be mirrored and **will not** cause robot motion.

15. If you selected FRAME as the Image Shift Method, you will see a screen similar to the following:

```
MIRROR IMAGE SHIFT
  Shift Frame Settings

1 Mirror Frame:               WORLD
2   Frame No. : (not used)    *****
3 Mirror Plane:                  XY
```

Move the cursor to Mirror Frame and press F4, [CHOICE]. Select the frame you want to use as the mirror frame and press ENTER.

16. If you selected USER or JOG, move the cursor to Frame No. and type the number of the existing frame you want to use.

17. Move the cursor to Mirror Plane and press F4, [CHOICE]. Select the plane you want to use and press ENTER.

18. **To mirror image the program,** press F2, EXECUTE.

```
Execute transform ok?
```

- **To execute the mirror image shift** , press F4, YES.
- **If you do not want to execute the mirror image shift** , press F5, NO.

19. Wait until software has finished processing the mirror image.

```
-PROCESSING-
```

Some positions in your program might not be able to be mirrored . When this happens, the mirror image software keeps the position in the program at the exact location and orientation it was in before the mirror image was executed. To correct this, you must reteach the position manually.

If this happens while your mirror image program is processing, you will see a message similar to the following.

```
Select P[1] : J6 angle (deg -234)
deg -234 deg 486  *Uninit*          QUIT
```

This message will be displayed for each position that cannot be mirrored. Note the position number; P[1] in this example.

To continue, press F5, QUIT. This will allow the mirror image program to continue processing. Continue noting the position number and axis for each position that did not mirror correctly.



Warning

Do not attempt to move the robot to a position that was not mirrored correctly; otherwise, you could injure personnel or damage equipment.

When the mirror image is complete, you must reteach each position that did not mirror correctly.

10.19 ERROR RECOVERY (OPTION)

10.19.1 Overview

A robot program can stop executing during production as a result of various alarms. For example, if a HOLD or EMERGENCY STOP input is detected, the robot stops moving and processing the application such as arc welding, dispensing, or material handling. In arc welding, for example, you might want to clean the torch and cut the wire before you resume the paused welding program. You can use *Error Recovery* to perform these operations automatically and eliminate the time required to jog the robot to and from a manual repair station.

This section is organized as follows:

- Overview - Resume Program, Maintenance Program, and operation
- Features
- Limitations
- I/O interface
- Setup
  - — Alarm code monitoring
  - — Digital input alarms
- Programming
- Testing
- Manual function
- I/O timing sequence

10.19.2 Features

Table 10–37 summarizes the features available in the Error Recovery option.

Table 10–37. Error Recovery Features

FEATURE	DESCRIPTION
Alarm Code Monitoring	This feature specifies whether Error Recovery can execute for all alarms or for only a set of specific alarms. Refer to Procedure 10-33 .
Automatic Start	This feature permits Error Recovery to execute the recovery sequence without waiting for the START input. The fault output is also suppressed. Refer to the I/O timing diagrams in Section 10.19.9 . Typically, when an alarm is defined using the Alarm Code Monitoring feature and an alarm occurs, the program is paused with the output of a fault signal. After the first START signal input is received, the Resume Program is executed. After the completion of the Resume Program execution, a second START signal input is received and the paused original program is resumed. If the Automatic Start feature is enabled, when the defined alarm occurs, the Resume Program is executed automatically without the FAULT signal output and without stopping the robot. After the completion of the Resume Program execution, the original program is resumed automatically. Therefore, if the Automatic Start feature is enabled, you do not need to input two START signals.
Resume Programs	This feature allows user-programmed error recovery at the point of the error.
Maintenance Programs	This feature allows user-programmed error recovery after exiting the original program.
Program Exit and Entry	This feature specifies whether Error Recovery automatically **exits** and **enters** a user program when using a Maintenance Program.
Teach Pendant Program Instructions	You use teach pendant instructions to define the names of the resume programs and maintenance programs in your teach pendant program.
Error Recovery Status DO	You can define a digital output signal to allow an external control device (such as a PLC) to monitor the recovery process. Refer to the I/O timing diagrams in Section 10.19.9 .
Error Recovery Approval DI	You can define a digital input signal to allow an external control device (such as a PLC) to approve or disapprove the execution of the recovery program. Refer to the I/O timing diagrams in Section 10.19.9 .
Process Disable *	This feature specifies whether Error Recovery disables the processing of the application during resume program execution, Exit and Entry paths, and maintenance program execution.
Dry Run Speeds	You can change the speed of recovery motions by using dry run speeds during Exit and Entry moves.

Table 10–37. Error Recovery Features (Cont'd)

FEATURE	DESCRIPTION
Test Mode	You can test Error Recovery execution from the teach pendant using the MANUAL FUNCTIONS screen.
Z Offset (DispenseTool only)	You can specify offset value for ENTRY Mode. All teach point during ENTRY mode will be shifted by this value in z direction of UTOOL frame. The offset value can be up to 50 mm.
Alarm Recovery Screen (SpotTool+ only)	SpotTool+ products include an Alarm Recovery screen. This screen allows you to make recovery choices from the teach pendant. This screen is available only in SpotTool+ products.

*This item does not apply to ArcTool.

Note Use Error Recovery only when the teach pendant is disabled. When the teach pendant is enabled, Error Recovery programs can be executed only from the Manual Function screen. Refer to Section 10.19.8 .

10.19.3 Limitations

Error Recovery is DISABLED when any of the following functions are installed:

- Line tracking
- Soft float
- Continuous turn
- MIG EYE

In addition, Error Recovery has the following limitations:

- Maintenance Program style error recovery is not supported for Coordinated Motion, TAST with Multi-Pass/RPM, or AVC.
- Error recovery is automatically disabled during touch sense searching and re-enabled for all other parts of the teach pendant program.
- Single step execution is disabled during Resume Program execution. Single step mode is available only for the execution of the original program. The single step LED on the teach pendant shows the status for the original program execution.
- If the original program is paused after the RESUME_PROG instruction and then the operator moves the cursor to another line, the Resume Program is not executed at the next program execution.

- The status of the Resume Program execution is not displayed at the monitor screen in the program EDIT screen.
- The status line does not indicate when the Resume Program is executing.
- For a multi-tasking system, when the alarm code monitor feature is disabled and the approval DI is not defined, if the HOLD key is pressed, both parent and child task are paused.
- The following limitations apply to z offset, a DispenseTool feature:
  - — The z offset feature modifies taught points during ENTRY move. This feature does not guarantee the specified offset of the whole path.
  - — The z offset is not applied for INCREMENTAL motion. If the system detects INCREMENTAL motion during ENTRY move and the z offset is non-zero, MOTN-341 "No Z offset for INC motion" warning is displayed.
  - — When the remote TCP option is used, the z offset is applied in the z direction of RTCP frame. The z offset is applied differently for RTCP and non-RTCP motion.
- MAINTENANCE PROG (fast fault) error recovery is not supported for TAST and AVC. An error will be posted if this recovery method is enabled and a monitor alarm occurs in a tracking program.



Caution

When the Remote TCP option is used, a z-offset is applied in the RTCP frame (User Frame), while a z-offset is applied in the Tool Frame for non-RTCP motions. To prevent unexpected motion or damage, do not use circular motion at the transition between RTCP and non-RTCP motion and do not use a z-offset that is large; otherwise, you could injure personnel or damage equipment.

  - — You might not see the z offset item on the Error Recovery screen. This item is displayed only when it is available.

10.19.4 I/O Interface

The Error Recovery sequence can be monitored and controlled remotely using digital I/O. The following I/O signals are available for use with Error Recovery.

- Approval DI
- Incomplete End DO
- Reset DI
- Status DO
- Maint DO

These signals can be assigned in the Error Recovery SETUP screen described in Section 10.19.5 . Refer Section 10.19.9 for more information on the I/O timing sequence.

Approval DI

If this input is defined (not zero), it is checked before error recovery is executed.

- If Approval DI is ON, error recovery is approved, and Resume Programs or Maintenance Programs are executed at the appropriate times.
- If Approval DI is OFF, error recovery is not approved and Resume Programs or Maintenance Programs are not executed.

Incomplete End DO

When an Error Recovery program is aborted before its normal completion, the Incomplete End DO is turned ON. This DO is turned OFF at the next program execution. The Incomplete End DO is not set if the original program is aborted.

Check the status of this digital output signal before you input the START signal. If this signal is ON, confirm the current robot position. If an interference exists between the current robot position and the paused position of the original program, jog the robot to the position near the paused position before you input the START signal.

Reset DI

When the Incomplete End DO is used as a condition for a start input in the PLC, you need to turn off the Incomplete End DO remotely. When the Reset DI is input, the Incomplete End DO is turned OFF. After the operator performs the appropriate operation (for example, moves the robot to the position near the paused position of the original program), input this DI signal.

Status DO and Maint DO

The Status DO and Maint DO signals are provided to indicate whether an error recovery program will execute at the next START input signal.

- When the Status DO output signal is ON, it indicates that a Resume Program will execute at the next START input.
- When the Maint DO output signal is ON, it indicates that a Maintenance Program will execute at the next START input.
- When both the Status DO output signal is OFF and the Maint DO input signal is OFF, the original program will execute at the next START input.

Warning

The Status DO and Maint DO signals are useful; without them, it is difficult to know which program will execute when resuming a paused program. For example, if you have set up Alarm Code Monitoring, only *some* errors will invoke Error Recovery, *not all* . As another example, if you have set up the Approval DI, its state dictates which program will execute.

Note You must disable single step mode before you begin error recovery. If single step mode is enabled during error recovery, it is ignored until the recovery sequence is completed.

10.19.5 Setup

You set up Error Recovery for either Resume Program or Maintenance Program execution using the Error Recovery Setup screen.

The Error Recovery Setup screen is shown in Figure 10–70 . The items on this screen are listed and described in Table 10–38 .

Figure 10–70. Error Recovery Setup Screen

```
Error Recovery Set
                                        1/12
Error recovery function common setup
1  Error recovery function:     DISABLED
2  Approval DI index No.:             0
3  Incomplete end DO index No.:       0
4  Reset DI index No.:                0
5  Automatic start feature:     DISABLED
RESUME PROGRAM type recovery
6  Status DO index No.:               0
7  Auto start Max count:              2
8  Auto start Max count R[ ]:          0
MAINTENANCE PROGRAM type recovery
9  Fast exit/entry feature:     DISABLED
10  Dry run exit/entry:         DISABLED
11  Maintenance program:        ********
12  MAINT DO index No.:                0
13  Z offset in ENTRY mode:            0.00
```

Table 10–38. Error Recovery Setup Items

ITEM	DESCRIPTION
Error Recovery Function Common Setup	
Error Recovery Function	This item enables and disables Error Recovery.
Approval DI Index No.*	This item defines a digital input for approval of error recovery program execution.
Incomplete End DO Index No.*	This item defines a digital output to indicate that an error recovery program has been aborted before completion.
Reset DI Index No.*	This item defines a digital output for resetting the "Incomplete end DO."
Automatic Start Feature	This item enables and disables the automatic start feature.
RESUME PROGRAM Type Recovery	
Status DO Index No.	This item defines a digital output to indicate if a Resume Program or the original program will execute with the next start input.
Auto Start Max Count	This item defines the number of times Error Recovery is attempted for a given fault at the same location.
Auto Start Max Count R[ ]	This item defines the register number used for counting the number of times the error recovery program is started automatically.
MAINTENANCE PROGRAM Type Recovery	
Fast Exit/Entry Feature	This item enables and disables Maintenance Program error recovery.
Dry Run Exit/Entry	This item enables and disables the use of dry run speeds during exit and entry operations.
Maintenance Program	This item defines the name of a default Maintenance Program to be run when a teach pendant program has not executed a MAINT_PROG instruction.
MAINT DO Index No. *	This item defines a digital output to indicate if a Maintenance Program or the original program will execute with the next start input.
Z Offset in ENTRY Mode (for DispenseTool only)	This item defines Z offset value which is applied for ENTRY move.

***** Set this to zero if you do not want to use this feature.

Use Procedure 10-32 to set up Error Recovery items.

Procedure 10-32 Setting Up Error Recovery Items

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].

4. Select Err recovery. You will see a screen similar to the following.

Note Items 9-13 are displayed only if the system variable $RSMFST_SV.$ffast_dsp = TRUE. If this variable is FALSE, these items are not displayed and Maintenance Program recovery is disabled.

```
Error Recovery Set
                                        1/12
Error recovery function common setup
1  Error recovery function:     DISABLED
2  Approval DI index No.:             0
3  Incomplete end DO index No.:       0
4  Reset DI index No.:                0
5  Automatic start feature:     DISABLED
RESUME PROGRAM type recovery
6  Status DO index No.:               0
7  Auto start Max count:              2
8  Auto start Max count R[ ]:          0
MAINTENANCE PROGRAM type recovery
9  Fast exit/entry feature:     DISABLED
10  Dry run exit/entry:         DISABLED
11  Maintenance program:        ********
12  MAINT DO index No.:               0
13  Z offset in ENTRY mode:           0.00
```

5. Move the cursor to each item and set it as desired. Refer to Table 10–38 .

Note The default configuration of Error Recovery assumes that control is from the UOP. Error recovery can be configured to execute from the standard operator panel by setting the system variable $RSMPRG_SV.$chk_remote = FALSE.



Caution

If you configure error recovery to execute from the operator panel, there is no way to know that RESUME_PROG will be executed at the next start input unless you view the Error Recovery Status screen in the MANUAL FUNCTIONS screen.

Alarm Code Monitoring

You can set up Error Recovery to execute after either **all** errors or after a specific set of user-specified errors. Use the ALARM function key (Procedure 10-33) to define a list of specific errors. If you do not define any specific errors, then all PAUSE severity errors will start Error Recovery program execution.

When you have specified alarms to be monitored, and a fault occurs that is not on the alarm code monitor list, the original program will pause and the Resume Program will not be executed.

The default maximum number of alarms that can be monitored is 10. You can change this number by setting the value of $RSMPRG_SV.$num_alarm and turning off the controller and then turning it on. The maximum value is 32.

Note If specific "Monitored alarm code" faults are not defined (they are all zero) and error recovery is enabled (Approval DI is ON), then all PAUSE severity errors will cause resume program execution at a START signal.

Use Procedure 10-33 to set up alarm code monitoring.

Procedure 10-33 Setting Up Alarm Code Monitoring

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Err recovery. You will see a screen similar to the following.

```
Error Recovery Set
                                        1/12
Error recovery function common setup
1  Error recovery function:     DISABLED
2  Approval DI index No.:             0
3  Incomplete end DO index No.:       0
4  Reset DI index No.:                0
5  Automatic start feature:     DISABLED
RESUME PROGRAM type recovery
6  Status DO index No.:               0
7  Auto start Max count:              2
8  Auto start Max count R[ ]:          0
MAINTENANCE PROGRAM type recovery
9  Fast exit/entry feature:     DISABLED
10  Dry run exit/entry:         DISABLED
11  Maintenance program:        ********
12  MAINT DO index No.:                0
13  Z offset in ENTRY mode:            0.00
```

5. Press F2, ALARM. Define the alarm code to be monitored. See the following screen for an example.

```
Error Recovery Setup
                                          1/10
  1   Monitored alarm code:             53013
  2   Monitored alarm code:             53018
  3   Monitored alarm code:             12278
  4   Monitored alarm code:                 0
  5   Monitored alarm code:                 0
  6   Monitored alarm code:                 0
  7   Monitored alarm code:                 0
  8   Monitored alarm code:                 0
  9   Monitored alarm code:                 0
```

The alarm code should be defined as "Alarm code ID + Alarm number." The alarm code ID indicates the kind of alarm. For example, the "Arc start failed" alarm is represented as follows:

```
ARC
   -
013
   Arc Start Failed   =
53

013


ID(53)    Number                                  ID Number
```

where on the lefthand side of the =, ARC is ID(53) and 013 is the Number, and on the right hand side of the =, 53 is the ID and 013 is the Number.

Refer to the *Error Code Manual* for alarm number definitions.

53013 means "ARC-013 Arc Start failed".

53018 means "ARC-018 Lost arc detect".

12278 is an INTP error that can be used to monitor user alarms defined on the User Alarm Setup screen and also on the DI_ALARM screen of Error Recovery Setup

6. **To display help information** , press F5, HELP. You will see a screen similar to the following.

```
Error Recovery Setup
 HELP
Typical alarm code IDs are specified as follows.
    PROG: 3,     SRVO:11,    INTP:12
    PRIO:13,     MOTN:15,    SPOT:23
    SYST:24,     PALT:26,    LASR:50
    SEAL:51,     ARC :53,    MACR:57
    SENS:58,     COMP:59
```

Note To select the alarms to monitor, refer to the *Error Code Manual*.

Digital Input Alarms

The Error Recovery option allows you to define digital input signals that will generate user alarms. These user alarms can be monitored as error code 12278, as illustrated in Procedure 10-33 . Use Procedure 10-34 to set up digital input alarms.

You set user alarm information on the Setting User Alarm screen, shown in Figure 10–71 . Refer to Section 3.7 for more information on User Alarm setup.

Figure 10–71. Setting User Alarm Screen

```
Setting/User Alarm
                                          1/10
 Alarm No.         User Message
    [1]:   [                            ]
    [2]:   [                            ]
    [3]:   [                            ]
    [4]:   [                            ]
    [5]:   [                            ]
    [6]:   [                            ]
    [7]:   [                            ]
    [8]:   [                            ]
    [9]:   [                            ]
```

Procedure 10-34 Setting Up Digital Input Alarms

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].

4. Select Err recovery. You will see a screen similar to the following.

```
Error Recovery Set
                                          1/12
Error recovery function common setup
1  Error recovery function:     DISABLED
2  Approval DI index No.:             0
3  Incomplete end DO index No.:       0
4  Reset DI index No.:                0
5  Automatic start feature:     DISABLED
RESUME PROGRAM type recovery
6  Status DO index No.:               0
7  Auto start Max count:              2
8  Auto start Max count R[ ]:          0
MAINTENANCE PROGRAM type recovery
9  Fast exit/entry feature:     DISABLED
10  Dry run exit/entry:         DISABLED
11  Maintenance program:        ********
12  MAINT DO index No.:               0
13  Z offset in ENTRY mode:           0.00
```

5. Press F3, DI_ALARM. See the following screen for an example.

```
Error Recovery Set SW
                                          1/3
     UALM     Severity   Type      Value
 1   [  1]    LOCAL     DI[  1]       ON
 2   [  0]    LOCAL     DI[  0]       ON
 3   [  0]    LOCAL     DI[  0]       ON
```

6. Select and set the items as desired.
7. **To display help information**, press F5, HELP.
8. **When you are finished setting DI_ALARM information** , press F4, DONE, to return to the previous screen.

10.19.6 Programming

You use the following teach pendant instructions to specify appropriate recovery program names when you use Error Recovery:

- Resume Program instructions
- Maintenance Program instructions

Resume Program Instructions

The auto error recovery function executes the resume program defined in the teach pendant program. To define which resume program is used, use the RESUME_PROG instruction. To clear the resume program, use the CLEAR_RESUME_PROG instruction. See Figure 10–72 and Figure 10–73 .

Figure 10–72. RESUME_PROGRAM Instruction

```
RESUME_PROG = resume program name
```

Figure 10–73. CLEAR_RESUME_PROG Instruction

```
CLEAR_RESUME_PROG
```

Figure 10–74 contains an example production program that sets RESUME_PROG to WIRE_CUT, which is shown in Figure 10–75 .

Figure 10–74. WELD.TP Example Program

```
1:  J P[1]  40%  FINE
2:  RESUME_PROG = WIRE_CUT
3:  L P[2]  300mm/sec  FINE  ARC START[1]
4:  L P[3]  50cm/min  CNT100
5:  L P[4]  50cm/min  FINE  ARC END[2]
6:  CLEAR_RESUME_PROG
7:  L P[5]  300mm/sec FINE
```

Figure 10–75. WIRE_CUT.TP (Resume Program) Example Program

```
1:  L P[10]  50mm/sec  FINE  INC
2:    PR[1]=LPOS
3:  J P[11]  50%  FINE
4:    WO[4] = ON pulse 0.5sec   Feed wire
5:  L P[12]  20mm/sec  FINE
6:    WAIT 0.8sec               Wait for completion
7:  L P[11]  20mm/sec  FINE      of cutting wire
8:  RESUME_PROG = WIRE_CUT2
9:  J PR[1]  50%  FINE
```

Figure 10–74 shows how to define the resume program. The WIRE_CUT program is defined as the resume program in line 2 using the RESUME_PROG instruction. The WIRE_CUT program is cleared

from the resume program in line 6 of WELD.TP using the CLEAR_RESUME_PROG instruction. Therefore, the WIRE_CUT program is available as the resume program only during program lines 3, 4, and 5 in WELD.TP.

When the automatic start feature is enabled and WELD.TP is paused by a monitored alarm and resumed on lines 3, 4, or 5, the WIRE_CUT program is executed as the resume program and the wire is automatically cut using the WIRE_CUT program. In the program WELD.TP, the resume program is not executed after line 6.

Note In the WIRE_CUT.TP program, PR[1] is near the fault position. The INC (incremental) motion option in line 1 puts PR[1] near the fault position by the value of the INC position. In general, the INC position will be a z offset, such as P[10]: 0, 0, 25, 0, 0, 0.

Note In WIRE_CUT.TP, line 8, the RESUME_PROGRAM is redefined to be WIRE_CUT **2** .TP. If another error occurs in WELD.TP after executing the recovery sequence, WIRE_CUT **2** .TP will be executed upon program resume instead of WIRE_CUT.TP.

Warning

If the wrong program is defined as the resume program, the robot will move toward an unexpected place. Be sure to define the correct resume program. Otherwise, you could injure personnel or damage equipment.

Maintenance Program Instructions

To define which maintenance program is used, use the MAINT_PROG instruction. To disable the ability to use the return path, use the RETURN_PATH_DSBL instruction. See Figure 10–76 and Figure 10–77 .

Figure 10–76. MAINT_PROGRAM Instruction

```
MAINT_PROG = maintenance program name
```

Figure 10–77. RETURN_PATH_DSBL Instruction

```
RETURN_PATH_DSBL
```

Figure 10–78 contains an example production program that sets MAINT_PROG to TORCHCLN, which is shown in Figure 10–75 .

Figure 10–78. WELD.TP Example Program

```
 1:  J P[1]  40%  FINE
 2:  MAINT_PROG = TORCHCLN
 3:  L P[2]  300mm/sec  FINE  ARC START[1]
 4:  L P[3]  50cm/min  CNT100
 5:  L P[4]  50cm/min  FINE  ARC END[2]
 6:  L P[5]  300mm/sec FINE
```

If the "RESUME_PROG=" or "MAINT_PROG=" instruction is not used in a program but "Fast exit/entry feature" is ENABLED in the error recovery setup, when a monitored alarm occurs, the MAINT_PROG style of error recovery will execute using the "Maintenance program" defined in the error recovery setup.

If you do not want to use either style of error recovery in a given program, you can use the parameter instruction to disable "Fast exit/entry feature," $RSMFST_SV.$FFAST_ENBL=0. To enable the feature, use the parameter instruction $RSMFST_SV.$FFAST_ENBL=1.

See Figure 10–79 for an example program.

Figure 10–79. Fast Exit/Entry Feature Enabled and Disabled in a Program

```
 1: J P[1] 100% FINE
 2: $RSMFST_SV.$FFAST_ENBL=0  ;
 3: J P[2] 40% FINE
  : Arc Start[1]
 4: L P[3] WELD_SPEED CNT100    ;
 5: L P[4] WELD_SPEED CNT100    ;
 6: L P[5] WELD_SPEED FINE
  : Arc End[1]
 7: $RSMFST_SV.$FFAST_ENBL=1   ;
 8: J P[6] 100% FINE   ;
 9: J P[7] 100% FINE   ;
10: J P[8] 100% FINE   ;
```

Programming Procedure

Use Procedure 10-35 to add Error Recovery instructions to a program.

Procedure 10-35 Adding Error Recovery Instructions to a Program

Conditions

- You are editing a teach pendant program.

Steps

1. Press F1, [INST], to display the list of instructions.
2. Select -- next page -- to display more instructions.
3. Select Program control. You will see a screen similar to the following.

```
Instruction
  1 PAUSE              5 CLEAR_RESUME_PROG
  2 ABORT              6 RETURN_PATH_DSBL
  3 ERROR_PROG         7 MAINT_PROG
  4 RESUME_PROG        8
PROGRAM
```

4. Select RESUME_PROG, CLEAR_RESUME_PROG, RETURN_PATH_DSBL, or MAINT_PROG.

10.19.7 Testing

Normally you use Error Recovery when the teach pendant is disabled during production operation. However, when you define a recovery program or test a production program, you might want to execute the recovery program even though all conditions for execution have not been met.

You can test error recovery with the teach pendant enabled from the Auto Error Recovery Manual Function screen when you select **TP_TEST** as the operation mode. Refer to the **Operation mode** item in Section 10.19.8 .

10.19.8 Error Recovery Manual Function

You can use the Manual Function screen to do the following:

- Display the status of error recovery status DO
- Display the resume program name defined by the original program
- Select the operation mode
- Monitor the conditions related to the status of the error recovery status DO

Table 10–39 lists and describes the items on the Manual Function screen.

Refer to Table 10–40 for a description of the items on the Manual Function DETAIL screen.

Use Procedure 10-36 to perform manual Error Recovery.

Table 10–39. Auto Error Recovery Manual Function Screen Items

ITEMS	DESCRIPTION
Error recovery DO status	This item displays the status of the error recovery status DO regardless of whether the error recovery status DO is defined in the Auto Error Recovery Setup screen. **Note** Status DO refers to an internal signal, not a digital output signal. When a digital output is configured, this internal signal will be reflected in the digital output.
Defined resume program	This item displays the resume program name defined by the original program.
Operation mode Default: AUTO	This item has the following three modes. The default mode is **AUTO** and it is automatically changed to **AUTO** when you exit from this screen. • AUTO **— This mode should be selected when the teach pendant is disabled.** When this mode is selected, the Resume Program is executed according to the status of the alarm code feature and the error recovery approval DI feature. When this mode is selected and the teach pendant is enabled, the resume program is not executed but the original program is executed when SHIFT FWD is pressed. • NOEXEC— When this mode is selected, the error recovery status DO is always turned off regardless of whether the teach pendant is enabled. This means that the Resume Program is not executed by the next program execution. • TP_TEST **— This mode should be selected when the teach pendant is enabled.** When this mode is selected and the teach pendant is enabled, even if the alarm code feature and the error recovery approval DI feature are not satisfied, the error recovery status DO is turned on. This means the Resume Program is always executed by the next program execution.

Manual Function Detail Screen

In the Manual Function DETAIL screen you can monitor recovery DO status, recovery program name, and operation mode related to the status of the error recovery status DO.

When F2, DETAIL, is pressed on the Auto Error Recovery Manual Function screen, the conditions related to the error recovery status DO are displayed.

When all elements are Yes or None (not used), the error recovery status DO is turned ON.

When the error recovery status DO is OFF and you are not sure of the cause, check the information on this screen.

Table 10–40. Auto Error Recovery Manual Function Detail Screen Items

ITEMS	DESCRIPTION
Auto error recovery enabled	This item shows whether the Error Recovery function is enabled in the auto error recovery setup screen.
PAUSED & resume prog incomp	This item shows the following: • The original program exists. • The original program is paused. • The execution of the resume program selected by the original program has not been completed.
Program has motion group	This item shows whether the original program has motion control.
Not in single step mode	This item shows whether the single step mode is disabled. The single step LED on the teach pendant specifies the status of single step for the original program ($TP_DEFPROG). When the Resume Program is paused and then single step LED is turned on, the error recovery DO remains ON because single step for the Resume Program is disabled.
Resume program is defined	This item shows whether the resume program is defined by the original program.
Mode is (xxxxxx)	This item shows whether the operation mode is the desired one for this current situation. If the teach pendant is disabled, AUTO is displayed in the field "xxxxxx". If the teach pendant is enabled, TP_TEST is displayed in the field.
Approval DI is ON	This item shows whether the status of the error recovery approval DI is ON. If the index of this DI is not defined or the teach pendant is enabled, "None" is displayed.
Defined alarm occurs	This item shows that the defined alarm has occurred and that the original program has been paused by the alarm, if the alarms are defined in the setup screen. If the alarm code is not defined or the teach pendant is enabled, "None" is displayed.
Remote when $RMT_MASTER is 0	This item shows whether the remote condition is satisfied. This feature is available only when the teach pendant is disabled and **$RMT_MASTER** is 0 and **$RSMPRG_SV.$CHK_REMOTE** is TRUE. If you want to remove the remote condition, you can set **$RSMPRG_SV.$CHK_REMOTE** to FALSE **.**
No disabled options	The Error Recovery function is incompatible with certain options. It shows whether an incompatible option exists in the software.
User condition param enable	This item shows whether the user condition parameter (system variable) is TRUE. The default is TRUE. This system variable is **$AUTORCV_ENB**.

Procedure 10-36 Manual Operation of Error Recovery

Steps

1. Press MENUS.
2. Select MANUAL FCTNS.
3. Press F1, [TYPE].
4. Select Err recovery. You will see a screen similar to the following.

```
Error Recovery MNFC
                                          1/1
 Error recovery DO status:              OFF
 Defined resume program:           WIRE_CUT
 1 Operation mode:                     AUTO
```

5. Select the mode you want to use. You can change operation mode between AUTO, NOEXEC, and TP_TEST by pressing F4, [CHOICE]. During production, this mode should be AUTO.
6. Press F2, DETAIL, and the following information screen for the error recovery status DO is displayed.

```
Error Recovery MNFC
                                          1/11
 1 Auto error recovery enabled:        Yes
 2 PAUSED & resume prog incomp:        Yes
 3 Program has motion group:           Yes
 4 Not in single step mode:            No
 5 Resume program is defined:          Yes
 6 Mode is ( AUTO ):                   Yes
 7 Approval DI is ON:                  None
 8 Defined alarm occurs:               Yes
 9 Remote when $RMT_MASTER is 0:       Yes
10 No disabled options:                No
11 User condition param enable:        Yes
```

Note The items are for display only; you cannot change them on this screen.

10.19.9 I/O Timing Sequence

See Figure 10–80 through Figure 10–84 for error recovery timing diagrams.

Figure 10–80. Normal Operation Auto Start Mode

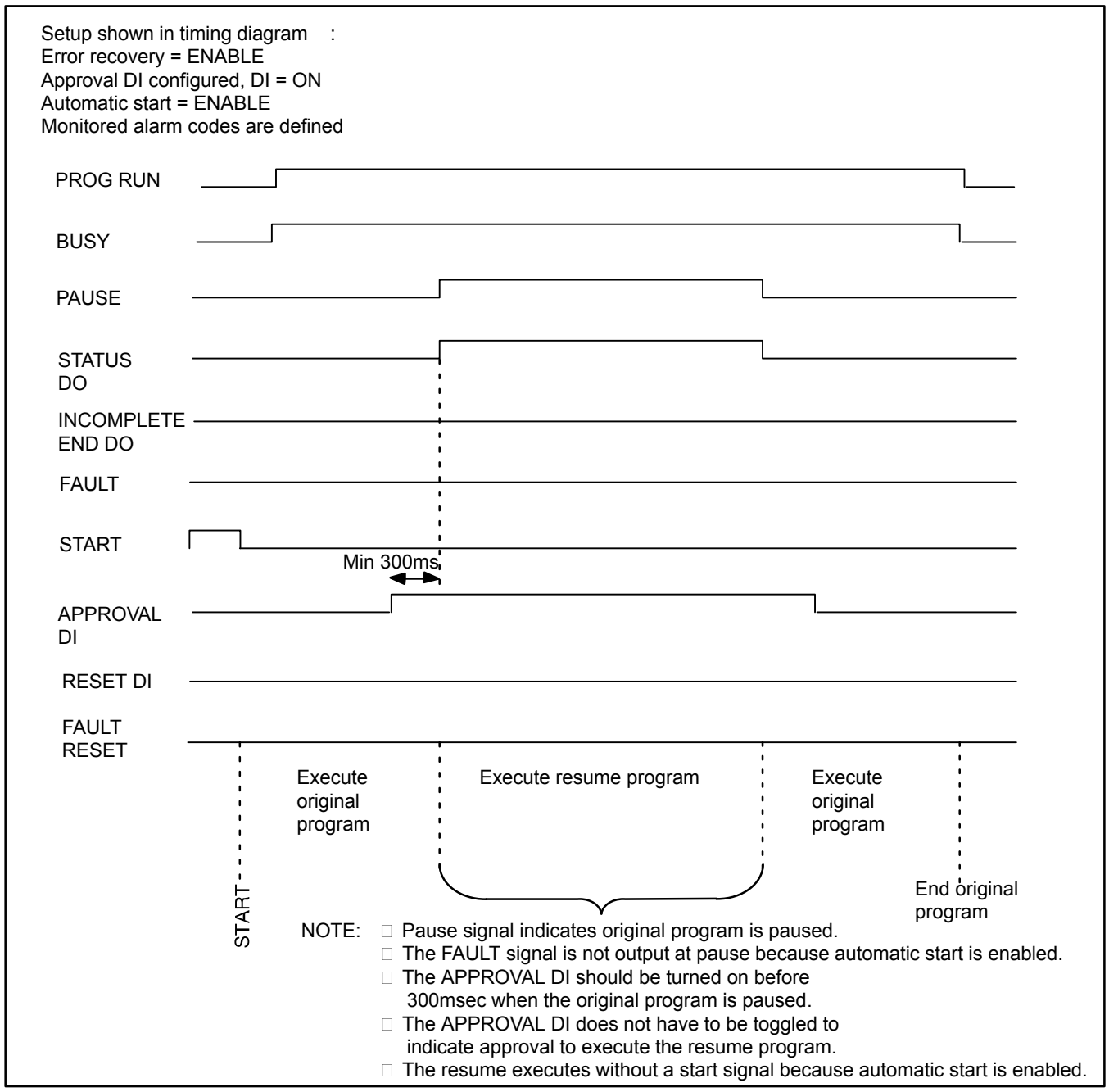


Figure 10–81. Normal Operation without Execution of Resume Program

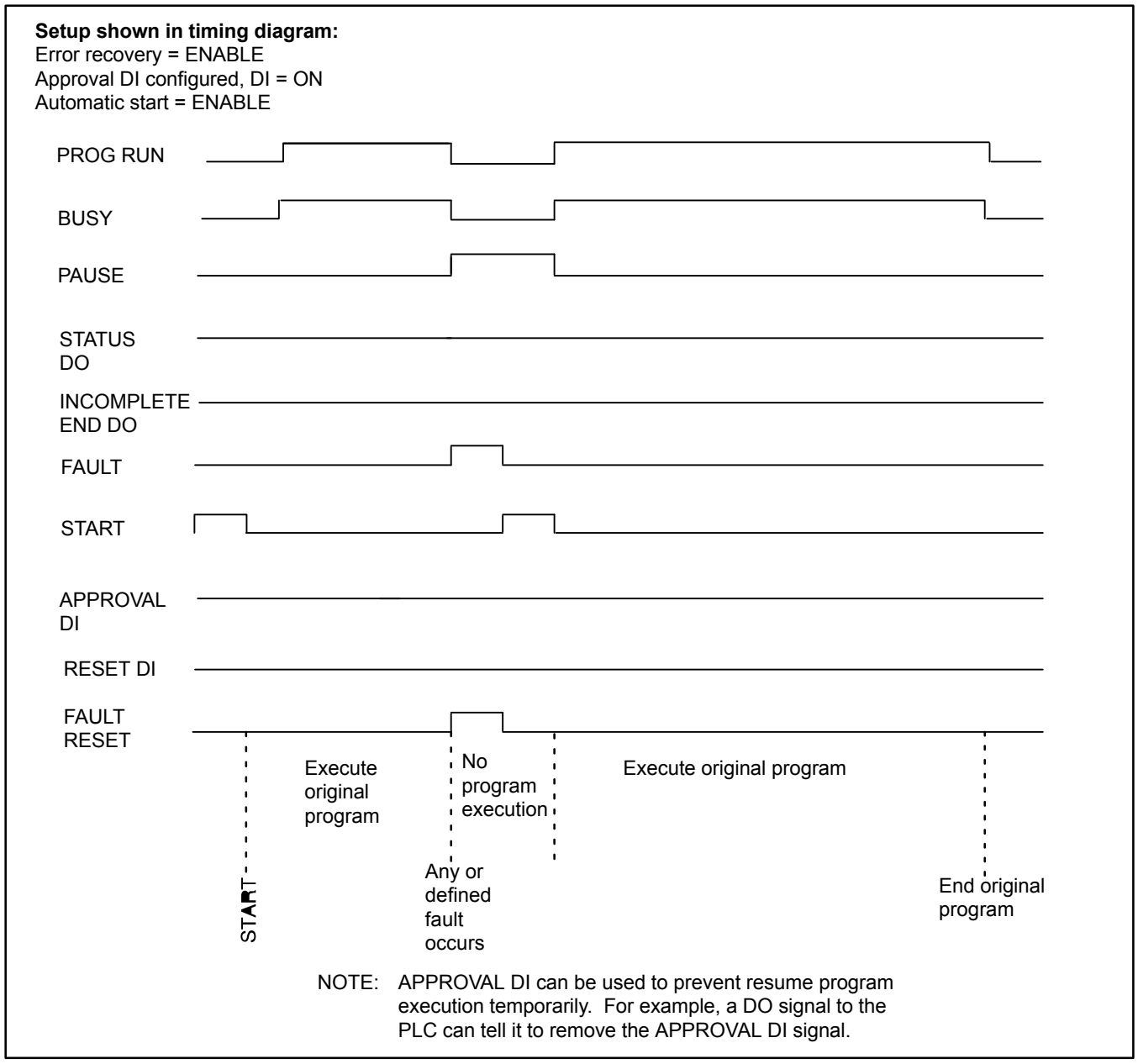


Figure 10–82. Resume Program Aborted

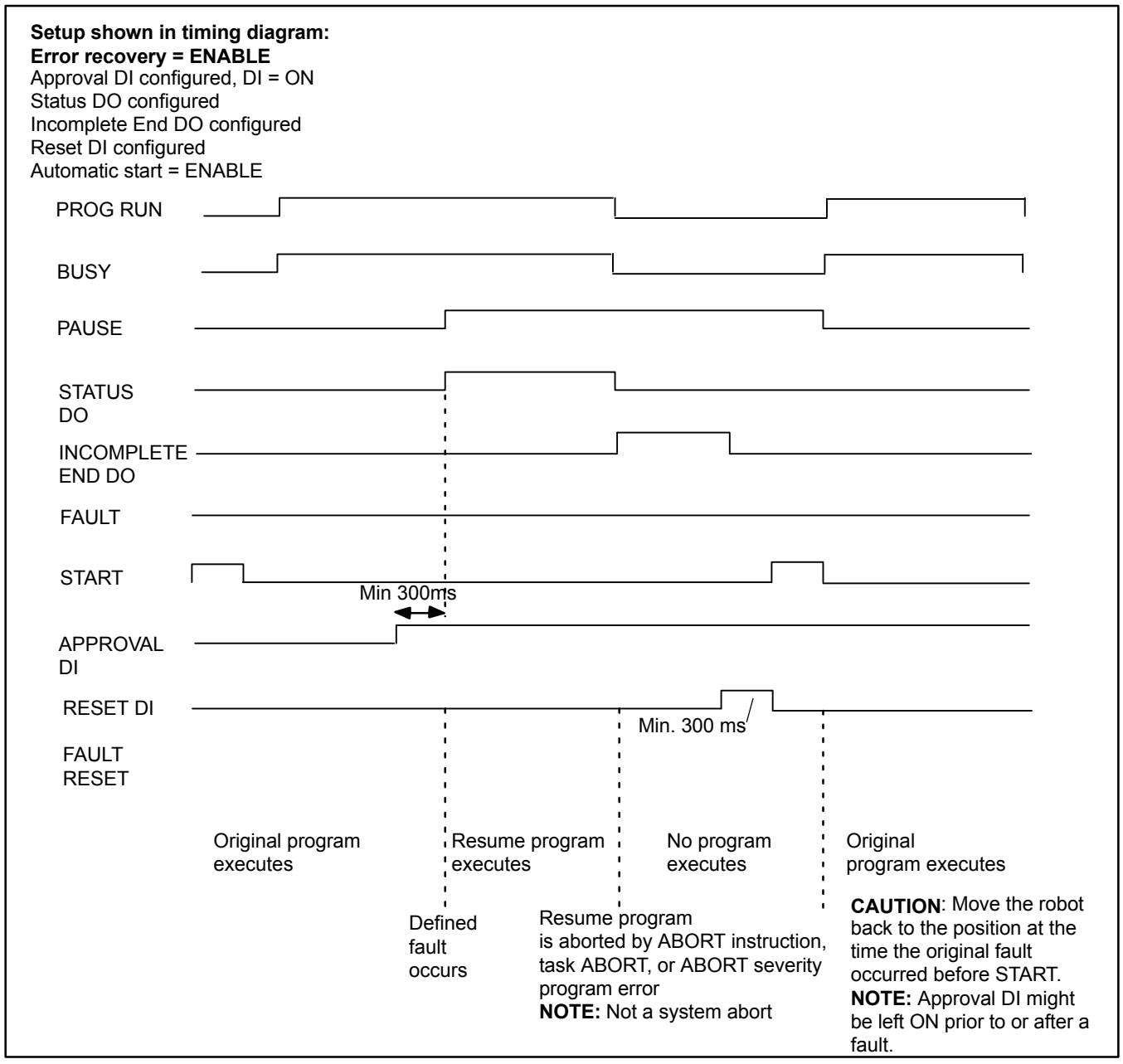


Figure 10–83. Normal Operation (Automatic Start DISABLED)

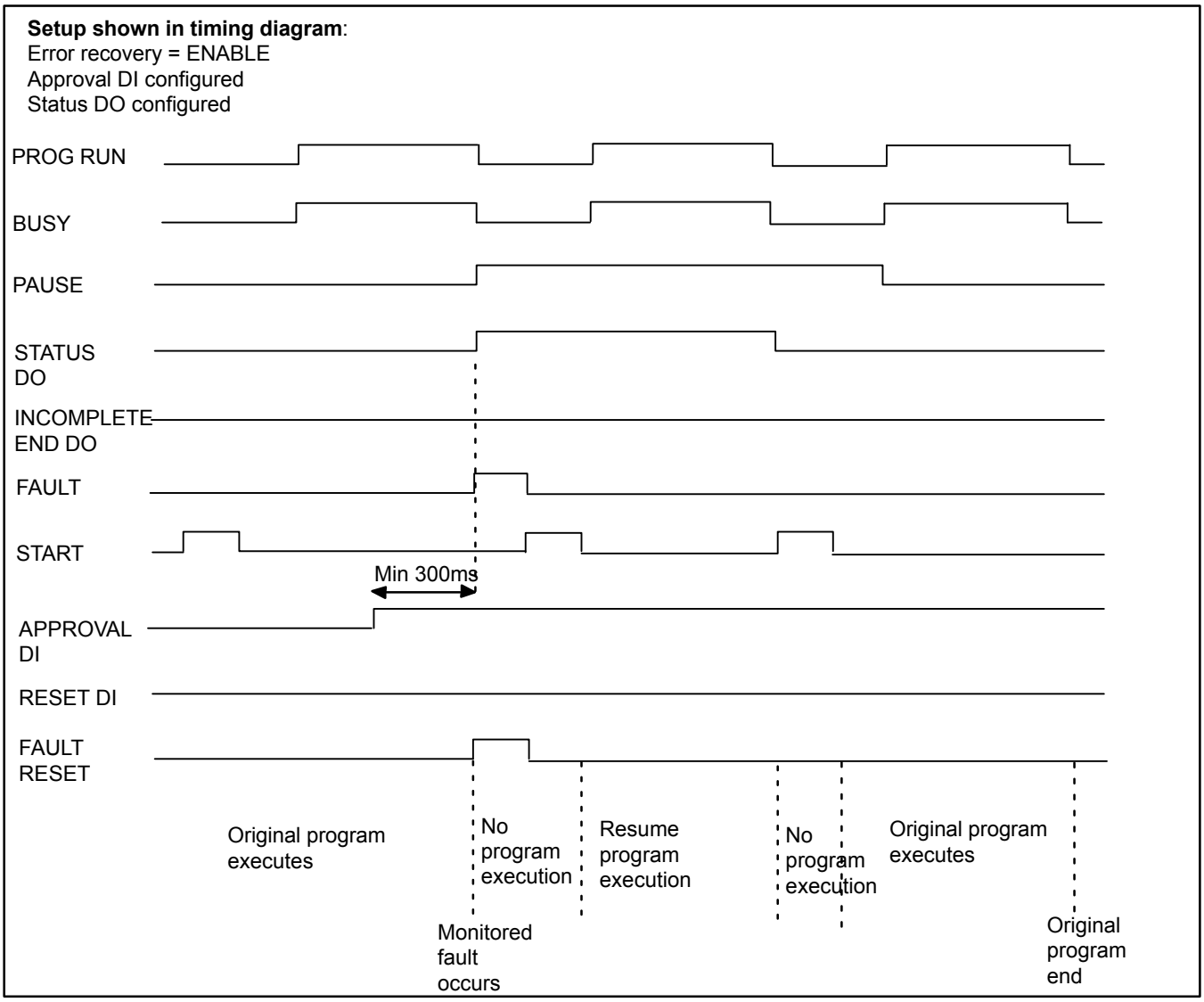
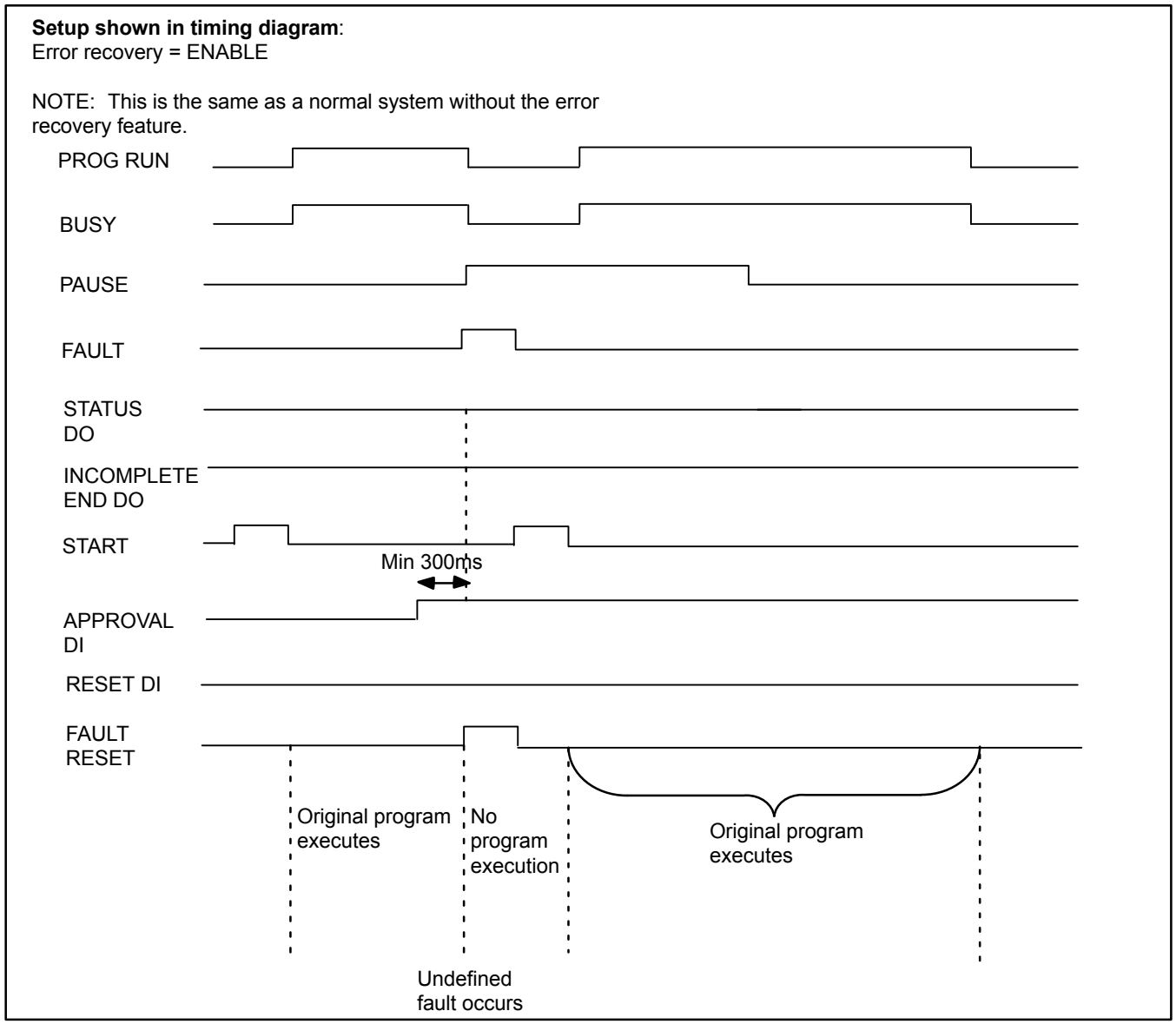


Figure 10–84. Auto Mode When an Undefined Alarm Occurs



10.20 EXECUTING MULTIPLE PROGRAMS (MULTI-TASKING)

10.20.1 Overview

Multi-tasking allows more than one program to run on the controller on a time-sharing basis, so that multiple programs appear to run simultaneously. The maximum number of user programs, or *tasks* , that can be executed simultaneously is four. The default number of simultaneous tasks is one.

To increase the number of user programs that can be executed simultaneously, perform a Controlled start and select the PROGRAM SETUP option from the Controlled start menus.

Refer to Section C.1.5 for information on performing a Controlled start.

You can execute multiple programs in the following ways:

- SOP cycle start, Section 10.20.5
- Program number select (PNS), Section 10.20.6 This item is only for Arctool, DispenseTool, HandlingTool, and KAREL applications
- RUN program instruction, Section 10.20.7
- Single step, Section 10.20.8

10.20.2 Guidelines

Use the guidelines in this section when you write a program for multi-tasking and when you execute multiple programs.

Writing a Program for Multi-Tasking

- Make sure all of the programs involved in the multi-tasking (up to four) each use a **different motion group** . Programs that are executed at the same time cannot use the same motion group.

  You specify the motion group for a program in the program header information. Refer to Section 6.2 .
- Use the **ignore pause** program attribute for programs you do not want to be paused by an error, by a command (such as the HOLD or EMERGENCY STOP button), or by enabling the teach pendant. Programs that use the ignore pause attribute **cannot** have a motion group specified. This means that these programs cannot contain any motion instructions.

  For example, if you have a program that monitors I/O signals, which must execute continuously regardless of external events, you must specify the ignore pause attribute.

  Ignore pause behaves differently during single step execution. Refer to Section 10.20.8 for more information.

  You specify whether to use ignore pause for a program in the program header information. Refer to Section 6.2.11 .

Executing Multiple Programs

The following restrictions apply to executing multiple programs:

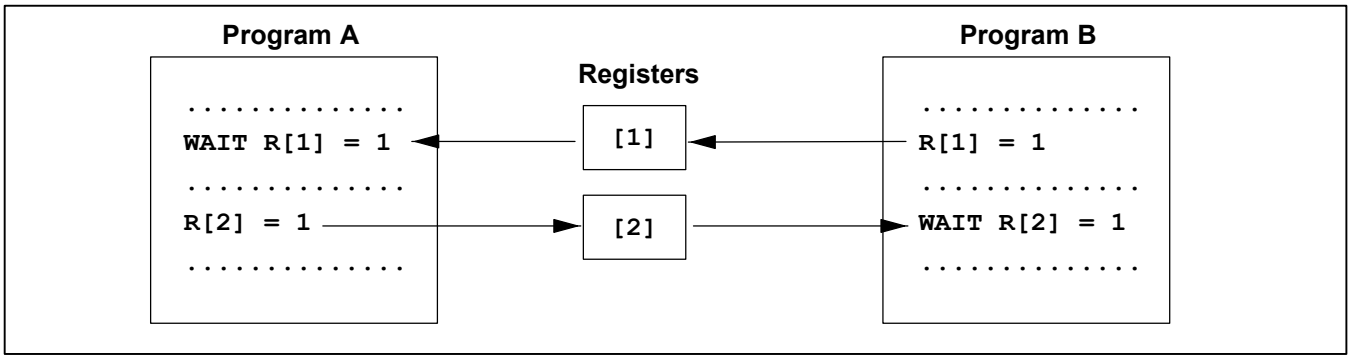
- Up to four programs can be run at a time.

- You cannot execute a program if that program is currently running or paused.
- The programs you run using multi-tasking cannot use the same motion group.
- If you run a program continuously using a PLC, you must enter enough delay in the execution loop. If you do not use any delay, this program can lock other program execution.

10.20.3 Synchronizing the Execution of Multiple Programs

To synchronize the execution of two programs, use register instructions within the two programs. Figure 10–85 shows an example of register instructions used to synchronize the execution of two programs.

Figure 10–85. Using Register Instructions to Synchronize Program Execution



10.20.4 Effect of Multi-Tasking on Dedicated I/O Signals

During multi-tasking program execution, keep in mind the following effects on dedicated I/O signals:

- ***IMS TP input (instantaneous stop)** is enabled for all motion groups.
- The operability of **CMDENBL output (input acceptable)** is checked for all motion groups.
- **SYSR DY output (system ready)** is checked whether servo power for all groups is supplied.

10.20.5 Standard Operator Panel (SOP) Cycle Start Execution

You can start the execution of multiple programs using the standard operator panel (SOP) CYCLE START button. Use Procedure 10-37 to execute multiple programs using the SOP CYCLE START button.

Procedure 10-37 Executing Multiple Programs Using the Standard Operator Panel (SOP) CYCLE START Button

Conditions

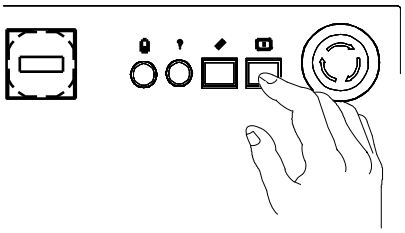
- The programs you will execute at the same time do not use the same motion group.
- You are executing no more than four programs at a time.
- All other conditions related to executing a program using SOP CYCLE START are satisfied.
- You are not executing the same program more than once simultaneously.
- The MODE SELECT switch is set to AUTO.

Steps

1. Select the first program you want to execute using the SELECT menu on the teach pendant.

> **Warning**
>
> **This procedure starts production run. Make sure all safety barriers are in place, all personnel are outside of the workcell, all equipment is in place, and all production conditions have been met before you continue; otherwise, you could injure personnel or damage equipment.**

2. Press the CYCLE START button on the operator panel.



3. Select the next program you want to execute using the SELECT menu on the teach pendant.
4. Press the CYCLE START button on the operator panel.
5. Repeat Step 3 and Step 4 for each program you want to execute.

10.20.6 Program Number Select (PNS) Execution

You can start the execution of multiple programs using program number select (PNS). Use Procedure 10-38 to execute multiple programs using PNS.

Procedure 10-38 Running Multiple Programs Using Program Number Select (PNS)

Conditions

- The programs you will execute at the same time do not use the same motion group such as, the group mask is different for each program.
- You are executing no more than four programs at a time and $SCR.$MAXNUMTASK > 2.
- All other conditions related to executing a program using PNS are satisfied. (Refer to Chapter 7 *TESTING A PROGRAM AND RUNNING PRODUCTION*)
- You are not executing the same program more than once simultaneously.
- The MODE SELECT switch is set to AUTO.
- The REMOTE/LOCAL setup item in the System Configuration Menu is set to REMOTE. Refer to Section 3.15 .



Warning

This procedure starts production run. Make sure all safety barriers are in place, all personnel are outside of the workcell, all equipment is in place, and all production conditions have been met before you continue; otherwise, you could injure personnel or damage equipment.



Warning

If you do not follow this procedure, temporary memory in the controller will be filled. This causes the process CPU to be locked into a busy and running condition. Make sure your PLC logic is correct and does not contain a high rate of production start calls; otherwise, you could injure personnel or damage equipment.

Steps

1. Set the 8 bit PNS input to the number that when added to the base number will determine which program is selected.
2. Strobe the PNSTROBE input. When the controller receives the input signal, the selected program will be displayed on the teach pendant screen. The ACK UOP signal indicates what binary input is being received. This stays ON until a new program is selected.
3. Press the production start button on the user operator panel to start production operation or, if your system uses a PLC, production operations will begin as soon as the PROD_START input is received.
4. Repeat Step 1 through Step 3 for each program you want to execute.

10.20.7 RUN Program Instruction Execution

Use the RUN program within the main program to execute a second, third, or fourth program simultaneously. When you execute a program in which you have added RUN program instructions, the program you specify will execute, and execution of the main program that contains the RUN program instruction will continue at the same time.

Figure 10–86 shows an example of using a RUN program instruction to execute multiple programs.

Figure 10–86. Multi-Tasking Using the RUN Program Instruction

```
Main Program
PROC0001
 1: J P[1] 100% CNT100
 2: J P[2] 100% CNT100
 3: RUN PROC0002
 4: L P[4] 500mm/sec CNT100
 5: RUN PROC0003
 6: L P[5] 500mm/sec CNT100 SE
```

Note In line three the second program will begin executing and in line five the third program will begin executing.

After you have included the RUN program instructions within your main program, execute the program using one of the execution methods available.

10.20.8 Single Step Program Execution

When the main task is executed in single step mode, the subtask is also executed in single step mode. A task in which a RUN instruction is issued is called a main task. A task activated by a RUN instruction within the main task is called a subtask. See Figure 10–87 .

Figure 10–87. Single Step Execution Example

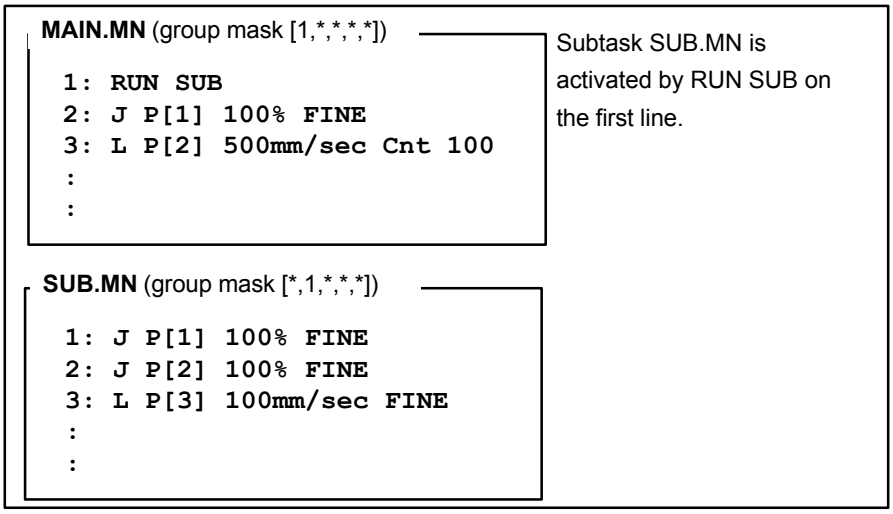


Figure 10–87 , the program MAIN.MN is the main task(motion group 1), and SUB.MN is the subtask (motion group 2), which is activated by a RUN program instruction within MAIN.MN.

When MAIN.MN is executed in single step mode, SUB.MN, activated by a RUN instruction in MAIN.MN, is also executed in single step mode.

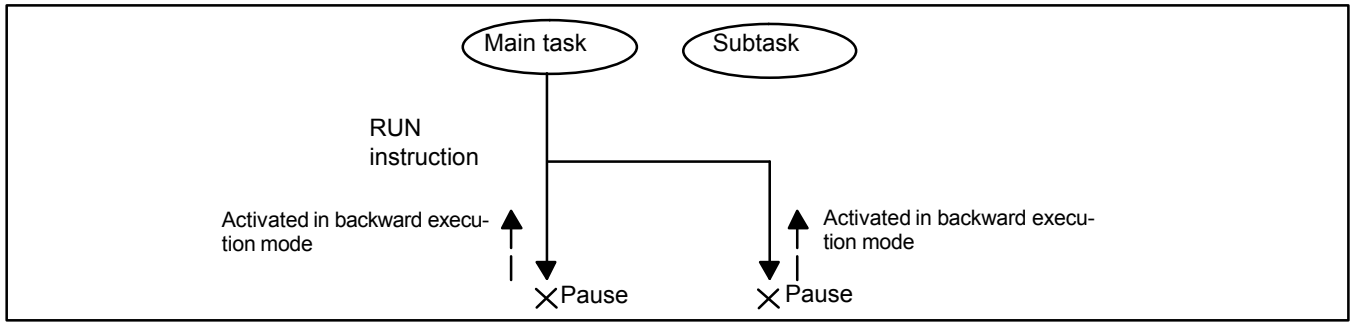
Special considerations for single step execution of multi-tasking programs must also be made in the following areas:

- Backward execution
- Backward execution of the RUN instruction

Backward Execution

When the main task is restarted in backward execution mode, the subtask is activated also in backward execution mode. See Figure 10–88 .

Figure 10–88. Single Step Backward Execution

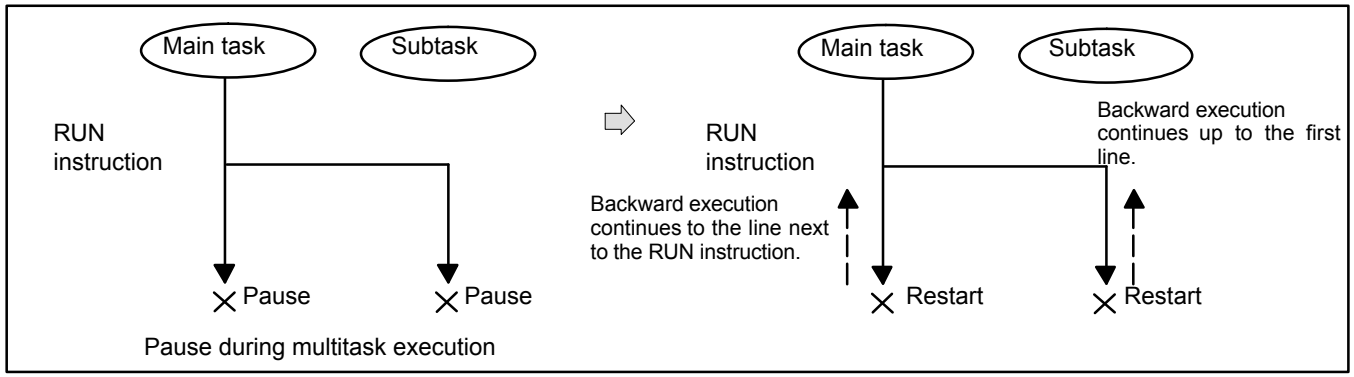


Backward Execution of a RUN Instruction

If a RUN instruction is encountered during backward execution, it is executed in that mode. After backward execution of the RUN instruction, the cursor moves to the instruction next to the RUN instruction. If you want to continue backward execution after the RUN instruction, move the cursor to the instruction before the RUN instruction manually, then specify the backward mode again.

In Figure 10–89 , using backward execution can easily return both main task and subtask to the states they were in immediately after the RUN instruction was executed.

Figure 10–89. Backward Execution of a RUN Instruction Example



In this example, a subtask is first activated by the RUN instruction in the main task. Both tasks are then caused to pause for any reason during multi-tasking execution.

When the program is restarted from a pause during backward execution, the main task is executed to the line next to the RUN instruction in backward mode. Backward execution will not continue any further. However, the subtask is executed to the first line in backward mode. As a result:

- The main task returns to the line next to the RUN instruction.
- The subtask returns to the first line.

This is equivalent to the state immediately after the RUN instruction is executed.

10.21 EXTENDED USER FRAMES

The Extended User Frames option (R696) enables you to increase the number of available tool frames from 10 to 29 and the number of user frames from 9 to 61.

In addition to increasing the number of available frames, this feature has minimal impact on the programming and operating procedures of the robot.

10.21.1 Configuring the Number of Frames

The controller must be in controlled start mode to configure the number of frames.

Set $SCR.$MAXNUMUFRAM to the number of user frames you wish to have. The limit is 61 with the Extended user frames option and 9 without it.

Set $SCR.$MAXNUMUTOOL to the number of tool frames you wish to have. The limit is 29 with the Extended user frames option and 10 without it.

The controller will reconfigure to the requested number of frames at the transition from controlled start to Cold Start.

A large number of frames takes up a large amount of PERM memory. If there is not enough available PERM memory to configure the requested number of frames, the alarm

```
SYS-287 Not enough memory to increase frames
```

will be posted. $SCR.$MAXNUMUFRAM and $SCR.$MAXNUMUTOOL will reflect the actual number of frames that were able to be configured.

10.21.2 Effect on the Jog Menu

The jog menu has two modes of operation: AutoClose and Sticky.

- **In AutoClose mode** the jog menu operates as is has in the past. To change an item, press a single key on the TOOL, JOG, USER or Group lines. The item will change and the menu will close. AutoClose is the default if Extended User Frames is not loaded and fewer than nine UFrames or ten TOOL Frames are configured.

  The (.=10) on the Tool line indicates that AutoClose mode is active.

  The dot (“.”) key is interpreted as entering 10. This allows you to set the TOOL Frame to 10 in AutoClose mode.

- **In Sticky mode** you can make multiple entries. You must close the menu when done.

  In Sticky Mode, the < > characters indicate that a number is being entered but has not been applied yet. In the following example screen the jog menu is displayed on a controller with two motion groups.

```
Tool       14
Jog         3
User       55
Group       2
```

In Sticky Mode with Tool selected, when you press 2 you will see a display similar to the following:

```
Tool      <2>
Jog         3
User       55
Group       2
```

The < > symbols show that the number is being entered but has not yet been applied. When you press another number key, for example 3, the display will show <23>. You can accept this entry by

— pressing any one of ENTER, COORD, PREV, or a function Key. This will close the jog menu.

— pressing the up or down arrow. This will accept the entry, move the cursor to the item above or below, and keep the jog menu displayed.

You can edit a number before it has been accepted:

— pressing Back Space will remove the last number you entered.

— pressing Back Space again will remove the next to last number you entered.

In the example above, when <23> is displayed, after you press Back Space the <2> will remain. Pressing backspace again will remove the 2 and display the original number 14 without the < > characters.

Sticky mode is the default mode when the Extended user frames option is loaded and more than nine UFrames or ten TOOL Frames are configured. If fewer than nine UFrames or ten TOOL Frames are configured, you can enter Sticky mode by pressing ENTER. The (.=10) missing from the Tool line indicates that Sticky mode is active.

10.21.3 Programming with Extended User Frames

If you write application programs, your code should take into account the following new features introduced with Extended User Frames:

Table 10–41 shows the changes in V7.40.

Table 10–41. Programming Considerations

Item	Old Value	Value for V7.40 and Later
Special User frame number to indicate GROUP[n].$UFRAME	14	62
Special Tool frame number to indicate $GROUP[n].$UTOOL	14	30
Special User frame number to indicate $MNUFRAMENUM[n].	15	63
Special Tool frame number to indicate $MNUTOOLNUM[n]	15	31
Maximum number of User Frames	constant 9	The value of $SCR.$MAXNUMUFRAM
Maximum number of Tool Frames	constant 10	The value of $SCR.$MAXNUMUTOOL

10.22 GRAVITY COMPENSATION

Gravity compensation calculates the bending of the robot arm caused by the tool/work on the flange, the equipment on the arm, and the self weight of the arm. Then it compensates the motor position depending on the calculation of the bending, and it improves the absolute position accuracy.

Gravity Compensation option is necessary to use this function.
This function can not be used with Small Circle (not supported).

10.22.1 System Variables

Gravity Compensation

```
$PARAM_GROUP[group].$SV_DMY_LNK[8]       FALSE
    BOOLEAN           RW        PU   TRUE/FALSE
```

[Name] Gravity Compensation Enable/Disable
[Meaning]
TRUE Gravity Compensation Enable
FALSE Gravity Compensation Disable
Gravity compensation is disabled when the robot is shipped.
Please set this variable to TRUE and cycle power before use.
To set back to be disabled, set this variable to FALSE, do controlled start, and execute robot setup again. (By doing that, the motion parameters are set back to the default values. If the motion parameters have been modified, they need to be modified again.)

```
$PARAM_GROUP[group].$MOUNT_ANGLE        0
    REAL             RW      PU    -100000 ~ 100000 (deg)
```

[Name] Mount Angle of Robot
[Meaning] Set 0deg for floor mount type, 180deg for upside down type, or the mount angle for wall mount or angle mount type. Cycle power after setting.

10.22.2 MOTION Screen

1. Payload and armload (equipment on the arm) parameters are set in this screen.
2. This setting screen has three sub–screens. (MOTION screen / PAYLOAD SET screen / ARMLOAD SET screen)
3. This screen is sub–screen in SYSTEM.

```
MOTION                        JOINT 100%
Group 1
No.            PAYLOAD[kg]        Comment
 1               100.00       [            ]
 2               120.00       [            ]
 .                  .                .
 .                  .                .
10               120.00       [            ]
Active PAYLOAD number = 1
[ TYPE ]  GROUP  DETAIL  ARMLOAD  SETIND >
```

4. Payload information (Schedule No.1 to 10) can be setup. Move cursor to the line of one of the schedule numbers, and press F3(DETAIL) to enter the payload set screen.

```
MOTION/PAYLOAD/SET       JOINT    100%
   Group 1
1. Schedule  No [ 1] : [Comment      ]
2. PAYLOAD             [kg]     100.00
3. PAYLOAD CENTER X    [cm]      10.00
4. PAYLOAD CENTER Y    [cm]       0.00
5. PAYLOAD CENTER Z    [cm]      10.00
6. PAYLOAD INERTIA X   [kgfcms^2] 0.00
7. PAYLOAD INERTIA Y   [kgfcms^2] 0.00
8. PAYLOAD INERTIA Z   [kgfcms^2] 0.00
[ TYPE ] GROUP  NUMBER DEFAULT  HELP
```

5. Setup the payload, payload center, and payload inertia. X, Y, and Z directions in this screen mean X, Y, and Z axes of the default (the settings are all 0) tool frame.

   After the value is input, the message “Path and Cycletime will change. Set it ?” is displayed. Please input F4(YES) or F5(NO).

6. To enter the payload set screen of the other schedule number, press F3(NUMBER). To enter the screen for other group, press F2(GROUP). (Only in the multi–group system)

7. Press PREV key to go back to the motion screen (default screen). Press F5(SETIND) and input the schedule number to use.

8. Press F4(ARMLOAD) in the motion screen (default screen) to enter the armload set screen.

   ARMLOAD SET Screen

```
MOTION/ARMLOAD/SET          JOINT   100%
     Group 1
 1.  ARM LOAD AXIS #1      [kg]    20.00
 2.  ARM LOAD AXIS #3      [kg]    10.00
[ TYPE ] GROUP           DEFAULT  HELP
```

9. Setup the armload on axis #1 and axis #3.
   After the value is input, the message “Path and Cycletime will change. Set it ?” is displayed. Please input F4(YES) or F5(NO).
   After setting up the armload, cycle power.

10.23 GROUP MASK EXCHANGE

10.23.1 Overview

Group Mask Exchange is used to copy a teach pendant program into a new program with a different group mask than an original program. The position data from the original teach pendant program will be copied into another group in the new teach pendant program. You can copy multiple group programs from a single robot or dual robots.

Note This feature does not validate the logic of the output program, its positional data, or the program header (except group mask information). The resulting program might also require some position touchup.

Table 10–42. Group Mask Exchange Screen Items

ITEM	DESCRIPTION
Original Program	Select the name of the original program from which to copy the group mask.
New Program	Select the name of the new program to which the group mask will be copied.
Copy Orig Grp[x] to New Grp[x]	Select a motion group in the original program to copy to the new program motion groups where x identifies the group number.

10.23.2 Setting Up Group Mask Exchange

Procedure 10-39 Copying a Program Using Group Mask Exchange

1. Press MENUS.
2. Select UTILITIES.
3. Press F1, [TYPE].
4. Select Group Exchg and press ENTER. You will see a screen similar to the following.

```
Group Exchange        G1     JOINT  10 %
Program Selection                    1/8
  1 Orig Prog : (*,*,*,*,*) [SIDE_A  ]
  2 New Prog  : (*,*,*,*,*) [SIDE_A  ]
Group Selection        New Prog Group
  3 Copy Orig Grp[1] to New Grp[0]
  4 Copy Orig Grp[2] to New Grp[0]
  5 Copy Orig Grp[3] to New Grp[0]
  6 Copy Orig Grp[4] to New Grp[0]
  7 Copy Orig Grp[5] to New Grp[0]
```

Note If a program is currently selected, it is used as the default for the original and new program displayed in lines 1 and 2 above.

5. **To select a different original program,** press F3, [CHOICE].
6. **To rename the new program,** move the cursor to line 2 and press ENTER. You can then choose a method to rename the program and type the new program name. You must then identify the new group for the copy to take place.
7. To select a motion group in the original program to copy to the new program motion groups:
   a. Move the cursor to the first New group [#] to copy.

b. Type a group number, and press ENTER.

Note If any of the entries for New Group [#] are left at 0, this group will be deleted from the new program.

8. **To clear a single entry,** press NEXT, > and then press F1, CLEAR.
9. **To clear all entries,** press NEXT, > and then press SHIFT and F1, CLEAR.

Note If lines 1 and 2 are changed, the New Grp column in lines 3 through 7 are cleared.

10. When you have finished identifying the groups to be copied, press F2, EXEC.

Note If the new program exists when you press F2, EXEC, you must confirm the overwrite operation before the execute takes place.

10.23.3 Troubleshooting Group Mask Exchange

Prompt messages will be displayed during Group Mask Exchange setup. Use Table 10–43 to determine the cause of a problem, and to solve it.

Table 10–43. Group Mask Exchange Troubleshooting

Prompt Message	Possible Cause	Solution
%s not aborted, abort program ?	%s is the name of the program. The original or new program is Running or Paused.	Press F4 for Yes to ABORT program. Press F5 for NO and operation will be canceled
Group %d does not exist in Orig program	%d is the original program group number. The group number does not exist in original program. This group number can not be copied to new program.	Set new group number to 0 or to a new group number for %d number.
Group %d does not exist on controller	%d is the original program group number. The group number does not exist on the controller. This group number cannot be copied on this controller.	Set the new group number to 0 or to a new group number for %d number.

Table 10–43. Group Mask Exchange Troubleshooting (Cont'd)

Prompt Message	Possible Cause	Solution
New Grp %d axis do not match Orig Grp %d	%d is the group number for new program and original program. The original group cannot be copied to new group.	Set the new group number to 0 or to a new group number for %d number of original group.
New Grp %d is an INDEXER cannot copy	%d is the new program group number. New program group is an INDEXER and cannot be copied.	Set new group number to 0 for %d number of new group.
New Program could not be created	1) Answering NO to Overwrite.	1) Overwrite the existing program or change the new program name.
	2) Refer to the error message posted in the alarm log.	2) Refer to the *Error Code Manual* for error remedy information.
New Program created successfully	This message is a status message only.	None
No New Grps specified	All New Grps are "0"	You must specify a New Grp to execute Group Mask Exchange.
Orig Grp %d is an INDEXER cannot copy	%d is the original program group number. Original program group is an INDEXER and cannot be copied.	Set new group number to 0 for %d number of original group.
Program exist, Overwrite ?	The new program exists on the controller.	Press F4, YES, to overwrite the program on the controller. Press F5, NO, to cancel the operation.
Program names match	Original and new program name are the same.	Change the original or the new program name.
Robot types do not match, Orig Grp %d	%d is the original program group number. Robot types do not match and cannot be copied. Robots must be copied to robots; nobots must be copied to nobots; positioners must be copied to positioners.	Set new group number to 0 or to a new group number for %d number of original group.
WORKING……	The utility is copying the program	Wait for the utility to complete the copy.

10.24 HIGH SPEED INPUT FUNCTIONS

10.24.1 Overview

The following High Speed Input Functions are available:

- Frequency Interface
- High Speed Scanning (for Line Tracking Part Detect)
- Position Snap and Map

10.24.2 Frequency Interface

Outputs are supplied as system variables that are accessible by the FANUC Robotics Software.

The following outputs are supplied:

- Frequency Reading (x1000 Hz)
- Total Pulse Count
- Standard Operator Panel Input (SOPIN)

Refer to the *FANUC Robotics High Speed Digital Input Module (HSI)) Option Manual* for more information.

10.24.3 High Speed Scanning (for Line Tracking Part Detect).

The High Speed Scanning feature ensures an accurate part detection process when the conveyor operates at very fast speeds. It uses the High Speed Digital Input (HSDI) #1, located on the JRL5 connector of the controller, in place of the standard digital input normally used for part detection.

To use High Speed Scanning, you must

- Enable the High Speed Scanning system variable
- Modify your line tracking program

Refer to the *Linetracking Setup and Operations Manual* for more information.

10.24.4 Position Snap & Map

The Position Snap and Map feature, available in software version 6.40 and later, records the current Robot Position in encoder counts when a single High Speed Digital Input (H.D.I.) is turned on. In a system with a robot-mounted vision system camera or a measurement sensor, a signal is provided to both the camera and the robot controller. This signal causes the robot to *snap* its current position at the same instant that the image is snapped. Later, additional software acquires and relates the robot positional data with the image taken - thus a *map* of the robot's position. This functions with the robot moving at reasonable speeds, thus increasing cycle time and accuracy of the system.

Refer to the *FANUC Robotics High Speed Digital Input Module (HSI)) Option Manual* for more information.

10.25 LIMIT SET

The limit set function allows you to determine the software axis limits for a robot automatically, and to determine the mechanical stopper locations. If you want to set the axis limits without the limit set function, you must run all programs step-by-step, record the maximum value for each angle, and then set the axis joint limits manually.

The limit set function reads all of the programmed positions on the robot automatically and determines the maximum and minimum taught joint angles used in all programs. It then takes this information and sets the specified joint limits automatically, while taking into account a limit buffer you specify. The limit set function also reports the appropriate angles for the mechanical stoppers.

By default, software axis limits will be set only for axes 1, 2, and 3. You can specify that they be set for more axes, if desired.

A *limits buffer* is added to the detected limits to allow tolerance for motion *between* positions when you run a program. When the robot moves from one position to another, the motion of the robot between the positions might be outside of the axis limits. The limits buffer is applied to the detected maximum and minimum taught points to ease the axis limits so that a joint limit error does not occur on the motion between the taught points.

You can set a limits buffer of from 0 to 50 degrees. The limits buffer is set to 10 degrees, by default. A limits buffer of 10 degrees provides adequate ease in many cases. After using the limit set function, if you encounter many joint limit errors while running programs, increase the limits buffer and try again.

Before you set limits, you select the axes for which you want to set limits and change the value of the limits buffer, if desired. You can reset the limits to factory settings if you do not want to use the new ones.

Procedure 10-40 Using the Limit Set Function to Set Software Axis Limits

Steps

1. Press MENUS.
2. Select UTILITIES.
3. Press F1, [TYPE].
4. Select Limit Set and press ENTER. You will see a screen similar to the following.

```
Limit Set
 Axis                  Set Limit
 1                     Yes
 2                     Yes
 3                     Yes
 4                     No
 5                     No
 6                     No
 Limits Buffer:        10 dg
```

5. **To view the currently active settings,** press PREV. To return to this menu, press PREV again.
6. **To select a motion group** for limit setting, press F2, GROUP, type the desired group number, and press ENTER.
7. **To select an axis for limit setting**
   a. Move the cursor to the number of the axis you want to select.
   b. Press F4, YES, to select the axis to set. If you press F5, NO, the limit will not be set for that axis.
8. To set the limits buffer, move the cursor to Limits Buffer, type the number of degrees, and press ENTER.
9. **To set the axis limits**, press F3, EXECUTE.

When the limits have been set, you will see a screen similar to the following.

```
New Limits
  Axis           LOWER         UPPER
  J1             -76           53 dg
  J2             -15          -15 dg
  J3             -15           30 dg
  J4             -190          200 dg
  J5             -10           90 dg
  J6             -100          280 dg
  J7              0            0 dg
  J8              0            0 dg
  J9              0            0 dg


J1 MECHANICAL STOPPER LOCATION
   Min Minus Side:  -76
   Min Plus Side:    28


J2 MECHANICAL STOPPER LOCATION
   Min Minus Side:  -17
   Min Plus Side:    12


J3 MECHANICAL STOPPER LOCATION
   Min Minus Side:  -25
   Min Plus Side:    23
YOU MUST COLD START TO TAKE EFFECT
```

Note The following operation is not supported on some robot models.

10. **To reset the axis limits to their factory settings,** press F6, DEFAULT.

    The MECHANICAL STOPPER LOCATION provides the minimum and maximum angles at which the mechanical stoppers can be placed for the specified axis. Refer to the applicable robot mechanical manual for diagrams of these locations.

    Note You must always cycle power on the controller in order for software axis limit changes to take effect.

11. **When you are finished setting axis limits, cycle power on the controller** so that the new software axis limits can take effect.

Note If you encounter many joint limit errors while running programs, increase the limits buffer and try running the Limit Set Utility again.

10.26 MENU UTILITY (OPTION)

10.26.1 Overview

You can create your own teach pendant menus to help your application run more smoothly. To use your own menus, first perform menu setup then call the menu macro program from a teach pendant program and select the desired menu in the parameter.

See Figure 10–90 for an example program and example menus.

Figure 10–90. Teach Pendant Menus Example

```
 1:  Prompt Box Msg('NotAtPerch')  prompt msg macro
 2:  Prompt Box YN (1, 21)         prompt macro,  Answer in R[21]
 3:  IF R[21]=0; CALL ABORTIT      0=NO, abort
 4:  IF R[21]=1,CALL MOVEPRCH      1=YES
 5:  Op. Entry Menu (1)            operator entry macro
 6:  LBL[1] ;
 7:  List Menu (44,22)             list macro, returns
 8  SELECT R[22]=1,CALL TOOL1          answer in R[22]
 9:        =2,CALL TOOL2
10:        =3,JMP LBL[2]
11:        ELSE, CALL ABORTIT
12:  LBL[2]
13:  Status Menu ('status1')      status menu macro
14:
15  LBL[3]
16 P[1] 50% FINE
17 P[2] 50% FINE
18 CALL TEST3
19:J P[4] 50% FINE
20:J P[1] 100% FINE
21:J P[2] 100% FINE
```

Prompt Box Message Menu

```
Robot is not at PERCH
position.
      OK
```

Prompt Box Yes/No Menu

```
Move to perch ?
YES    NO
```

Operator Entry Menu

```
          Check Chute
1 Are parts in chute?  YES
2 How many parts ?    3
          DONE
```

List Menu

```
Tool Selection Menu
  1 Tool 1
  2 Tool 2
  3 No Change
  4 Abort Production
Enter Selection:__
```

Status Menu

```
        Status Menu
Cur. Program  TESTIV
Cur. Routine  TEST3
Status        Moving
Time          ##-XXX-## 09:04
              MENU  C_STOP
```

This section describes how to

- Create menus. Refer to Table 10–44 .
- Include instructions in a teach pendant program to display these menus

Table 10–44. SETUP Menu Utility Screen Items

ITEM	DESCRIPTION
Prompt box msg	This item is used to make sure the operator is aware of something. When a prompt box message screen is displayed, the teach pendant program pauses until the operator presses ENTER.
Prompt box yes/no	This item asks the operator a question that can be answered YES or NO. When a prompt box yes/no screen is displayed, the teach pendant program pauses until the operator selects YES or NO and presses ENTER.
Select from a list	This item asks the operator to select an item from a list of items. When a list menu is displayed, the teach pendant program pauses until the operator selects an item from the list.
Status menus	This item displays status information to the teach pendant screen. When a status menu is executed, the teach pendant displays the status menu and then continues executing. The values that are displayed are refreshed at least once every second.
Operator entry	This item requires the operator to enter INTEGER, REAL, BOOLEAN, or text values. When an operator entry menu is executed, the program pauses until the operator [TYPE]s the information and then presses F3, DONE.

The Teach Pendant Menus option provides some predefined menus that you can use as examples. You can use these sample menus as they are or you can change them as necessary.

10.26.2 Prompt Box Message Menu

A prompt box message menu is used to make sure the operator is aware of something. When a prompt box message screen is displayed, the teach pendant program pauses until the operator presses ENTER. In addition, when the prompt box is displayed, the operator cannot access any other menus. See Figure 10–91 for an example of a prompt box message menu.

Figure 10–91. Prompt Box Message Menu

```
Robot is not at
            PERCH position
                  OK
```

Macro: Prompt Box Msg (menu)

A prompt box message menu uses the Prompt Box Msg(menu) macro.

Parameter 1: menu - This is the menu number or menu name. The menu can be chosen by number or name.

Errors

The following errors are possible:

- If the Prompt Box Msg(menu) macro was used in the teach pendant program without a parameter, an alarm is given and the program is aborted.
- If the selected prompt box message menu does not exist when it is called from a teach pendant program, a prompt box similar to the one shown in Figure 10–92 will be displayed. To abort the program, select OK.

Figure 10–92. Prompt Box Message Menu Does not Exist Screen



Menu File Name

Each prompt box message menu is stored in its own variable file that begins with the prefix " **MENU1** ." The next two characters of the filename are the menu number. For example, the prompt box message menu 1 is stored in the variable file named "MENU1 **01** .VR."

Examples

See Prompt Box Msg(menu) Macro Example 1 and Prompt Box Msg(menu) Macro Example 2 for examples of how to use the Prompt Box Msg(menu) instruction in a teach pendant program.

Prompt Box Msg(menu) Macro Example 1

```
PROG  TEST1
                                    1:  Prompt Box Msg(1)
```

Prompt Box Msg(menu) Macro Example 2

```
PROG  TEST2
                                    1:  Prompt Box Msg('NotAtPerch')
```

Setup Procedure

You can enter the text you want displayed in the prompt box from the Prompt box msg menus screen. Table 10–45 lists the Prompt box msg Menu Screen Items. Table 10–46 lists the Prompt box msg DETAIL Menu Screen Items.

Use Procedure 10-41 to create a prompt box message menu.

Table 10–45. Prompt box msg Menus Screen Items

ITEM	DESCRIPTION
Item	This item refers to item numbers (1-99) always in sequence.
Menu number	This item refers to unique menu numbers (1-99).
Menu name	This item is the name of the menu.

Table 10–46. Prompt msg menus DETAIL Screen Items

ITEM	DESCRIPTION
Prompt msg no:	This item refers to unique menu numbers (1-99).
Menu name	This item is the name of the menu.
Line	This item is the line number for the message. Maximum five line.
Text	This item is the display text used in the prompt box.

Procedure 10-41 Creating a Prompt Box Message Menu

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Menu Utility. You will see a screen similar to the following.

```
SETUP Menu Utility
Choose a menu type
  1      Prompt box msg
  2      Prompt box yes/no
  3      Select from a list
  4      Status menus
  5      Operator entry
```

5. To display menu tips:

Menu tips are prompt boxes that are displayed the first time you press the menu detail key for each menu type. The prompt box contains information that describes the menus, macros, and parameters.

a. Press F4, CONFIG. You will see a screen similar to the following.

```
SETUP Menu Utility
Config Menus
 1   Show menu tips:              YES
```

b. **To turn off menu tips** , press F5, NO.

To turn on menu tips , press F4, YES. You will see a screen similar to the following.

```
These menus are used with
         macro Prompt Box Msg.
         Parameter 1 selects the menu.
                    OK
```

When you are finished viewing menu tips , press ENTER to select OK.

6. Move the cursor to Prompt box msg and press F2, DETAIL. You will see a screen similar to the following.

```
SETUP Menu Utility
Prompt box msg menus
ITEM Menu no      Menu name
 1      1         NotAtPerch
 2      2         NotAtPerch
 3      3         Machine error
```

7. To create a new menu:

a. Press F3, CREATE.

b. Type the number of the new menu and press ENTER. You will see a screen similar to the following.

```
SETUP Menu Utility
Prompt msg menu     1
 Menu name: NotAtPerch
 Line    Text
  1
  2
  3
  4
  5
```

c. Move the cursor to Menu name and press ENTER. Use the appropriate function keys to name the menu and press ENTER.

d. Move the cursor to each line and press ENTER. Use the appropriate function keys to type the menu text and press ENTER.

e. **To test the menu you created** , press F3, TEST. You will see a screen similar to the following.

```
SETUP Menu Utility
Prompt msg menu     1
 Menu name: NotAtPerch
 Line    Text
  1
  2                   Robot is not at
  3                   PERCH position
  4
  5                         OK
```

8. To modify an existing menu:

a. Move the cursor to the menu name and press F2, DETAIL. You will see a screen similar to the following.

```
SETUP Menu Utility
Prompt msg menu     1
 Menu name: NotAtPerch
 Line    Text
  1
  2      Robot is not at PERCH
  3      position.
  4
  5
```

b. Move the cursor to the lines you want to change, press ENTER, and press the appropriate function keys to type the text. When you are finished with each line, press ENTER.

9. To save (back up) a menu to the default device:

a. Set the default device. Refer to Section 9.1.3 .

b. Move the cursor to the name of the menu you want to save.

c. Press NEXT, >, and press F1, SAVE. The menu will be saved to the default device.

Note If the file already exists on the default device, you will be prompted whether to replace the existing file with the one you are backing up. To replace the file, select YES. To cancel, select NO.

Note You can also back up a menu using FILE, BACKUP APPLICATION. Refer to Chapter 9 *PROGRAM AND FILE MANIPULATION* for more information.

Note Each prompt box message menu is stored in its own variable file that begins with the prefix " **MENU1** ." The next two characters of the filename are the menu number. For example, the prompt box message menu 1 is stored in the variable file named "MENU1 **01** .VR."

10. To load (restore) a menu:

a. Move the cursor to the name of the menu you want to load.

b. Press NEXT, >, and press F2, LOAD. The menu will be loaded from the default device.

Note If the file already exists on the controller, you will be prompted whether to replace the existing file with the one you are loading. To replace the file, select YES. To cancel, select NO.

Note You can also load a menu using FILE, LOAD, *.VR. Refer to Chapter 9 *PROGRAM AND FILE MANIPULATION* for more information.

11. To clear a menu:

a. Move the cursor to the name of the menu you want to clear.

b. Press NEXT, >, and press F3, CLEAR.

c. Confirm the clear:

- To clear the menu, select YES and press ENTER.

- To cancel the clear, select NO and press ENTER.

12. To copy a menu:

a. Move the cursor to the name of the menu you want to copy.

b. Press NEXT, >, and press F4, COPY.

c. Type the menu number to which you want to copy the menu and press ENTER.

Note If the new menu already exists, you will be prompted whether to replace the existing menu with the one you are copying. To replace the old menu with the new menu, select YES. To cancel the operation, select NO.

10.26.3 Prompt Box Yes/No Menu

A prompt box yes/no menu asks the operator a question that can be answered YES or NO. When a prompt box yes/no screen is displayed, the teach pendant program pauses until the operator selects YES or NO and presses ENTER. In addition, when the prompt box yes/no screen is displayed, the operator cannot access any other menus. See Figure 10–93 for an example of a prompt box yes/no menu.

Figure 10–93. Prompt Box Yes/No Menu

```
Send the robot home?
                  YES     NO
```

Macro: Prompt Box YN(menu, answer_reg)

A prompt box yes/no menu uses the Prompt Box YN(menu, answer_reg) macro.

Parameter 1: menu - This is the menu number or menu name. The menu can be chosen by number or name.

Parameter 2: answer_reg - This is the register that will contain the answer from the operator.

- R[answer_reg] = 0 if operator selected NO
- R[answer_reg] = 1 if operator selected YES

Errors

The following errors are possible:

- If the Prompt Box YN(menu, answer_reg) macro was used in the teach pendant program without two parameters, an alarm is given and the program is aborted.
- If the selected prompt box yes/no menu does not exist when it is called from a teach pendant program, a prompt box similar to the one shown in Figure 10–94 will be displayed. To abort the program, select OK.

Figure 10–94. Prompt Box Yes/No Menu Does not Exist Screen

```
Error:
               Selected menu
               does not exist.

                    OK
```

- If the register answer_reg does not exist, then an alarm is given and the program is aborted.

Menu File Name

Each prompt box yes/no menu is stored in its own variable file that begins with the prefix " **MENU2** ." The next two characters of the filename are the menu number. For example, the prompt box yes/no menu 1 is stored in the variable file named "MENU2 **01** .VR."

Examples

See Prompt Box YN(menu, answer_reg) Macro Example 1 and Prompt Box YN(menu, answer_reg) Macro Example 2 for examples of how to use the Prompt Box YN(menu, answer_reg) instruction in a teach pendant program.

Prompt Box YN(menu, answer_reg) Macro Example 1

```
PROG  TEST1
                                    1:  Prompt Box YN(1,2)
                                    2:  IF R[2] = 0, CALL ABORTIT
```

Prompt Box YN(menu, answer_reg) Macro Example 2

```
PROG  TEST2
                                        1:  Prompt Box YN('NotAtPerch',6)
                                        2:  IF R[6] = 1, JMP LBL[5]
```

Setup Procedure

Table 10–47 lists the Prompt Box Yes/No Menu Screen Items. Table 10–47 lists the Prompt Box Yes/No Menu DETAIL Screen Items. Use Procedure 10-42 to create a prompt box yes/no menu.

Table 10–47. Prompt Box Yes/No Menu Screen Items

ITEM	DESCRIPTION
Item	This item refers to item numbers (1-99) always in sequence.
Menu number	This item refers to unique menu numbers (1-99).
Menu name	This item is the name of the menu.

Table 10–48. Prompt Box Yes/No Menu DETAIL Screen Items

ITEM	DESCRIPTION
Prompt Y/N menu no:	This item refers to unique menu numbers (1-99).
Menu name	This item is the name of the menu.
Line	This item is the line number for the message. Maximum five lines.
Text	This item is the display text used in the prompt box.

Procedure 10-42 Creating a Prompt Box Yes/No Menu

Conditions

- The default device is set. Refer to Procedure 9-1 .

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Menu Utility. You will see a screen similar to the following.

```
SETUP Menu Utility
Choose a menu type
 1       Prompt box msg
 2       Prompt box yes/no
 3       Select from a list
 4       Status menus
 5       Operator entry
```

5. To display menu tips:

Menu tips are prompt boxes that are displayed the first time you press the menu detail key for each menu type. The prompt box contains information that describes the menus, macros, and parameters.

a. Press F4, CONFIG. You will see a screen similar to the following.

```
SETUP Menu Utility
Config Menus
 1   Show menu tips                YES
```

b. **To turn off menu tips** , press F5, NO.

To turn on menu tips , press F4, YES. You will see a screen similar to the following.

```
These menus are used with
         macro Prompt Box YN.
         Parameter 1 selects the menu.
         Parameter 2 returns:
                     1=yes  0=no
                   OK
```

When you are finished viewing menu tips , press ENTER to select OK.

6. Move the cursor to Prompt box yes/no and press F2, DETAIL. You will see a screen similar to the following.

```
SETUP Menu Utility
Prompt box YN menus
ITEM  Menu no     Menu name
 1       1        SendHome
 2       5        Send Perch
```

7. To create a new menu:

a. Press F3, CREATE.

b. Type the number of the new menu and press ENTER. You will see a screen similar to the following.

```
SETUP Menu Utility
Prompt box YN menus     1
 Menu name:
 Line    Text
  1
  2      Send the robot home?
  3
  4
  5
```

c. Move the cursor to Menu name and press ENTER. Use the appropriate function keys to name the menu and press ENTER.

d. Move the cursor to each line and press ENTER. Use the appropriate function keys to type the menu text and press ENTER.

e. **To test the menu you created** , press F3, TEST. You will see a screen similar to the following.

```
SETUP Menu Utility
Prompt box YN menus   1
 Menu name: SendHome
 Line
  1
  2           Send the robot home?
  3
  4                YES   NO
  5
```

8. To modify an existing menu:

a. Move the cursor to the menu name and press F2, DETAIL. You will see a screen similar to the following.

```
SETUP Menu Utility
Prompt box YN menus     1
 Menu name:  MoveHome
 Line    Text
  1
  2      Move the robot home?
  3
  4
  5
```

b. Move the cursor to the lines you want to change, press ENTER, and press the appropriate function keys to type the text. When you are finished with each line, press ENTER.

9. To save (back up) a menu to the default device:

a. Set the default device. Refer to Procedure 9-1 in Chapter 9 *PROGRAM AND FILE MANIPULATION* .

b. Move the cursor to the name of the menu you want to save.

c. Press NEXT, >, and press F1, SAVE. The menu will be saved to the default device.

Note If the file already exists on the default device, you will be prompted whether to replace the existing file with the one you are backing up. To replace the file, select YES. To cancel, select NO.

Note You can also back up a menu using FILE, BACKUP APPLICATION. Refer to Chapter 9 *PROGRAM AND FILE MANIPULATION* for more information.

Note Each prompt box yes/no menu is stored in its own variable file that begins with the prefix " **MENU2** ." The next two characters of the filename are the menu number. For example, the prompt box yes/no menu 1 is stored in the variable file named "MENU2 **01** .VR."

10. To load (restore) a menu:

a. Move the cursor to the name of the menu you want to load.

b. Press NEXT, >, and press F2, LOAD. The menu will be loaded from the default device.

Note If the file already exists on the controller, you will be prompted whether to replace the existing file with the one you are loading. To replace the file, select YES. To cancel, select NO.

Note You can also load a menu using FILE, LOAD, *.VR. Refer to Chapter 9 *PROGRAM AND FILE MANIPULATION* for more information.

11. To clear a menu:

a. Move the cursor to the name of the menu you want to clear.

b. Press NEXT, >, and press F3, CLEAR.

c. Confirm the clear:

- To clear the menu, select YES and press ENTER.
- To cancel the clear, select NO and press ENTER.

12. To copy a menu:

a. Move the cursor to the name of the menu you want to copy.

b. Press NEXT, >, and press F4, COPY.

c. Type the menu number to which you want to copy the menu and press ENTER.

Note If the new menu already exists, you will be prompted whether to replace the existing menu with the one you are copying. To replace the file, select YES. To cancel, select NO.

10.26.4 List Menu

A list menu asks the operator to select an item from a list of items. When a list menu is displayed, the teach pendant program pauses until the operator selects an item from the list. See Figure 10–95 for an example of a list menu.

Figure 10–95. List Menu

```
USER
    Move Menu
    1 Ignore
    2 Move to Maint. Pos.
    3 Move to Perch Pos.
```

Macro:List Menu(menu, answer_reg)

A list menu uses the List Menu(menu, answer_reg) macro.

Parameter 1: menu - This is the menu number or menu name. The menu can be chosen by number of name.

Parameter 2: answer_reg - This is the register that will contain the answer from the operator.

- R[answer_reg] = item number the operator selected

Display

To display a list menu you create, use the List Menu macro. The menu will be displayed automatically on the USER screen. If the operator changes the teach pendant display to another screen during teach pendant execution of this menu but before making the selection, he can return to this screen by pressing MENUS and selecting USER.

Setup

In a list menu, you can specify

- Up to eight list items from which to choose
- A macro program to execute when the item is selected, if desired; this is called the **ActionTP program** .

  Note The ActionTP program is a teach pendant program that is executed when the corresponding item number is selected. Do not use ActionTP teach pendant programs that use menu macros.

Errors

The following errors are possible:

- If the List Menu(menu, answer_reg) macro was used in a teach pendant program without two parameters, an alarm is given and the program is aborted.
- If the selected list menu does not exist when it is called from a teach pendant program, a prompt box similar to the one shown in Figure 10–96 will be displayed. To abort the program, select OK.

Figure 10–96. List Menu Does not Exist Screen

```
Error:
              Selected menu
              does not exist.
                    OK
```

- If the register answer_reg does not exist, then an alarm is given and the program is aborted.

Menu File Name

Each list menu is stored in its own variable file that begins with the prefix " **MENU3** ." The next two characters of the filename are the menu number. For example, the list menu 1 is stored in the variable file named "MENU3 **01** .VR."

Examples

See List Menu(menu, answer_reg) Macro Example 1 and List Menu(menu, answer_reg) Macro Example 2 for examples of how to use the List Menu(menu, answer_reg) macro instruction in a teach pendant program.

List Menu(menu, answer_reg) Macro Example 1

```
PROG  TEST1
1:  List Menu(2, 21)
2:  SELECT R[21]=1,CALL TOOL1
3:              =2,CALL TOOL2
4:              =3,JMP LBL[2]
5:              ELSE,CALL ABORTIT
```

List Menu(menu, answer_reg) Macro Example 2

```
PROG  TEST2
1:  List Menu('ErrorRecov',2)
2:  SELECT R[2]=1,JMP LBL[2]
3:         =2,CALL MOVEMANT
4:         =3,CALL MOVEPRCH
5:          ELSE,CALL ABORTIT
```

Setup Procedure

Table 10–49 lists the Select from a list Screen setup items. Table 10–50 lists the Select from a list DETAIL Screen setup items.

Use Procedure 10-43 to create a list menu.

Table 10–49. Select from a list Screen Items

ITEM	DESCRIPTION
Item	This item refers to item numbers (1-99) always in sequence.

Table 10–49. Select from a list Screen Items (Cont'd)

Menu number	This item refers to unique menu numbers (1-99).
Menu name	This item is the name of the menu.

Table 10–50. List Menu DETAIL Screen Items

ITEM	DESCRIPTION
List menu no:	This item refers to unique menu numbers (1-99).
Menu name	This item is the name of the menu.
Line	This item is the line number for the message. Maximum eight lines.
Text	This item is the display text used in the prompt box.
ActionTP	This item is the TP program you want to run when the corresponding item is selected.
Title	This item is the title for the message.
PROMPT	This item allows you to enter a selection.

Procedure 10-43 Creating a List Menu

Conditions

- The default device is set. Refer to Procedure 9-1 .

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Menu Utility. You will see a screen similar to the following.

```
SETUP Menu Utility
Choose a menu type:
  1      Prompt box msg
  2      Prompt box yes/no
  3      Select from a list
  4      Status menus
  5      Operator entry
```

5. To display menu tips:

Menu tips are prompt boxes that are displayed the first time you press the menu DETAIL key for each menu type. The prompt box contains information that describes the menus, macros, and parameters.

a. Press F4, CONFIG. You will see a screen similar to the following.

```
SETUP Menu Utility
Config Menus
 1   Show menu tips:                YES
```

b. **To turn off menu tips**, press F5, NO.

To turn on menu tips, press F4, YES. You will see a screen similar to the following.

```
These menus are used with
        macro List Menu and
        macro Check Cycle Stop.
        Parameter 1 selects the menu.
        Parameter 2 returns
        the operators choice.  (0=ERROR)
                    OK
```

When you are finished viewing menu tips, press ENTER to select OK.

6. Move the cursor to Select from a list and press F2, DETAIL. You will see a screen similar to the following.

```
SETUP Menu Utility
LIST menu
ITEM Menu no     Menu name
 1     1         MoveMenu
 2     11        Cycle Stop
```

7. To create a new menu:

a. Press F3, CREATE.

b. Type the number of the new menu and press ENTER. You will see a screen similar to the following.

```
SETUP Menu Utility
List Menu
 Menu name:  MoveMenu
 Line     Text                          ActionTP
TITLE: Move Menu
  1      Ignore
  2      Move to Maint. Pos.           MOVEMAINT
  3      Move to Perch Pos.            MOVEPRCH
  4
  5
  6
  7
  8
PROMPT: Choose one and press ENTER:
```

c. Move the cursor to Menu name and press ENTER. Use the appropriate function keys to name the menu and press ENTER.

d. Move the cursor to each item and press ENTER. Use the appropriate function keys to type the list item text and press ENTER.

Note The ActionTP program is a teach pendant program that is executed when the corresponding item number is selected. Do not use ActionTP teach pendant programs that use menu macros.

e. **If you want to associate an ActionTP program** with a list item, move the cursor to the item and

- Press F2, [ALPH], use the appropriate function keys to type the teach pendant name, and press ENTER, **or**

- Press F4, [CHOICE], choose the name of the teach pendant program, and press ENTER.

f. **To test the menu you created** , press F3, TEST.

The list menu's display variables are checked to make sure that any ActionTP program listed exists.

If no errors were found , you will see the following screen and then the menu will be displayed.

```
List menu no       1
          Checking variable definitions
   No errors found in menu definitions !
  Testing menu...
```

```
Move Menu
       1 Ignore
       2 Move to Maint. Pos.
       3 Move to Perch Pos.
         Choose one and press ENTER:
```

If errors were found , they will be listed on the teach pendant screen and the menu will not be displayed. The previous menu will be displayed. Correct the ActionTP program names and press F3, TEST, again.

g. **To test the ActionTP program** , move the cursor to the ActionTP you want to test, press and continue pressing the DEADMAN switch, turn the teach pendant ON/OFF switch to ON, and press SHIFT and F3, TEST.

8. To modify an existing menu:

a. Move the cursor to the menu name and press F2, DETAIL. You will see a screen similar to the following.

```
SETUP Menu Utility
List Menu
 Menu name:  MoveMenu
 Line     Text                          ActionTP
TITLE: Move Menu
  1      Ignore
  2      Move to Maint. Pos.           MOVEMAINT
  3      Move to Perch Pos.            MOVEPRCH
  4
  5
  6
  7
  8
PROMPT: Choose one and press ENTER:
```

b. Move the cursor to the lines you want to change, press ENTER, and press the appropriate function keys to type the text. When you are finished with each line, press ENTER.

9. To save (back up) a menu to the default device:

a. Set the default device. Refer to Procedure 9-1 in Chapter 9 *PROGRAM AND FILE MANIPULATION* .

b. Move the cursor to the name of the menu you want to save.

c. Press NEXT, >, and press F1, SAVE. The menu will be saved to the default device.

Note If the file already exists on the default device, you will be prompted whether to replace the existing file with the one you are backing up. To replace the file, select YES. To cancel, select NO.

Note You can also back up a menu using FILE, BACKUP APPLICATION. Refer to Chapter 9 *PROGRAM AND FILE MANIPULATION* for more information.

Note Each list menu is stored in its own variable file that begins with the prefix " **MENU3** ." The next two characters of the filename are the menu number. For example, the list menu 1 is stored in the variable file named "MENU3 **01** .VR."

10. To load (restore) a menu:

a. Move the cursor to the name of the menu you want to load.

b. Press NEXT, >, and press F2, LOAD. The menu will be loaded from the default device.

Note If the file already exists on the controller, you will be prompted whether to replace the existing file with the one you are loading. To replace the file, select YES. To cancel, select NO.

Note You can also load a menu using FILE, LOAD, *.VR. Refer to Chapter 9 *PROGRAM AND FILE MANIPULATION* for more information.

11. To clear a menu:

a. Move the cursor to the name of the menu you want to clear.

b. Press NEXT, >, and press F3, CLEAR.

c. Confirm the clear:

- To clear the menu, select YES and press ENTER.

- To cancel the clear, select NO and press ENTER.

12. To copy a menu:

a. Move the cursor to the name of the menu you want to copy.

b. Press NEXT, >, and press F4, COPY.

c. Type the menu number to which you want to copy the menu and press ENTER.

Note If the new menu already exists, you will be prompted whether to replace the existing menu with the one you are copying. To replace the file, select YES. To cancel, select NO.

10.26.5 Status Menu

A status menu displays status information to the teach pendant screen. When a status menu is executed, the teach pendant displays the status menu and then continues executing. The values that are displayed are refreshed at least once every second. This is a read-only menu.

See Figure 10–97 for an example of a status menu.

Figure 10–97. Status Menu

```
STATUS Menu
               Production Status
      Program:           TESTIII
      Routine:           TESTIII
      Status:            RUNNING
      PART Count         101
      Part on Conv1      ON
      Pallet Count       15
```

Macro: Status Menu(menu)

A status menu uses the Status Menu(menu) macro.

Parameter 1: menu - This is the menu number or menu name. The menu can be chosen by number or name.

Display

When a status menu is executed, the teach pendant displays the status menu and then continues executing. The values that are displayed are refreshed.

A status menu can also be displayed as follows:

1. Press MENUS.
2. Select STATUS
3. Press F1, [TYPE].
4. Select Prod. Status.

Several status menus can be displayed. If the status menu you want to see is not the one displayed, either press F2, MENU, and then select the number of the status menu you want to display, or press F4, [CHOICE], and select the menu by name.

Errors

The following errors are possible:

- If the Status Menu macro was used in a teach pendant program without a parameter, an alarm is displayed. However, the program **is not** aborted and the teach pendant screen is not changed.
- If a menu that was selected by name does not exist, or the menu is the wrong kind of menu, an alarm is given. However, the program **is not** aborted and the teach pendant screen is not changed.
- If the selected status menu does not exist when it is called for in a teach pendant program by number, an alarm is given and the teach pendant screen is changed to the default status menu, shown in Figure 10–98 .

Figure 10–98. Default Status Menu

```
STATUS Menu
           ProductionStatus
      Current Program   TESTIV
      Current Routine
      Current Line      0
      Status            ABORTED
      Time              20-JAN-XX 12:27
```

Menu File Name

Each status menu is stored in its own variable file that begins with the prefix " **MENU4** ." The next two characters of the filename are the menu number. For example, the status menu 1 is stored in the variable file named "MENU4 **01** .VR."

Setup

In a status menu, you can specify

- Up to nine status items
- The names of the variables that correspond to the status items, and detailed variable information

Table 10–51 lists and describes the variable detail information.

Table 10–51. Status Menu Variable Detail Information

ITEM	DESCRIPTION
Display Text	This item allows you to specify the text that will be displayed for the status item. You can type up to 17 characters of display text.

Table 10–51. Status Menu Variable Detail Information (Cont'd)

<table>
<tr><th>ITEM</th><th colspan="3">DESCRIPTION</th></tr>
<tr><td rowspan="18">Variable Type</td><td colspan="3">This item allows you to specify the kind of variable for the status item:</td></tr>
<tr><td>Type of Variable</td><td>What to Define</td><td>Data Display Type</td></tr>
<tr><td>NONE</td><td>none</td><td>none</td></tr>
<tr><td>R[ ]</td><td>REG Number</td><td>INTEGER, REAL, BOOLEAN</td></tr>
<tr><td>DIN[ ]</td><td>I/O Number</td><td>INTEGER, BOOLEAN, TEXT</td></tr>
<tr><td>DOUT[ ]</td><td>I/O Number</td><td>INTEGER, BOOLEAN, TEXT</td></tr>
<tr><td>RDI[ ]</td><td>I/O Number</td><td>INTEGER. BOOLEAN, TEXT</td></tr>
<tr><td>RDO[ ]</td><td>I/O Number</td><td>INTEGER, BOOLEAN, TEXT</td></tr>
<tr><td>GIN[ ]</td><td>I/O Number</td><td>INTEGER</td></tr>
<tr><td>GOUT[ ]</td><td>I/O Number</td><td>INTEGER</td></tr>
<tr><td>KAREL VAR</td><td>Prog. Name, Var. Name</td><td>INTEGER, REAL, BOOLEAN, TEXT</td></tr>
<tr><td>SYSTEM VAR</td><td>System Variable Name</td><td>INTEGER, REAL, BOOLEAN, TEXT</td></tr>
<tr><td>TIME</td><td>none</td><td>special type — displays current time</td></tr>
<tr><td>CURR PROGRAM</td><td>none</td><td>special type — displays currently running program</td></tr>
<tr><td>CURR ROUTINE</td><td>none</td><td>special type — displays currently running routine</td></tr>
<tr><td>TIMER (SEC)</td><td>Prog. Name, Var. Name</td><td>Displays KAREL timer in seconds</td></tr>
<tr><td>TIMER (MIN)</td><td>Prog. Name, Var. Name</td><td>Displays KAREL timer in minutes</td></tr>
<tr><td>TIMER (HR)</td><td>Prog. Name, Var. Name</td><td>Displays KAREL timer in hours</td></tr>
<tr><td>PROG STATUS</td><td>none</td><td>special type — displays status of current program: RUNNING, MOVING, PAUSED, WAITING, WAITING FOR DI[ ], WAITING FOR RI[ ], ABORTED</td></tr>
<tr><td colspan="3">For DI[ ], DO[ ], RI[ ], and RO, the variable data types are as follows:
• INTEGER— displays as 1, 0
• BOOLEAN— displays as TRUE, FALSE
• TEXT — displays as ON, OFF</td></tr>
</table>

Table 10–51. Status Menu Variable Detail Information (Cont'd)

ITEM	DESCRIPTION
REG + I/O	This item allows you to specify the number of the register or the I/O signal, when the variable type is R[ ], DI[ ], DO[ ], RI[ ], RO[ ], GI[ ], or GO[ ].
KAREL Program	This item allows you to specify the name of the KAREL program that contains the variable you are displaying, when the variable type is KAREL VAR.
Variable Name	This item allows you to specify the name of the variable you are displaying.
System Variable	This item allows you to specify the name of the system variable you are displaying, when the variable type is SYSTEM VAR.
Data Type	This item allows you to specify the variable data types: • NONE • INTEGER • REAL • TEXT • BOOLEAN

Examples

See Status Menu(menu) Macro Example 1 and Status Menu(menu) Macro Example 2 for examples of how to use the Status Menu(menu) macro instruction in a teach pendant program.

Status Menu(menu) Macro Example 1

```
PROG  TEST1
   1: Status Menu(2)
```

Status Menu(menu) Macro Example 2

```
PROG  TEST2
   1: Status Menu('ProductionStatus)
```

Setup Procedure

Use Procedure 10-44 to create a status menu.

Table 10–52. Status Menu Screen Items

ITEM	DESCRIPTION
Item	This item refers to item numbers (1-99) always in sequence.
Menu number	This item refers to unique menu numbers (1-99).
Menu name	This item is the name of the menu.

Table 10–53. Status Menu DETAIL Screen Items

ITEM	DESCRIPTION
Status menu no:	This item refers to unique menu numbers (1-99).
Menu name	This item is the name of the menu.
Line	This item is the line number for the status items. Maximum of nine items
Text	This item is the display text used in the prompt box.

Table 10–54. Status Menu Sub-DETAIL Screen Items

ITEM	DESCRIPTION
Display Text	This item allows you to specify the text that will be displayed for the status item. You can [TYPE] up to 17 characters of display text.

Table 10–54. Status Menu Sub-DETAIL Screen Items (Cont'd)

Variable [TYPE]	This item allows you to specify the kind of variable for the status item:		
	[TYPE] of Variable	What to Define	Data Display [TYPE]
	NONE	none	none
	R[ ]	REG Number	INTEGER, REAL, BOOLEAN
	DIN[ ]	I/O Number	INTEGER, BOOLEAN, TEXT
	DOUT[ ]	I/O Number	INTEGER, BOOLEAN, TEXT
	RDI[ ]	I/O Number	INTEGER, BOOLEAN, TEXT
	RDO[ ]	I/O Number	INTEGER, BOOLEAN, TEXT
	GIN[ ]	I/O Number	INTEGER
	GOUT[ ]	I/O Number	INTEGER
	KAREL VAR	PROG. Name, Var. Name	INTEGER, REAL, BOOLEAN, TEXT
	SYSTEM VAR	System Variable Name	INTEGER, REAL, BOOLEAN, TEXT
	TIME	none	special [TYPE] — displays current time
	CURR PROGRAM	none	special [TYPE] — displays currently running program
	CURR ROUTINE	none	special [TYPE] — displays currently running routine
	TIMER (SEC)	Prog. Name, Var. Name	Displays KAREL timer in seconds
	TIMER (MIN)	Prog. Name, Var. Name	Displays KAREL timer in minutes
	TIMER (HR)	Prog. Name, Var. Name	Displays KAREL timer in hours
	PROG STATUS	none	special [TYPE] - displays status of current program: RUNNING, MOVING, PAUSED, WAITING, WAITING FOR DI[], WAITING FORRI[], AT HOME, AT REPAIR, ABORTED

Table 10–54. Status Menu Sub-DETAIL Screen Items (Cont'd)

	For DI[], DO[], RI[], and RO[], the variable data [TYPE]s are as follows: • INTEGER - displays as 1, 0 • BOOLEAN - displays as TRUE, FALSE • TEXT - displays as ON, OFF
REG + I/O	This item allows you to specify the number of the register or the I/O signal, when the variable [TYPE] is R[], DI[], DO[], RI[], RO[], GI[], or GO[].
KAREL Program	This item allows you to specify the name of the KAREL program that contains the variable you are displaying, when the variable [TYPE] is KAREL VAR.
Variable Name	This item allows you to specify the name of the variable you are displaying.
System Variable	This item allows you to specify the name of the system variable you are displaying, when the variable [TYPE] is SYSTEM VAR.
Data [TYPE]	This item allows you to specify the variable data [TYPE]s: • NONE • INTEGER • REAL • TEXT • BOOLEAN

Procedure 10-44 Creating a Status Menu

Conditions

- The default device is set. Refer to Procedure 9-1 .

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Menu Utility. You will see a screen similar to the following.

```
SETUP Menu Utility
Choose a menu type:
  1      Prompt box msg
  2      Prompt box yes/no
  3      Select from a list
  4      Status menus
  5      Operator entry
```

5. To display menu tips:

Menu tips are prompt boxes that are displayed the first time you press the menu DETAIL key for each menu type. The prompt box contains information that describes the menus, macros, and parameters.

a. Press F4, CONFIG. You will see a screen similar to the following.

```
SETUP Menu Utility
Config Menus
 1   Show menu tips:              YES
```

b. **To turn off menu tips** , press F5, NO.

To turn on menu tips , press F4, YES. You will see a screen similar to the following.

```
These menus are used with
         macro Status Menu.
         Parameter 1 selects the menu.
                     OK
```

When you are finished viewing menu tips , press ENTER to select OK.

6. Move the cursor to Status menus and press F2, DETAIL. You will see a screen similar to the following.

```
SETUP Menu Utility
Status menus
ITEM Menu no     Menu name
 1      1        Status Page 1
 2      2        Status Page 2
```

7. To create a new menu:

a. Press F3, CREATE.

b. Type the number of the new menu and press ENTER. You will see a screen similar to the following.

```
SETUP Menu Utility
 Status menu no      1
  Menu name/Title:   ProductionStatus
     Line   Text
 1  Display : Current Program
 2  Display : Current Routine
 3  Display : Status
 4  Display : Current time
 5  Display : Execution time
 6  Display :
 7  Display :
 8  Display :
```

c. Move the cursor to Menu name and press ENTER. Use the appropriate function keys to name the menu and press ENTER.

d. Move the cursor to each item and press F2, DETAIL. You will see a screen similar to the following.

```
SETUP Menu Utility
 Status menu no      1
  Menu name:ProductionStatus  Line no.:2
 1  Text:                    Current Routine
 2  Var Type:                CURR ROUTINE
 3  REG + I/O:               0
 4  KARELProg:
 5  Var Name:
 6  SysVar:
 7  DataType:                TEXT
```

e. Select each item and set as desired. To display the next variable, press F2, NEXT V. To display the previous variable, press F3, PREV V. Refer to Table 10–51 for information on how to set each item. When you are finished setting variable detail information, press PREV.

f. **To test the menu you created** , press F3, TEST.

The status menu's display variables are checked to make sure that the proper information was specified and that the variables exist.

If no errors were found , you will see the following screen briefly.

```
Status menu no       1
         Checking variable definitions
   No errors found in menu definitions !
  Testing menu...
```

When testing is finished, you will see a screen similar to the following.

```
STATUS Menu
                  Production Status
    Program:                TESTIII
    Routine:                TESTIII
    Status:                 RUNNING
    Part Count              101
    Part on Conv1           ON
    Pallet Count            15
```

If errors were found , they will be listed on the teach pendant screen and the menu will not be displayed. The previous menu will be displayed. Correct the variable definitions and press F3, TEST, again.

8. To modify an existing menu:

a. Move the cursor to the menu name and press F2, DETAIL. You will see a screen similar to the following.

```
SETUP Menu Utility
Status menus
ITEM Menu no      Menu name
 1       1        Status Page 1
 2       2        Status Page 2
```

b. Move the cursor to the lines you want to change, press F2, DETAIL, and make the changes you want.

9. To save (back up) a menu to the default device

a. Set the default device. Refer to Procedure 9-1 in Chapter 9 *PROGRAM AND FILE MANIPULATION* .

b. Move the cursor to the name of the menu you want to save.

c. Press NEXT, >, and press F1, SAVE. The menu will be saved to the default device.

Note If the file already exists on the default device, you will be prompted whether to replace the existing file with the one you are backing up. To replace the file, select YES. To cancel, select NO.

Note You can also back up a menu using FILE, BACKUP APPLICATION. Refer to Chapter 9 *PROGRAM AND FILE MANIPULATION* for more information.

Note Each status menu is stored in its own variable file that begins with the prefix " **MENU4** ." The next two characters of the filename are the menu number. For example, the status menu 1 is stored in the variable file named "MENU4 **01** .VR."

10. To load (restore) a menu:

a. Move the cursor to the name of the menu you want to load.

b. Press NEXT, >, and press F2, LOAD. The menu will be loaded from the default device.

Note If the file already exists on the controller, you will be prompted whether to replace the existing file with the one you are loading. To replace the file, select YES. To cancel, select NO.

Note You can also load a menu using FILE, LOAD, *.VR. Refer to Chapter 9 *PROGRAM AND FILE MANIPULATION* for more information.

11. To clear a menu:

a. Move the cursor to the name of the menu you want to clear.

b. Press NEXT, >, and press F3, CLEAR.

c. Confirm the clear:

- To clear the menu, select YES and press ENTER.

- To cancel the clear, select NO and press ENTER.

12. To copy a menu:

a. Move the cursor to the name of the menu you want to copy.

b. Press NEXT, >, and press F4, COPY.

c. Type the menu number to which you want to copy the menu and press ENTER.

Note If the new menu already exists, you will be prompted whether to replace the existing menu with the one you are copying. To replace the file, select YES. To cancel, select NO.

10.26.6 Operator Entry Menu

An operator entry menu requires the operator to enter INTEGER, REAL, BOOLEAN, or text values. When an operator entry menu is executed, the program pauses until the operator types the information and then presses F3, DONE.

See Figure 10–99 for an example of an operator entry menu.

Figure 10–99. Operator Entry Menu

```
USER
              Select Tool
    1 Tool number to use:        2
    2 Number to pickup:          1
    Press DONE to continue.
```

The operator uses the:

- Numeric keys to type numbers
- Function keys to select YES or NO
- ENTER key and function keys to type letters

Macro: Op. Entry Menu(menu)

An operator entry menu uses the Op. Entry Menu(menu) macro.

Parameter 1: menu - This is the menu number or menu name. The menu can be chosen by number or name.

Display

To display an operator entry menu you create, use the Op. Entry Menu(menu) macro. The menu will be displayed automatically on the USER screen. If the operator changes the teach pendant display to another screen during teach pendant execution of this menu but before pressing DONE, the operator can press MENUS and select USER to display this screen again.

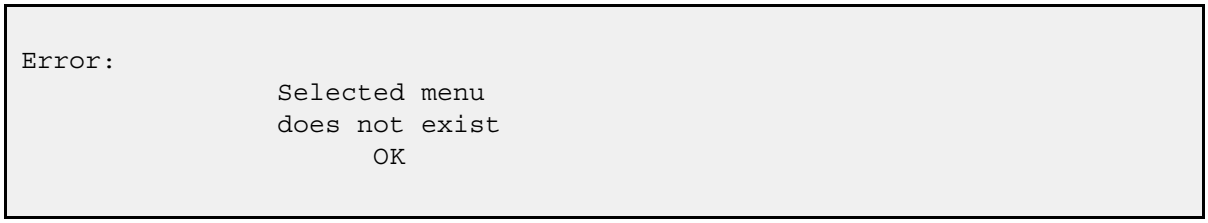
Error

The following errors are possible:

- If the Op. Entry Menu(men) macro was used in a teach pendant program without a parameter, an alarm is given and the program is aborted.

- If the selected operator entry menu does not exist or the menu is the wrong kind of menu when it is called from a teach pendant program, a prompt box similar to the one shown in Figure 10–100 will be displayed. To abort the program, select OK and press ENTER.

Figure 10–100. Operator Entry Menu Does Not Exist Screen



Menu File Name

Each operator entry menu is stored in its own variable file that begins with the prefix " **MENU5** ." The next two characters of the filename are the menu number. For example, the operator entry menu 1 is stored in the variable file named "MENU5 **01** .VR."

Setup

In an operator entry menu, you specify

- Up to ten prompt items
- The names of the variables that correspond to the prompt items, and detailed variable information

Table 10–55 lists and describes the variable detail information.

Table 10–55. Operator Entry Menu Variable Detail Information

ITEM	DESCRIPTION
Prompt	This item allows you to specify the text that will be displayed for the prompt item. You can type up to 17 characters of display text.

Table 10–55. Operator Entry Menu Variable Detail Information (Cont'd)

<table>
<tr><th>ITEM</th><th colspan="3">DESCRIPTION</th></tr>
<tr><td rowspan="5">Variable Type</td><td colspan="3">This item allows you to specify the kind of variable for the value of the prompt item:</td></tr>
<tr><td>Type of Variable</td><td>What to Define</td><td>Data Display Type</td></tr>
<tr><td>NONE</td><td>none</td><td>none</td></tr>
<tr><td>R[ ]</td><td>REG Number</td><td>INTEGER, REAL, BOOLEAN</td></tr>
<tr><td>KAREL VAR</td><td>Prog. Name, Var. Name</td><td>INTEGER, REAL, BOOLEAN, TEXT</td></tr>
<tr><td></td><td>SYSTEM VAR</td><td>System Variable</td><td>INTEGER, REAL, BOOLEAN, TEXT</td></tr>
<tr><td>REG Number</td><td colspan="3">This item allows you to specify the number of the register, when the variable type is R[ ].</td></tr>
<tr><td>KAREL Program</td><td colspan="3">This item allows you to specify the name of the KAREL program that contains the variable you are using, when the variable type is KAREL VAR.</td></tr>
<tr><td>Variable Name</td><td colspan="3">This item allows you to specify the name of the KAREL variable you are using, when the variable type is KAREL VAR.</td></tr>
<tr><td>System Variable</td><td colspan="3">This item allows you to specify the name of the system variable you are using, when the variable type is SYSTEM VAR.</td></tr>
<tr><td>Data Type</td><td colspan="3">This item allows you to specify the variable data types:
NONE
INTEGER
REAL
TEXT
BOOLEAN</td></tr>
<tr><td>Variable Minimum</td><td colspan="3">This item allows you to specify the minimum value allowed for the INTEGER or REAL variable.</td></tr>
<tr><td>Variable Maximum</td><td colspan="3">This item allows you to specify the maximum value allowed for the INTEGER or REAL variable.</td></tr>
</table>

Examples

See Op Entry Menu(menu) Macro Example 1 and Op. Entry Menu(menu) Macro Example 2 for examples of how to use the Op. Entry Menu(menu) macro instruction in a teach pendant program.

Op Entry Menu(menu) Macro Example 1

```
PROG  TEST1
                                        1: Ops. Entry Menu(2)
```

Op. Entry Menu(menu) Macro Example 2

```
PROG  TEST2
                                        1: Ops. Entry Menu('SetupTool')
```

Setup Procedure

Table 10–56 through Table 10–58 list the Operator Entry Menu Screen setup items.

Use Procedure 10-45 to create an operator entry menu.

Table 10–56. Operator Entry Menu Screen Items

ITEM	DESCRIPTION
Item	This item refers to the ITEM number that you selected.
Menu number	This item refers to the menu number you selected
Menu name	This item refers to the name of the menu you selected

Table 10–57. Operator Entry Menu DETAIL Screen Items

ITEM	DESCRIPTION
OP Entry Menu no	This item is the number of the menu (unique ID).
Title	This item is the name of the menu.
Prompt	This item has up to ten prompts.

Table 10–58. Operator Entry Menu Sub-DETAIL Screen Items

ITEM	DESCRIPTION
Prompt	This item allows you to specify the text that will be displayed for the prompt item. You can enter up to 17 characters of display text.

Table 10–58. Operator Entry Menu Sub-DETAIL Screen Items (Cont'd)

Variable [TYPE]	This item allows you to specify the kind of variable for the value of the prompt item: **[TYPE] of Variable What to Define Data Display [TYPE]** NONE none none R[ ] REG Number INTEGER, REAL, BOOLEAN KAREL VAR Prog. Name, Var. Name INTEGER, REAL, BOOLEAN, TEXT SYSTEM VAR System Variable Name INTEGER, REAL, BOOLEAN, TEXT
REG Number	This item allows you to specify the number of the register, when the variable [TYPE] is R[].
KAREL Program	This item allows you to specify the name of the KAREL program that contains the variable you are using, when the variable [TYPE] is KAREL VAR.
Variable Name	This item allows you to specify the name of the KAREL variable you are using, when the variable [TYPE] is KAREL VAR.
System Variable	This item allows you to specify the name of the system variable you are using, when the variable [TYPE] is SYSTEM VAR.
Data [TYPE]	This item allows you to specify the variable data [TYPE]s: • NONE • INTEGER • REAL • TEXT • BOOLEAN
Variable Minimum	This item allows you to specify the minimum value allowed for the INTEGER or REAL variable.
Variable Maximum	This item allows you to specify the maximum value allowed for the INTEGER or REAL variable.

Procedure 10-45 Creating an Operator Entry Menu

Conditions

- The default device is set. Refer to Procedure 9-1 .

Steps

1. Press MENUS.

2. Select SETUP.
3. Press F1, [TYPE].
4. Select Menu Utility. You will see a screen similar to the following.

```
SETUP Menu Utility
Choose a menu type:
 1       Prompt box msg
 2       Prompt box yes/no
 3       Select from a list
 4       Status menus
 5       Operator entry
```

5. **To display menu tips**

   Menu tips are prompt boxes that are displayed the first time you press the menu detail key for each menu type. The prompt box contains information that describes the menus, macros, and parameters.

   a. Press F4, CONFIG. You will see a screen similar to the following.

```
SETUP Menu Utility
Config Menus
 1   Show menu tips:                YES
```

   b. **To turn off menu tips** , press F5, NO.

   To turn on menu tips , press F4, YES. You will see a screen similar to the following.

```
These menus are used with
        macro Op. Entry Menu.
        Parameter 1 selects the menu.
                     OK
```

   When you are finished viewing menu tips , press ENTER to select OK.

6. Move the cursor to Operator entry and press F2, DETAIL. You will see a screen similar to the following.

```
SETUP Menu Utility
Operator entry menus
ITEM Menu no      Menu name
 1       1        Chute
 2       2        SelectTool
```

7. To create a new menu:

a. Press F3, CREATE.

b. Type the number of the new menu and press ENTER. You will see a screen similar to the following.

```
SETUP Menu Utility
Op. Entry menu no:      2
 Menu name:  MoveMenu
   Title    Select tool
 1 Prompt:
 2 Prompt:
 3 Prompt:
 4 Prompt: Tool number to use:
 5 Prompt: Number to pickup:
 6 Prompt:
 7 Prompt:
 8 Prompt:
 9 Prompt:
10 Prompt:
```

c. Move the cursor to Menu name and press ENTER. Use the appropriate function keys to name the menu and press ENTER.

d. Move the cursor to each item and press F2, DETAIL. You will see a screen similar to the following.

```
SETUP Menu Utility
Op. Entry menu no:     2
 Menu name:  SelectTool   Line no.:4
 1 Prompt:        Tool number to use:
 2 Var. Type:     R[ ]
 3 REG. no:       1
 4 KARELProg:
 5 Var Name:
 6 SysVar:
 7 DataType:      INTEGER
 8 Var Min.:      1
 9 Var Max.:      5
```

e. Select each item and set as desired. To display the next variable, press F2, NEXT V. To display the previous variable, press F3, PREV V. Refer to Table 10–55 for information on how to set each item. When you are finished setting variable DETAIL information, press PREV.

f. **To test the menu you created** , press F3, TEST.

The operator entry menu's display variables are checked to make sure that the proper information was specified and that the variables exist.

If no errors were found , you will see the following screen briefly.

```
Status menu no      1
         Checking variable definitions
  No errors found in menu definitions !
  Testing menu...
```

When testing is finished, you will see a screen similar to the following.

```
USER
              Selects Tool
    1 Tool number to use:       2
    2 Number to pick up:        1
    Press DONE to continue
```

If errors were found , they will be listed on the teach pendant screen and the menu will not be displayed. Correct the variable definitions and press F3, TEST, again.

Note If the variable information is not corrected before the menu macro is executed from a teach pendant program, the incorrect variables will be set to undefined and the values will not be able to be changed.

8. To modify an existing menu:

a. Move the cursor to the menu name and press F2, DETAIL. You will see a screen similar to the following.

```
SETUP Menu Utility
Status menus
ITEM Menu no     Menu name
 1      1        Status Page 1
 2      2        Status Page 2
```

b. Move the cursor to the lines you want to change, press F2, DETAIL, and make the changes you want.

9. To save (back up) a menu to the default device:

a. Set the default device. Refer to Procedure 9-1 in Chapter 9 *PROGRAM AND FILE MANIPULATION* .

b. Move the cursor to the name of the menu you want to save.

c. Press NEXT, >, and press F1, SAVE. The menu will be saved to the default device.

Note If the file already exists on the default device, you will be prompted whether to replace the existing file with the one you are backing up. To replace the file, select YES. To cancel, select NO.

Note You can also back up a menu using FILE, BACKUP APPLICATION. Refer to Chapter 9 *PROGRAM AND FILE MANIPULATION* for more information.

Note Each operator entry menu is stored in its own variable file that begins with the prefix " **MENU5** ." The next two characters of the filename are the menu number. For example, the operator entry menu 1 is stored in the variable file named "MENU5 **01** .VR."

10. To load (restore) a menu:

a. Move the cursor to the name of the menu you want to load.

b. Press NEXT, >, and press F2, LOAD. The menu will be loaded from the default device.

Note If the file already exists on the controller, you will be prompted whether to replace the existing file with the one you are loading. To replace the file, select YES. To cancel, select NO.

Note You can also load a menu using FILE, LOAD, *.VR. Refer to Chapter 9 *PROGRAM AND FILE MANIPULATION* for more information.

11. To clear a menu:

a. Move the cursor to the name of the menu you want to clear.

b. Press NEXT, >, and press F3, CLEAR.

c. Confirm the clear:

- To clear the menu, select YES and press ENTER.

- To cancel the clear, select NO and press ENTER.

12. To copy a menu:

a. Move the cursor to the name of the menu you want to copy.

b. Press NEXT, >, and press F4, COPY.

c. Type the menu number to which you want to copy the menu and press ENTER.

Note If the new menu already exists, you will be prompted whether to replace the existing menu with the one you are copying. To replace the file, select YES. To cancel, select NO.

10.26.7 Using Your Teach Pendant Menus

After you have set up a menu, you can use the menu macro instruction to display it.

When you edit a teach pendant program to use a teach pendant menu, you select the name of the menu from a sub-window. In the sub-window, you will see a list of **all** menus that were created using the Menu Utility SETUP screens. This list is updated at Cold start and when menus are added and deleted on the SETUP screen. Because there are five kinds of menus and each kind of menu can be used only with its own specific macro, you must choose the appropriate menu from the list.

A maximum of 20 menu names will be listed. If the menu you need is not listed, you can type its name or number manually with the teach pendant number and letter keys.

Use Procedure 10-46 to use your teach pendant menus.

Procedure 10-46 Using Your Teach Pendant Menus

Conditions

- You have created a teach pendant program.

Steps

1. Select the program you want to edit from the Select menu.
2. Press ENTER.
3. Insert the menu macro instruction you want.
4. Press the right arrow key to move the cursor to the end of the instruction as shown in the following screen.

```
...
5:  Prompt Box Msg
...
```

5. Press F4, [CHOICE].
6. Select String and then select Menus.
7. Select the menu you want.
   - To select a menu from a list of already defined menus, move the cursor to the string you want and press ENTER.
   - **If the menu you want is not listed** , press F5, String, press the appropriate function keys to type the string, and press ENTER.

10.27 MIRROR IMAGE UTILITY

The mirror image utility allows you to translate an entire teach pendant program or portion of a teach pendant program to mirror image the original programmed points. This option can be used to teach symmetrical parts easily.

Note For PaintTool and paint robots, this is almost always accomplished using left-hand and right-hand robots. Use of this utility is not recommended.

Mirror imaging of a program can be accomplished either as a

- Parallel mirror image
- Parallel and rotational mirror image

Refer to Table 10–59 for information on Mirror Image Shift screen items.

Table 10–59. MIRROR IMAGE SHIFT Screen Items

ITEM	DESCRIPTION
Original Program	This item is the name of the program that you want to shift using mirror image. To select a program, press ENTER, type the name of the program, and press ENTER again.
Range	This item indicates whether the whole program or a portion will be shifted using mirror image.
Start line	This item is the first line in the program that will be shifted using mirror image.
End line	This item is the last line in the program that will be shifted using mirror image.
New Program	This item is the name of the program that results when you offset or shift the Original Program. If you want the resulting offset or shifted program to replace the Original Program, make the New Program name the same as the Original Program name.
Insert line	This item indicates the location in an existing program where new shifted position information will be inserted.
Position data	This item is the current x, y, and z position information.
Rotation	This item, if set to ON, indicates that you want to rotate three position about an axis. If set to OFF, you can shift a single source position to a single destination position.
Source position	This item is the original position that you will be shifting using mirror image. If Rotation is set to ON, there will be three source positions.
Destination position	This item is the destination position to which you will be shifting using mirror image. If Rotation is set to ON, there will be three destination positions.

Parallel Mirror Image

A parallel mirror image mirrors the program about a mirror plane without an offset or a rotation. See Figure 10–101 and Figure 10–102 .

Figure 10–101. Parallel Mirror Image with Mirror Plane in Center of Robot

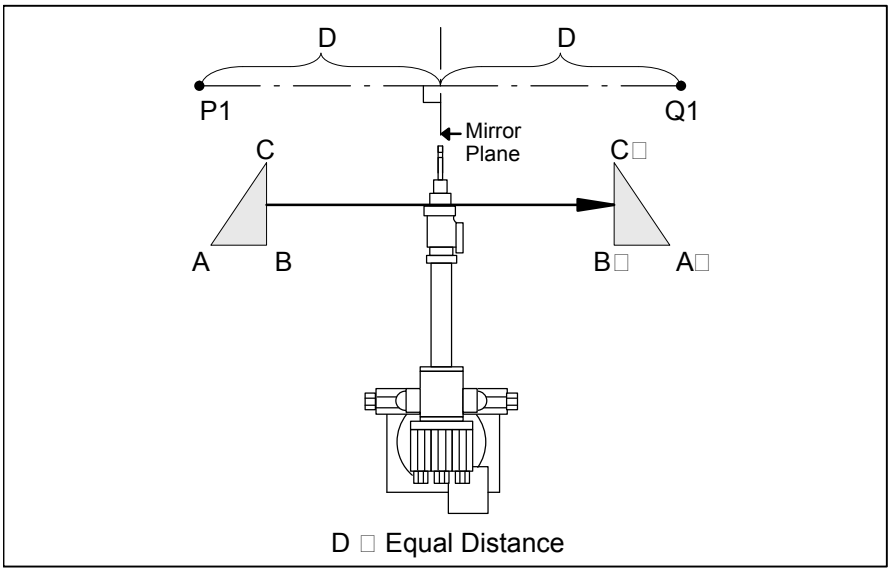
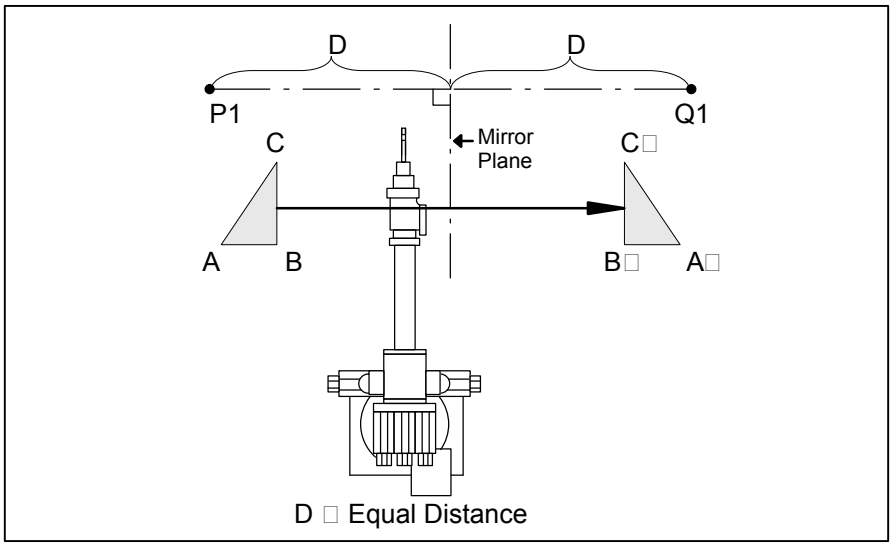


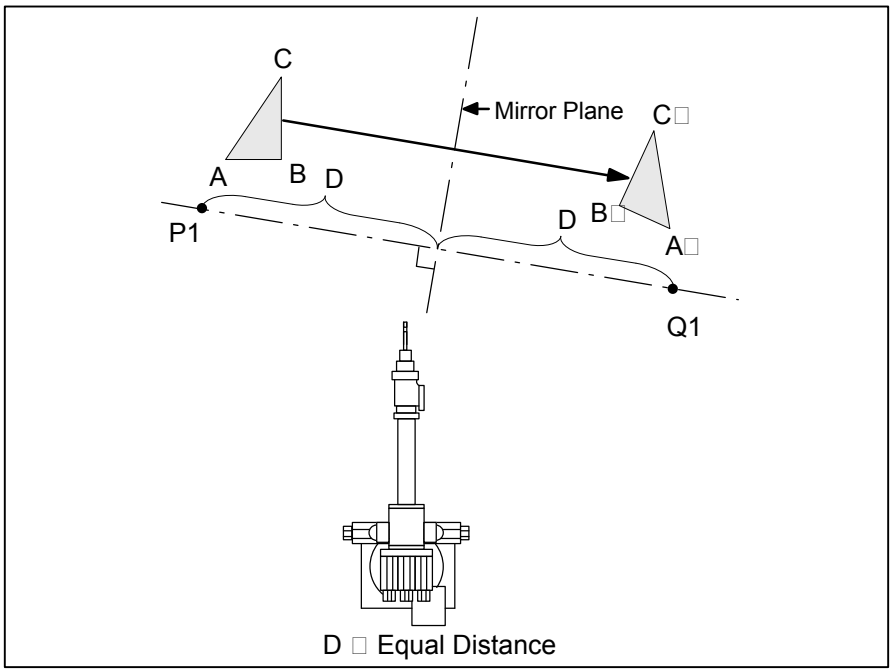
Figure 10–102. Parallel Mirror Image with Mirror Plane Offset from Center of Robot



Caution

To be sure the parallel mirror image works correctly, you must have an exact TCP. If you do not, the resulting mirror image program will contain an offset value. See Figure 10–103 .

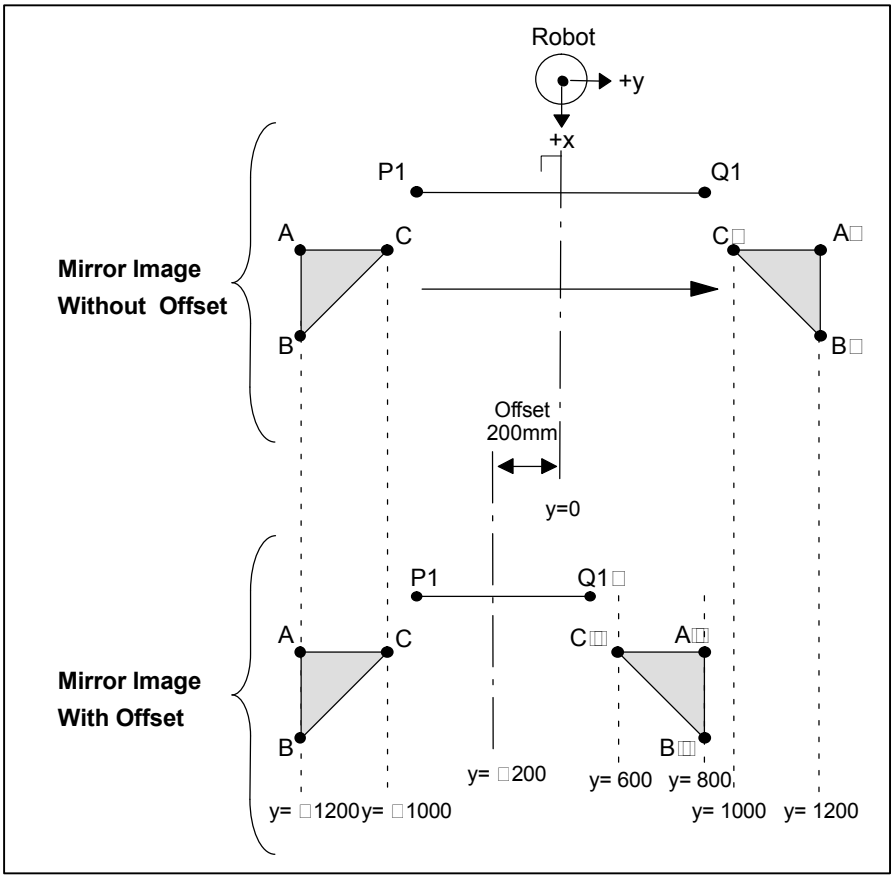
Figure 10–103. Parallel Mirror Image with Offset



Mirror Image Example

Figure 10–104 displays a mirror image of A, B, and C to A', B', and C' when P1 and Q1 are taught positions. Figure 10–104 also displays a mirror image with an offset when P1 and Q1 are taught but Q1 is taught at an offset of 200mm. In this case, the result is A'', B'', and C''.

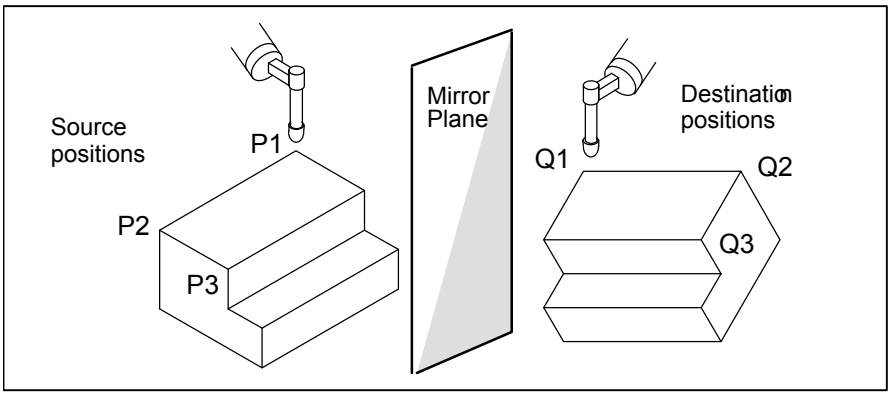
Figure 10–104. Positional Mirror Image



Rotational Mirror Image

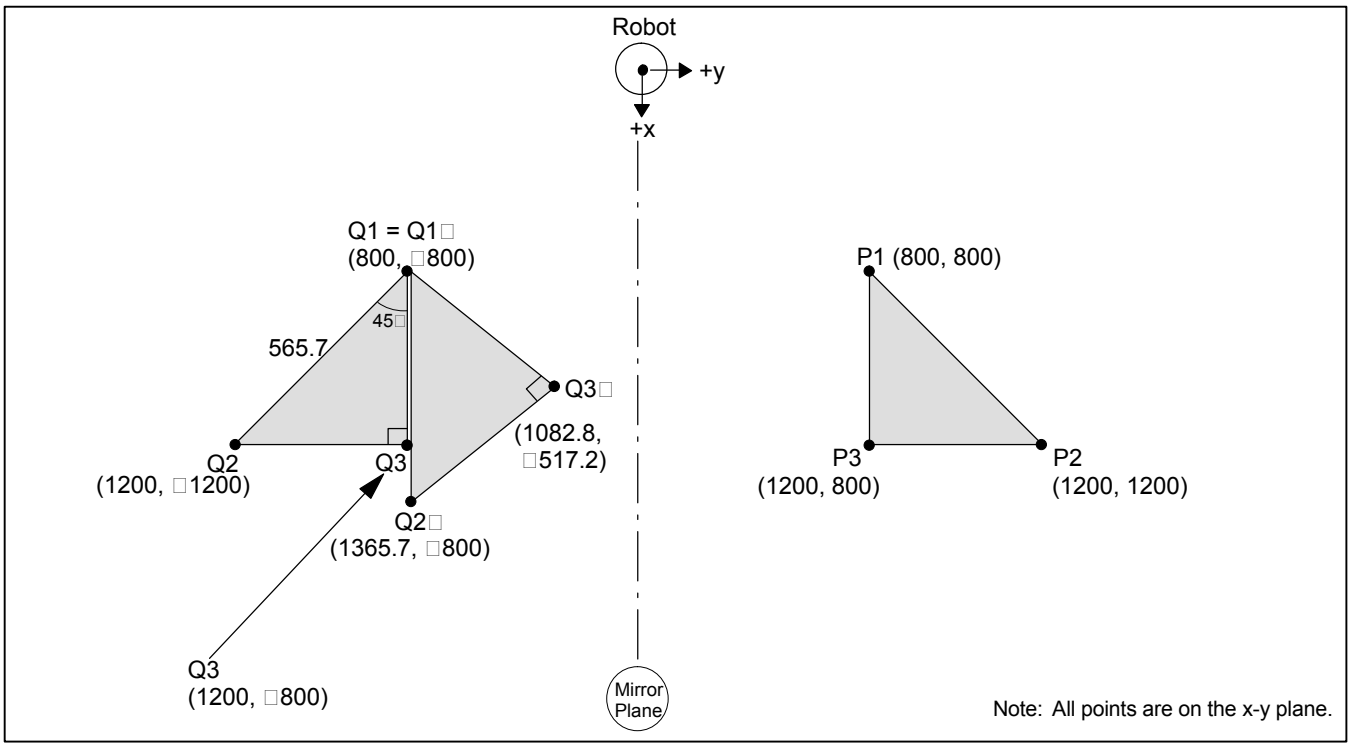
A parallel mirror image mirrors the program about a mirror plane first, and then the mirrored program is rotated about a center of rotation. Orientation of the part to be imaged is rotated about one or more of its axes relative to the mirror plane. See Figure 10–105 .

Figure 10–105. Rotational Mirror Image



In Figure 10–106 the positions, P1, P2 and P3 are mirrored about the mirror plane as Q1, Q2 and Q3. These positions are then rotated 45°about Q1 and stored as Q1', Q2' and Q3'.

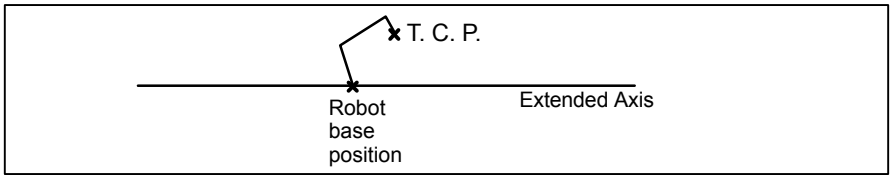
Figure 10–106. Rotational Mirror Image



Mirror Image of Extended Axes

Extended (Ext) axes determines how the mirror image function will translate the program when you are using extended axes. Figure 10–107 displays a key for use with Figure 10–108 through Figure 10–110 .

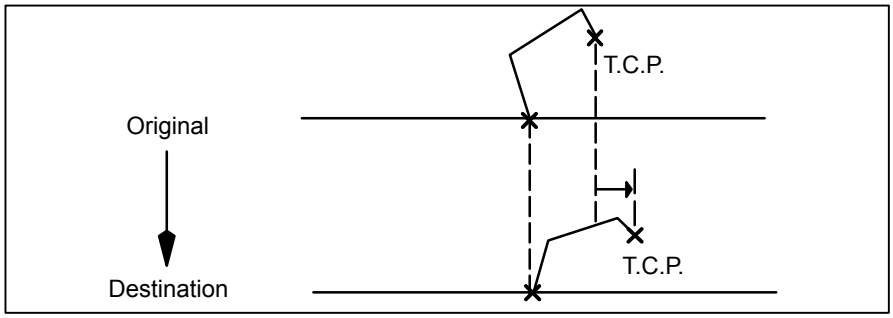
Figure 10–107. Mirror Image Key



Possible extended axes configurations are:

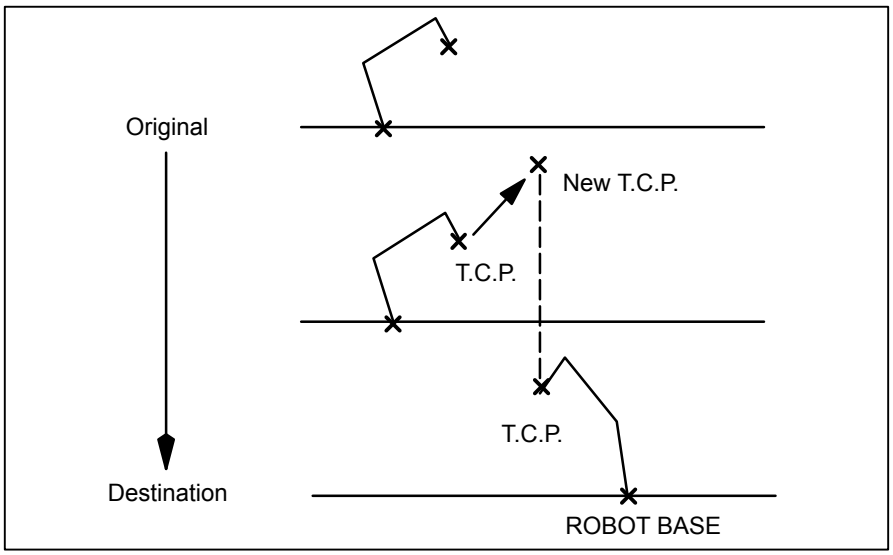
- **Robot axes only** - allows you to mirror the axes of the robot without mirroring any non-integrated extended axes such as a positioning table. The shift is calculated using the change in the robot tool center point (TCP).

Figure 10–108. Example of Robot Axes Only Mirror Image



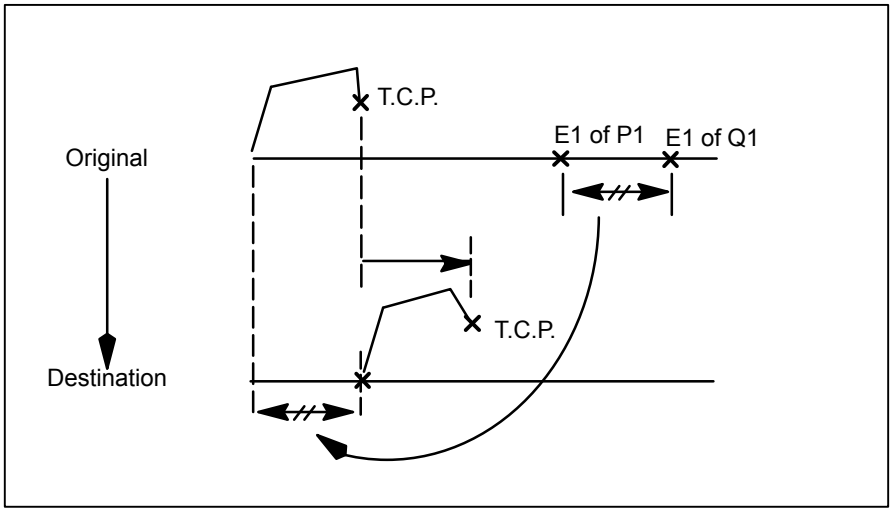
- **Ext integrated** - allows you to mirror the axes of the robot and any integrated axes. The amount of mirror image for the robot and the extended axes is calculated using the change in the TCP. See Figure 10–109 .

Figure 10–109. Example of Extended Axes Integrated Mirror Image



- **With ext axes** - allows you to mirror positions for robot axes and any extended axes in your system. The amount of mirror image for the robot is calculated by using the change in the TCP. The amount of shift for the extended axes is calculated using the center of the difference between an original position (P1) and a new position (Q1) as the point where the mirror image occurs. See Figure 10–110 .

Figure 10–110. Example of With Extended Axes Mirror Image



- **Ext axes only** - Ext axes only is not available for a mirror shift.
- **Replace Ext axes** - Replace Ext axes is not available for a mirror shift.

Use Procedure 10-47 to perform a mirror image of a program.

Procedure 10-47 Using Mirror Image

Conditions

- The program you want to mirror has been created and contains recorded positions.
- All robot joint axes are at zero degrees.
- If you are using PaintTool applications and are using mirror image to transfer positions between right and left hand robots, be sure you are performing the mirror image on the **destination** robot and not the source robot. Therefore, you must first copy the program from the source robot to the destination robot before you perform the mirror image.

Steps

1. Press MENUS.
2. Select UTILITIES.
3. Press F1, [TYPE].
4. Select Mirror Image. You will see the Mirror Image Shift screen.
5. Move the cursor to Original Program. If the program you want to mirror is not selected, press ENTER. Use the appropriate function keys to type the name of the program and press ENTER.

   Note The last program selected using the SELECT menu will automatically be named as the original program.

6. Move the cursor to Range and select to mirror the WHOLE program or PART of the program.

   - **To mirror the whole program, press F5, WHOLE.** You will see a screen similar to the following.

```
MIRROR IMAGE SHIFT
PROGRAM
 1 Original Program:              [STYLE37]
 2   Range:                          WHOLE
 3   Start line: (not used)           ****
 4   End line: (not used)             ****
 5 New Program:                   [       ]
 6  Insert line: (not used)           ****
 7 EXT axes:             Robot axes only
To move page with SHIFT + DOWN, SHIFT + UP
```

Note EXT axes will only be displayed if you are using mirror image for extended axes.

- **To shift part of the program, press F4, PART.** You will see a screen similar to the following.

```
MIRROR IMAGE SHIFT
PROGRAM
 1 Original Program:              [STYLE37]
 2   Range:                          PART
 3   Start line:                        0
 4   End line:                          0
 5 New Program:                   [       ]
 6  Insert line: (not used)           ****
 7 EXT axes:             Robot axes only
To move page with SHIFT + DOWN, SHIFT + UP
```

7. If you selected to mirror PART of a program,

- Move the cursor to Start line and type the starting line number. Press ENTER.
- Move the cursor to End line and type the ending line number. Press ENTER.

8. Move the cursor to New Program and press ENTER. Use the appropriate function keys to type the name of the new program and press ENTER. This is the program to which you will be mirroring the positions.

Note You can mirror the positions of an entire program or portion of a program from within a program or from one program to another. See Figure 10–111 and Figure 10–112 .

Figure 10–111. Mirroring an Entire Program

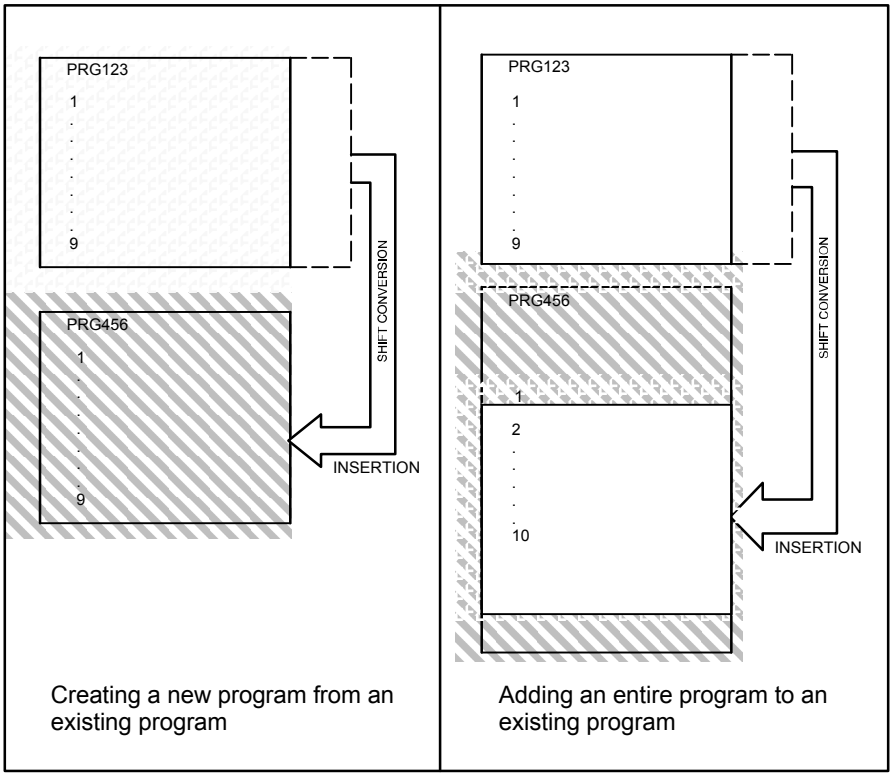
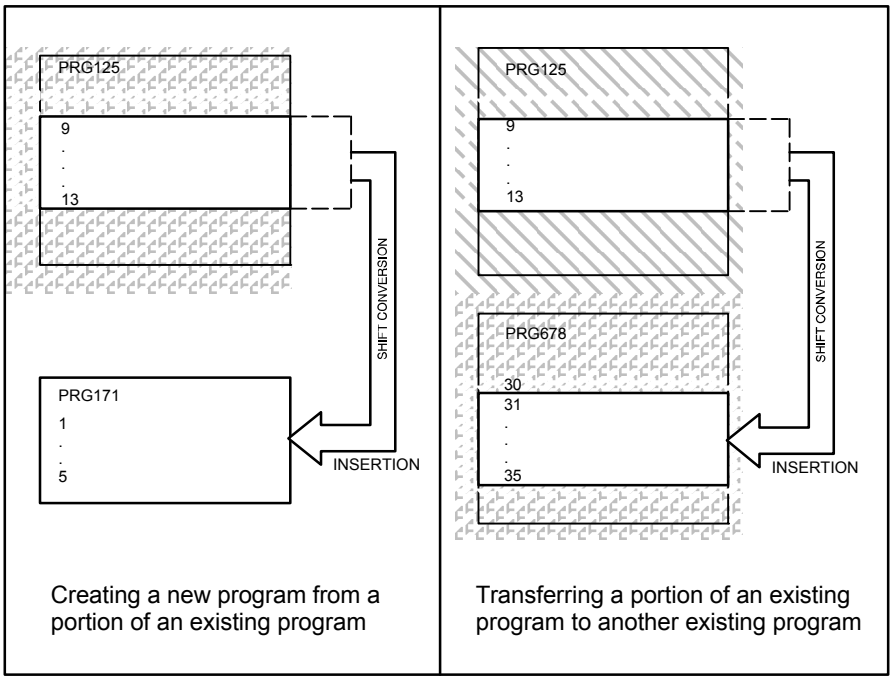


Figure 10–112. Mirroring a Portion of a Program



Note The New Program can be the same as the original program, an already existing program, or a program that does not exist.

- **If you are inserting lines into the original program** , type in the name of the program and the corresponding line numbers.
- **If you are inserting lines into an existing program,** move the cursor to Insert line and type the line number at which you want to insert the shifted information.

9. Press the down arrow key. You will see the Mirror Image Shift (Position) screen.

10. Move the cursor to rotation. You will see a screen similar to the following.

```
MIRROR IMAGE SHIFT
Shift amount/Teach
  Position data
  X :********  Y :********  Z :********
  1 Rotation:                        OFF
  2 Source position:         P1:
  3 Destination position:    Q1:
```

HINTS: P1 and Q1 should not be part of the program to be mirrored. They should be new positions that are located at an equal distance from the mirror plane. (See Figure 10–101 .) Use the following guidelines to teach these positions:

a. Jog the robot to zero degrees by matching up the witness marks on the robot (if they are available), or by displaying the POSITION screen.

b. Jog the robot:

- Set the jog coordinate system to WORLD.
- Jog the robot in +Y by a known distance, 200 mm for example.

OR

- Set the jog coordinate system to JOINT.
- Jog the robot a known angle, 20°, for example.

c. Record this position as P1.

d. Jog the robot back to zero.

e. Jog the robot in the opposite direction by the exact same distance from the mirror plane as you jogged the robot in Step 10b .

f. Record this position as Q1. See Figure 10–101 .

11. If you do not want to rotate the positions,

a. Press F5, OFF. You will see a screen similar to the following.

```
MIRROR IMAGE SHIFT
Shift amount/Teach
  Position data
  X :********  Y :********  Z :********
  1 Rotation:                      OFF
  2 Source position:        P1:
  3 Destination position:   Q1:
```

b. Move the cursor to Source position (see Figure 10–101). You will see a screen similar to the following.

```
MIRROR IMAGE SHIFT
Shift amount/Teach
  Position data
  X :********  Y :********  Z :********
  1 Rotation:                      OFF
  2 Source position:        P1:
  3 Destination position:   Q1:
```

c. Move the robot to the source position (P1) and either record or specify the position:

- **To record a position,** jog the robot to the position you want, press and hold in the SHIFT key and press F5, RECORD.
- **To specify a previously recorded position or position register,** press F4, REFER. Type the number of a previously defined position or position register, and press ENTER.

```
Select reference position type
               P[ ]   PR[ ]
```

- **To clear a position setting** , press NEXT, >, and then press F1, CLEAR. At the prompt "Clear all data," press F4, YES, to confirm.

d. Move the cursor to Destination position (see Figure 10–101). You will see a screen similar to the following.

```
MIRROR IMAGE SHIFT
Shift amount/Teach
  Position data
  X :********  Y :********  Z :********
  1 Rotation:                      OFF
  2 Source position:        P1:    P[1]
  3 Destination position:   Q1:
```

- **To record a position,** jog the robot to the destination position (Q1). Press and hold in the SHIFT key and press F5, RECORD.
- **To specify a previously recorded position register,** press F4, REFER. Type the number of the position register and press ENTER.

```
Input position register number:
```

- **To clear a position setting** , press NEXT, >, and then press F1, CLEAR. At the prompt "Clear all data," press F4, YES, to confirm.

e. Record or specify the destination position.

12. If you want to rotate the positions,

a. Press F4, ON. You will see a screen similar to the following.

```
MIRROR IMAGE SHIFT
Shift amount/Teach
  Position data
  X :********  Y :********  Z :********
  1 Rotation:                      ON
  2 Source position         P1:
  3                         P2:
  4                         P3:
  5 Destination position    Q1:
  6                         Q2:
  7                         Q3:
```

b. Move the cursor to Source position (see Figure 10–101). You will see a screen similar to the following.

```
MIRROR IMAGE SHIFT
Shift amount/Teach
  Position data
  X :********  Y :********  Z :********
  1 Rotation:                        ON
  2 Source position          P1:
  3                          P2:
  4                          P3:
  5 Destination position     Q1:
  6                          Q2:
  7                          Q3:
```

c. Move the robot to the first source position (P1) and either record or specify the position:

- **To record a position,** jog the robot to the position you want, press and hold in the SHIFT key and press F5, RECORD.
- **To specify a previously recorded position or position register,** press F4, REFER. Select the position or position register.

```
Select reference position type
                P[ ]   PR[ ]
```

- **To clear a position setting** , press NEXT, >, and then press F1, CLEAR. At the prompt "Clear all data," press F4, YES, to confirm.

d. Record or specify all the source positions.

e. Move the cursor to Destination position (see Figure 10–101). You will see a screen similar to the following.

```
MIRROR IMAGE SHIFT
Shift amount/Teach
  Position data
  X :********  Y :********  Z :********
  1 Rotation:                      ON
  2 Source position         P1:    P[1]
  3                         P2:    P[2]
  4                         P3:    P[3]
  5 Destination position    Q1:
  6                         Q2:
  7                         Q3:
```

- **To record a position,** jog the robot to the destination position (Q1). Press and hold in the SHIFT key and press F5, RECORD.
- **To specify a previously recorded position register,** press F4, REFER. Type the number of the position register and press ENTER.

```
Input position register number:
```

- **To clear a position setting** , press NEXT, >, and then press F1, CLEAR. At the prompt "Clear all data," press F4, YES, to confirm.

f. Record or specify all the destination positions.

Note Pressing F2, EXECUTE, will cause the positions you have selected to be mirrored and **will not** cause robot motion.

13. **To mirror image the program,** press F2, EXECUTE.

```
Execute transform ok?
```

- **To execute the mirror image shift** , press F4, YES.
- **If you do not want to execute the mirror image shift** , press F5, NO.

14. Wait until software has finished processing the mirror image.

```
-PROCESSING-
```

Some positions in your program might not be able to be mirrored . When this happens, the mirror image software keeps the position in the program at the exact location and orientation it was in before the mirror image was executed. To correct this, you must reteach the position manually.

If this happens while your mirror image program is processing, you will see a message similar to the following.

```
Select P[1] : J6 angle (deg -234)
deg -234 deg 486  *Uninit*          QUIT
```

This message will be displayed for each position that cannot be mirrored. Note the position number; P[1] in this example.

To continue, press F5, QUIT. This will allow the mirror image program to continue processing. Continue noting the position number and axis for each position that did not mirror correctly.



Warning

Do not attempt to move the robot to a position that was not mirrored correctly; otherwise, you could injure personnel or damage equipment.

When the mirror image is complete, you must reteach each position that did not mirror correctly.

10.28 MONITORING PROGRAMS

10.28.1 Montioring Teach Pendant Programs

You can monitor a running teach pendant program from the SELECT menu. When you monitor a running program, the program is displayed and the cursor highlights the line currently being executed.

Use Procedure 10-48 to monitor a running teach pendant program.

Procedure 10-48 Monitoring a Running Teach Pendant Program

Conditions

- The teach pendant program or programs you want to monitor is (are) currently executing.

Steps

1. Press SELECT.

```
Select
                      50983 BYTES FREE
No. Program name                    Comment
 1  SUB1                          PR[            ]
 2  MAIN25                        JB[            ]
 3  PRG7                          MR[            ]
 4
 5
 6
```

2. Press F4, MONITOR. The program will be displayed on the screen.

If more than one program is running, you will see a screen similar to the following. The cursor will be on the line number of the instruction that is currently being executed. The cursor will move to each instruction as it is executed.

```
LONGDELAYPROGRA>  LINE 1    AUTO RUNNING\\\\\\\\\\\JOINT\100\%
Select\(monitor)\\\\\\\\\\\\\\\\
              TASK          STATUS
\1\EXEC1\\\\LONGDEL>\[LONGDEL>:Running\]
 2 EXEC2    DELAY1   [DELAY1  :Running ]
 3 EXEC3    DELAY2   [DELAY2  :Running ]









Select monitor program
 PROG   ROUT      PAUSE   ABORT
```

If the program or task name is longer than 8 characters, the full name will be displayed on the prompt line. The PROG and TASK keys designate which name is displayed on the prompt line. See the following screen for an example.

```
LONGDELAYPROGRA>  LINE 1    AUTO RUNNING\\\\\\\\\\\JOINT\100\%
Select\(monitor)\\\\\\\\\\\\\\\\
             TASK          STATUS
\1\EXEC1\\\\LONGDEL>\[LONGDEL>:Running\]
 2 EXEC2    DELAY1   [DELAY1  :Running ]
 3 EXEC3    DELAY2   [DELAY2  :Running ]









LONGDELAYPROGRAMNAME
 PROG   ROUT      PAUSE   ABORT
```

3. **To look at another area of the program while the program is being displayed,** press F2, LOOK. When you want the cursor to return to the line number of the instruction currently being executed, press F2, MONITOR.

10.29 MOTION GROUP DO OUTPUT FUNCTION (MULTIPLE MOTION GROUP OPTION)

10.29.1 Overview

This section contains information about the Motion DO feature. Motion groups can be robots, positioners, index devices, or nobots. Basic systems include a single motion group, the robot, so the busy indicator (when running a program) or the TP enable output (when jogging the robot) are sufficient for a PLC to monitor motion group use. When a system has more than one motion group, motion group usage can only be monitored by viewing the teach pendant. This information can not be known externally without using the teach pendant.

Motion DO is a status mechanism that uses digital or robot I/O to indicate:

- A motion group is selected /locked for use by a motion control program
- A motion group is being jogged via the Teach Pendant
- A motion group is not in use.

Two signals are provided for each group configured in the multi-group system. In this way, it is possible to recognize the currently "in use" motion groups without using the teach pendant, providing control information to the PLC, thus enhancing safety.

Note The motion group DO output function is available only when the multiple motion group option is used.

10.29.2 Restrictions

Be aware of the following restrictions when using the motion group DO output function:

- The same signal cannot be defined for different motion groups.
- When a program is running or paused, the type of signal (DO/RO) or signal number cannot be changed.

10.29.3 Operations

The motion group DO output function can assign two types of digital output signals to one motion group: jog signals and programmed signals. Any digital output signal the robot has or a robot output signal can be used as the Motion DO signal.

Each specified digital output signal can be turned on and off under the following conditions:

- Jog signals are turned on when the teach pendant is enabled for the selected jog group.
- Program signals are turned on when a motion group is locked by a paused or running motion program.

The same digital output can be assigned to the program signal and the jog signal of the same motion group. In this case, the two signals are logically OR-ed. This means that if either signal is ON, the OR-ed signal is also ON. The output signal is OFF only when both signals are OFF.

Jog Signals

All jog signals are turned OFF when the teach pendant is disabled. If the teach pendant is enabled, only the signals corresponding to the motion group selected by the teach pendant are turned ON; the other signals are turned OFF.

Programmed Signals

Regardless of whether the teach pendant is enabled, programmed signals might be turned on depending on whether the motion group specified by the program that is running or paused; the signals

will not be turned on merely by selecting a program. If a program running or paused in multi-tasking mode has a motion group, the programmed signals for that motion group are also turned on.

Note For coordinate motion jogging, only the leader group DO[ ] will be turned ON.

10.29.4 Setup

You can set up jog and programmed digital output signals from the Motion group DO screen.

The Motion group DO screen displays the items in Table 10–60 .

Use Procedure 10-49 to set up jog and programmed digital output signals.

Table 10–60. Motion group DO Screen Items

ITEM	DESCRIPTION
Program signal	This item is turned on when a motion group is locked by a paused or running motion program.
Jog signal	This item is turned on when the teach pendant is enabled for the selected jog group.

Procedure 10-49 Setting up the Motion Group DO Output Function

Steps

1. Press MENUS.
2. Select SETUP
3. Press F1, [TYPE].
4. Select Motion group DO. You will see a screen similar to the following.

```
Motion Group DO
                                        5/5
   GROUP No.  PROGRAM    JOG
      1        DO[  1]    DO[  6]
      2        DO[  2]    DO[  7]
      3        DO[  3]    DO[  8]
      4        DO[  4]    DO[  9]
      5        DO[  5]    DO[\10]
      6        DO[  6]    DO[ 11]
      7        DO[  7]    DO[ 12]
      8        DO[  8]    DO[ 13]
```

5. **To change between RO and DO,** move the cursor to the signal you want to change and press either F4, RO, or F5, DO.
6. Set the signals as desired.

Note If 0 is specified for a signal number, the corresponding signal is unspecified, and no output is produced.

10.29.5 Subprogram and Multi-tasking Execution

This section explains how the motion group DO output function works when a subprogram is called or when multi-tasking mode is used.

Example 1

If a program A with no motion group CALLs a subprogram B that has a motion group or groups, the signals for the subprogram B motion group(s) is (are) kept ON only when the subprogram is being executed.

Example 2

If a motion program A CALLs a subprogram B that has a motion group, if the subprogram has the same group mask as the CALLing program, the Motion DO outputs will not appear to change. In the case where the subprogram contains a different motion group than the CALLing program A, the output associated to the different motion group will turn ON only while the subprogram is running.

Example 3

If program A with no motion group RUNs program B with motion groups, the digital outputs associated to the motion groups in program B will be turned ON only while program B is executing.

Example 4

If motion program A RUNs program B with motion groups, the digital outputs associated to the motion groups in program A will be ON while program A executes, the DO's for motion groups in program B will be turned ON only while program B is executing. In this multi-tasking case, program B can only use groups that are not used in program A.

Example 5

Program "mainpro" does not have a motion group. It issues a RUN instruction to activate "progA" and "progB," which both have motion groups. "progA" operates motion group 1, and "progB," motion group 2.

Merely selecting program "mainpro" does not turn on programmed signals of any group.

When line 1 of "mainpro" is executed, "progA" is activated, resulting in the program signal of motion group 1 being turned ON.

When line 2 of "mainpro" is executed, "progB" is activated, resulting in the program signal of motion group 2 being turned ON.

When "progA" ends, the program signal of motion group 1 is turned OFF. When "progB" ends, the program signal of motion group 2 is turned OFF.

10.30 MOTION START DELAY DETECTION FUNCTION

10.30.1 Overview

When a robot must repeat motions within a short time (for example, because of short travel distance or high motion speed), the motions might be decelerated or oscillate even if the termination type of the corresponding motion instruction is "CNT 100." This is because the robot might not be ready for the next motion when one short motion ends, and cannot start immediately.

The motion start delay detection function detects and displays this kind of a delay in motion start. If deceleration or oscillation occurs, it is possible to determine the program line to be corrected according to the information detected by this function.

The motion start delay detection function is always enabled. The detected information is available at any time.

10.30.2 Restrictions

Keep in mind the following restrictions when you use the motion start delay detection function.

- If there is a non-motion instruction (such as a register, I/O, or wait instruction) between motion instructions, it might prevent an immediate start of motion even if the motion instructions use a termination type of CNT100, possibly causing a motion start delay to be detected.
- A CNT termination type setting is ignored when the tool shifts from joint motion to linear or circular motion, or vice versa. The robot always decelerates at such points. Therefore, a motion start delay is detected if the CNT termination type has been specified.
- No motion start delay is detected during single-step execution.

10.30.3 Detection Information and System Variables

The motion start delay detection function detects a motion start delay if the following condition is satisfied:

Condition: When a motion instruction that uses CNT 100 ends, the next motion cannot be started immediately.

If a delay is detected, the related information is recorded in a system variable.

All information related to a motion start delay is recorded in system variable **$PODATA_GRP[group]** . This system variable is provided for individual motion groups. The contents of the system variable are all cleared at cold start. The fields of this system variable are described in the following sections.

$PODATA_GRP[ ].$OVERRUN_CNT

This system variable indicates the number of times motion start delays are detected.

$PODATA_GRP[ ].$CUR_INDEX

This system variable indicates the index of the latest motion start delay data. The information about motion start delays are recorded in 50 memory locations (called buffers) in the order of indexes (sequence number).

Each buffer has a ring structure. When it becomes full of 50 (fixed) pieces of data, the latest data is overwritten to the oldest data. See Figure 10–113 .

This system variable holds the index of the information about the motion start delay that occurred most recently. First check this index, then the contents (program ID, program line number, and amount of delay, described later) of the buffer that corresponds to this index.

$PODATA_GRP[ ].$PROGRAM_ID[50]

This is a program ID memory buffer for a detected motion start delay. It indicates the internal ID of a program encountered with the motion start delay.

$PODATA_GRP[ ].$LINE_NO[50]

This is a program line number memory buffer for a detected motion start delay. It indicates the program line where the motion start delay was detected.

Keep in mind that the line number indicated here only correspond to the line where a motion start delay was detected. The line that caused the delay has a number lower than this line.

$PODATA_GRP[ ].$OVERRUN_ITP[50]

This is a motion start delay memory buffer. It indicates the amount of a detected delay. The larger the value, the larger the delay.

If you reference these buffers with the same index (in $PODATA_GRP[ ].$cur_index). you can identify the following information:

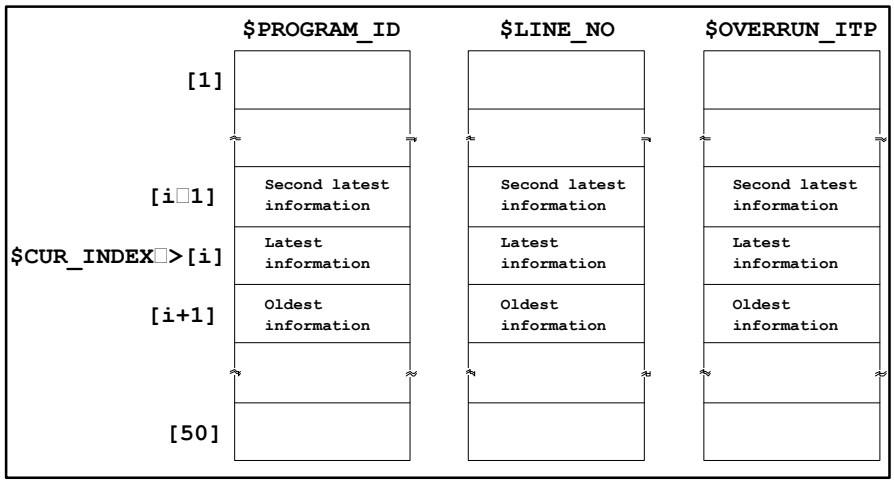
- Which program ($PROGRAM_ID[i])
- Which motion, or program line ($LINE_NO[i])
- How much delay ($OVERRUN_ITP[i])

where "i" is an index.

Each memory area consists of 50 buffer locations. It is always possible to check up to 50 pieces of the latest information about motion start delays.

The memory buffers are shown in Figure 10–113 .

Figure 10–113. Motion Start Delay Memory Buffers



10.30.4 Motion Start Delay Detection Function Example

The following sequence describes an example of how to use the motion start delay detection function.

1. Test run a program given to the robot to assure safety.
2. Reset the following system variables. The function can be used without being reset. However, if you reset it, the data will be more readable.

```
$PODATA_GRP[ ].$OVERRUN_CNT = 0
$PODATA_GRP[ ].$CUR_INDEX = 0
```

3. Run the program with the override for actual operation applied. Consider the following example program.

```
Example:  Program (internal ID:  1290)
1: J P[1] 100% Cnt 100
2: L P[2] 1000mm/sec Cnt 100
3: L P[3] 1000mm/sec Cnt 100
4: L P[4] 1000mm/sec Cnt 100
5: L P[5] 1000mm/sec Cnt 100
[END]
```

4. Check the results of the detection. Display the SYSTEM Variables screen, move the cursor to $PODATA_GRP[ ], and press ENTER to view the contents of the system variables.

   Assume the following conditions:

```
Example:  Conditions immediately after the above program
is
executed $OVERRUN_CNT        1
$CUR_INDEX           1
$PROGRAM_ID[1]       1290
$LINE_NO[1]          4
$OVERRUN_ITP[1]      1
```

   This example means that a motion start delay occurred when line 4 in a program with internal ID 1290 was executed.

5. Correct the program. According to the detected result, it is likely that motions before line 4 (P[4]) might be too close to each other. Check the positional data and correct the position or speed as required.

6. Repeat 1 through 5 until deceleration or oscillation does not occur any more.

10.31 OFFLINE SINGULARITY AVOIDANCE (Formerly Singularity Avoidance)

10.31.1 Overview

The offline singularity avoidance option allows you to modify a teach pendant program based on feedback supplied by the controller so that the program will be free of singularity problems.

Note This function works for six-axis robots only.

10.31.2 How it Works

After you execute the offline singularity avoidance function, the controller begins to collect internal motion data. By analyzing and processing this motion data, the system can detect potential singularity errors in your teach pendant program, and will provide a modified program free of singularity errors. The original taught points (both location and orientation) remain the same, but the modified program will use the WJNT motion option to change orientation configurations, and might add additional points as necessary.

The offline singularity avoidance function provides two methods for modifying your teach pendant programs.

- Method 1: If the modification to J5 falls within a specified threshold, the function will add WJNT to the linear motion line in the program where singularity was detected and will change the configuration.
- Method 2: If the modification to J5 exceeds a specified threshold, the function will partition the linear motion line into three sections; that is, it will add two additional points around the singular position, and will add a WJNT instruction with an alternative configuration (either flip or noflip) to the line in the program where singularity was detected.

Original Program Listing is a sample program in which singularity was detected. Modified Program Listing Using Method 1 is the same program after modification using method 1. Figure 10–114 is the same program after modification using method 2, and also shows a graphic example of method 2.

Original Program Listing

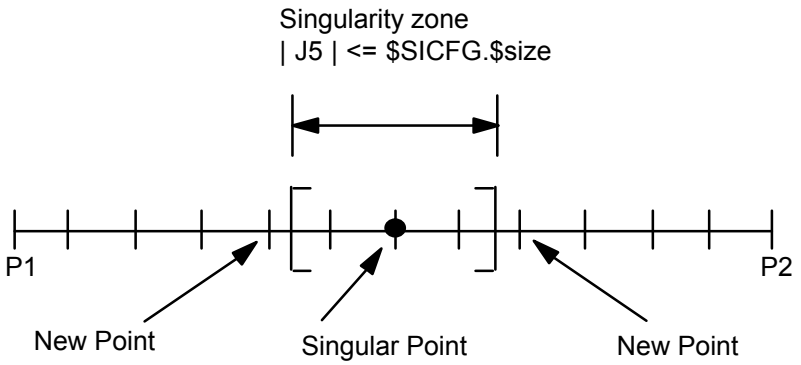
```
Original program:
      J P[1] 100% FINE
      L P[2] 1000 mm/s FINE <- singularity detected
```

Modified Program Listing Using Method 1

```
Modified program with Method 1:
      J P[1] 100% FINE
      L P[2] 1000 mm/s FINE WJNT <- modified with
                                    WJNT instruction
```

Figure 10–114. Modified Program Listing Using Method 2

```
Modified program with Method 2:
      J P[1] 100% FINE
      L P[3] 1000 mm/s CNT100 <- new point added
      L P[4] 1000 mm/s CNT100 WJNT <- new point and
                              change configuration
      L P[2] 1000 mm/s FINE <- change configuration
```



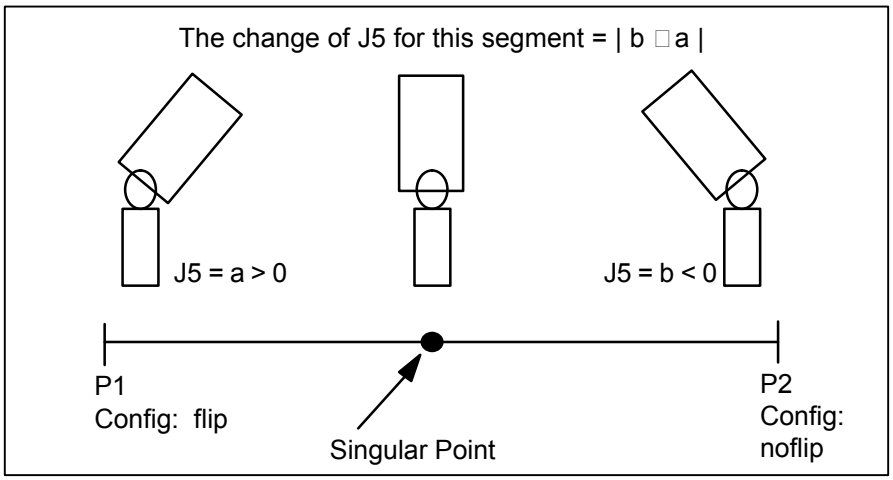
System Variables

You can choose which method the offline singularity avoidance function will use to modify your teach pendant programs by setting the system variable group, $SICFG, via the system variable screen on the teach pendant. Refer to Table 10–61 for information on the offline singularity avoidance system variables and on specifying which method to use.

Table 10–61. Offline Singularity Avoidance System Variables

VARIABLE	DESCRIPTION
$SICFG.$ssize Units: degrees Default: 3.5 degrees Range: 3.5 - 20 degrees	This item defines the threshold, or size, of the singularity zone (see Figure 10–114). A smaller value (smaller singularity zone) means that a singularity point is less likely to be detected. Conversely, a larger value (larger singularity zone) means that singularity is more likely to be detected.
$SICFG.$dj5_lim Units: degrees Default: 100 degrees Range: 3.5 - 200 degrees	This item is a threshold that determines whether method 2 will be used. If, after changing the configuration for a segment, the change of J5 between the start and destination positions exceeds this threshold, the system will use method 2 to modify the teach pendant program. The smaller this value, the more likely that method 2 will be used (see Figure 10–115).

Figure 10–115. J5 After Configuration Change



10.31.3 Compatibility

The offline singularity avoidance feature is compatible with the following items:

- Linear motion with Joint and Cartesian filters
- Extended axes
- TurboMove
- Rail tracking

Offline singularity avoidance is**not** compatible with the following items:

- AccuPath
- Line tracking
- Circular tracking
- Circular motion
- Multi-group motion with more than one group having full kinematics

Note The offline singularity avoidance feature will not work if the teach pendant program is unable to complete (for example, when limit errors occur), or if the controller does not have sufficient memory. The amount of memory necessary depends on the number of teach pendant programs currently loaded in the controller.

10.31.4 Limitations

The offline singularity avoidance function has the following limitations:

- This function is only supported on teach pendant programs that can be run with MACHINE LOCK on. Be aware that I/O is still **active** with MACHINE LOCK on.
- This function does not modify teach pendant programs that use position registers, or that contain JMP/LBL instructions.
- Because this function changes the configuration of some points in your teach pendant programs, position ID numbers might be different from the originals. Therefore, you should verify that the modified program works correctly.
- This function does not correct singularity that occurs at a destination point.
- This function requires memory from both SRAM and RAM disk, and will not work correctly if the amount is insufficient.
- This function can modify up to 24 teach pendant programs at one time.
- If you are using a rail tracking program, you must set up tracking parameters (boundaries, schedule and simulate encoder, and so forth) before using this function.

10.31.5 Procedure

Refer to Table 10–62 through Table 10–64 for information on the kinds of items you will set using the Singularity Avoidance screens.

Use Procedure 10-50 to create a singularity-free teach pendant program.

Table 10–62. Singularity Avoid Screen Items

ITEM	DESCRIPTION
Prompt line	This item is a prompt field for the Singularity Avoid screen.
Program name	This item is the name of the program to test for singularity.

Table 10–63. Singularity Avoidance Func Screen Items

ITEM	DESCRIPTION
Enable Singularity Avoidance	This item allows you to enable or disable Auto Singularity Avoidance.

Table 10–64. Singularity Avoid DETAIL Screen Items

ITEM	DESCRIPTION
No.	This item is a line identifier.
Original Program	This item is a listing of the programs in which singularity was detected.
New program	This item is a listing of the corresponding new program names. These programs are copies of the original programs in which singularity was detected, but with new names.

Procedure 10-50 Creating a Singularity-Free Teach Pendant Program

Warning

When you modify a program with the offline singularity avoidance function, you are changing the configuration of taught points. This affects the motion behavior of the robot. When you execute the modified program, make sure the program is running at safety speed to verify that it is working correctly; otherwise, you could injure personnel or damage equipment.

Conditions

- You have created a teach pendant program.
- You have determined whether your program contains I/O instructions. I/O will still be executed with MACHINE LOCK on.
- You have disabled the teach pendant and have verified that the controller is not in error status.

Steps

1. Press MENUS.
2. Select Utilities.
3. Press F1, [TYPE].
4. Select Singularity. You will see a screen similar to the following.

```
Singularity Avoid
  Please input TP program name
  for Singularity Avoidance
  Program name  [               ]
```

5. Press F4, [CHOICE], and select the appropriate teach pendant program.
6. Press F2, EXECUTE. You will see a screen similar to the following.

```
Singularity Avoid
  Please make sure that robot is in
  proper start position!
  WARNING !
  Program will run with MACHINE LOCK !
  I/O may still be active !

  Start ?
```

Note Make sure that the robot is at the proper starting position. If not, you should jog the robot to the proper starting position. This position should be the same as the original teach pendant program.

7. Decide if you want to run the offline singularity avoidance function.

   - **If you do NOT want to run this function** , and you want to exit this procedure, press F3, NO. You will return to the initial Singularity Avoidance screen.
   - **If you do want to run this function** , press F4, YES. You will see a screen similar to the following.

```
Singularity Avoid
  Please make sure that robot is in
  proper start position!
  WARNING !
  Program will run with MACHINE LOCK !
  I/O will still be active !

  Please wait!  Processing...
```

Note The teach pendant will be locked while the singularity avoidance function is processing. If you would like to terminate this process, press the HOLD key.

If singularity is NOT detected, you will see a message similar to the following:

No singularity is detected.

If singularity is detected, you will be reminded that you should back up your original teach pendant program. You will see a screen similar to the following.

```
Singularity Avoid
  Singularity is detected!
  Make sure to backup your original
  TP program(s)!
  Recommend:
     Don't use original program name
     for modification because it may
     be called by other programs!
```

8. If you would like to see the line in the program in which singularity was detected, perform the following steps:

a. Press MENUS.

b. Select Alarm.

c. Select Hist.

d. When you are finished, press MENUS and select UTILITIES. You can now continue to Step 9 .

9. Press F4, CONTINUE. You will see a screen similar to the following.

```
Singularity Avoid
 No.      Original Program      New program
  1       TEST1                 [TEST1  ]
  2                             [       ]
  3                             [       ]
  4                             [       ]
  5                             [       ]
  6                             [       ]
  7                             [       ]
  8                             [       ]
  9                             [       ]
```

The Original Program column in the center of the screen lists the names of the programs in which singularity was detected. The New program column on the right side of the screen allows you to change the names of the programs before they are modified by the singularity avoidance function.

Note By default, the program names in the New program column have the same names as the programs in the Original Program column.

10. Decide if you want to modify the program in which singularity was detected.

- **If you do NOT want to modify the program**, and you want to exit this procedure, press F5, NO. You will return to the initial singularity avoidance screen.
- **If you do want to modify the program**, complete the following steps.

a Move your cursor to the name of the program in the New program column.

b Type a new name for the program that you want to modify.

Note Make sure you do not use the same name as an existing program; otherwise, the original program will be overwritten.

c After you have typed the name for the new modified program, press F4, YES. You will see a screen similar to the following.

```
Singularity Avoid
  Make sure to backup your original
  TP program(s) !
  Modify program(s)?
```

- **If you do NOT want to modify the program**, press F3, NO. You will return to the previous screen.
- **If you do want to modify the program**, press F4, YES. You will see a screen similar to the following.

```
Singularity Avoidance
     Singularity is detected and
     TP program is being modified!
  Please wait !  Process...
```

When the modification is complete, you will see a message similar to the following:

WARNING configuration is changed!

> **Warning**
>
> **The configuration of taught points has been changed. This affects the motion behavior of the robot. When you execute the modified program, make sure the program is running at safety speed to verify that it is working correctly; otherwise, you could injure personnel or damage equipment.**

The offline singularity avoidance program modification procedure is now complete.

10.31.6 Troubleshooting

Table 10–65 lists and describes error messages that you might receive when using the offline singularity avoidance function. These error messages will be displayed along the bottom of the teach pendant screen.

Table 10–65. Error Messages, Descriptions and Remedies

ERROR MESSAGE	DESCRIPTION AND REMEDY
Please disable Teach Pendant!	This message indicates that in order to run the offline singularity avoidance function, the teach pendant should be disabled. To start over, press the RESET key, then press the PREV key.
Error! TP-Prog paused	This message indicates that in order to use the offline singularity avoidance function, the original teach pendant program must be run through without any problems such as limit errors, position not reachable errors, illegal speed errors, and so forth. Make sure that there are no unresolved errors in order to use this function. To start over, press the RESET key, then press the PREV key.
Error! TP has no motion line!	This message indicates that the specified teach pendant program has no motion line (that is, it is either empty or no motion line exists). Make sure that the specified teach pendant program contains a motion instruction. To start over, press the RESET key, then press the PREV key.
WARNING: New prog exists!	This message indicates that the new program name you have typed is the same as that of an existing program. This is a warning message, and if you ignore it, the existing program will be overwritten with the modifications. Either rename the new program or make sure that you want to overwrite the original program.
Error! Program name conflicts	This message indicates that the new program name you have typed is the same as that of one of the original programs queued to be modified. The system will automatically change the name back to the default if this error occurs. Type in another name for the program.

Table 10–65. Error Messages, Descriptions and Remedies (Cont'd)

ERROR MESSAGE	DESCRIPTION AND REMEDY
Error! Can't support multi_group!	This message indicates that the system is trying to use method 2 to modify the teach pendant program with multi-group motion. Because multi-group programs need to be modified using method 1, increase the value of $SICFG.$dj5_lim. To start over, press the PREV key.
Error! Can't copy TP!	This message indicates that the system cannot copy the teach pendant program that you have specified, and therefore, cannot complete the function. Use the SELECT key to select the original teach pendant program and return to the user interface for the singularity avoidance function to attempt this process again. To start over, press the PREV key.
Not support position register in TP!	This message indicates that the offline singularity avoidance function cannot modify a program with a position register. Use a normal position ID to represent a position in your teach pendant program. To start over, press the PREV key.
Too many programs involved!	This message indicates that too many teach pendant programs need to be modified. It is possible that several sub-programs are being called and need to be modified in order to avoid the detected singularity. Use fewer than 24 programs in order to use the singularity avoidance function. To start over, press the PREV key.
Sub-program called twice!	This message indicates that a sub-program was called more than once. The singularity avoidance function will not modify this sub-program. To start over, press the PREV key.
Error! Can't read dat-file	This message indicates that a data file in RAM Disk (RD) cannot be opened. You must make sure that there is enough memory in RD; if not, cycle power to clear RD memory and try again. If the problem persists, either the specified teach pendant program has too many motion lines, or the system does not have enough memory.

10.32 OPERATION LOG BOOK [OPTION]

10.32.1 Overview

Operation Log Book automatically records teach pendant operations and alarms in a buffer that can be displayed in the Log Book menu of the teach pendant or saved as a text file. Some operations can include a screen image to help in analyzing the event.

A system can have multiple Log Books and you can specify which events are recorded in each Log Book. In this way, frequent and infrequent events can be recorded to separate Log Books.

When a Log Book buffer is full and a new record is added, the oldest record is deleted. The number of events that can be stored in a Log Book buffer varies based on the type of events saved because not all events are the same size. You can change the size of the buffer by setting a system variable.

Note This function is independent of the password log function.

See the following example of a Log Book.

```
Log book                      JOINT  10 %
Operation        (20.0k)           1/790
  1*SHIFT,F5(TOUCHUP) is pressed, line 2
  2 '+Y^(J6)' is pressed
  3 SHIFT+FWD is pressed
  4*Select 'YES' in 'You are in the diff
  5 SHIFT+FWD is pressed
  6 SHIFT+FWD is pressed
  7 Single step ON
  8 'STEP' is pressed
  9 'LINER' is selected
 10*SHIFT,F1(POINT) is pressed line 2/2
 11 '-Y^(J6)' is pressed
 12*'50' is entered
 13*'Cnt' is selected in 'Motion modify'
 14*SHIFT,F1(POINT) is pressed, line 1/1
 15 '-Y^(J2)' is pressed
 16 '-Y^(J3)' is pressed
 17 '+Y^(J5)' is pressed
 18 '-Y^(J5)' is pressed
 19 Override 40%
 20 '+%' is pressed
 21 Override 35%
 22 '+%' is pressed
 23 Override 30%
 24 '+%' is pressed
 25 Override 25%
 26 '+%' is pressed
 27 Override 20%
 28 '+%' is pressed
 29 Override 15%
 30 '+%' is pressed
 31 '-Y^(J5)' is pressed
 32 'RESET' is pressed
 33 Menu changed 'PNS0001'
 34*'PNS0001' is entered
 35*F2(CREATE) is pressed, line 1/16
 36 Menu changed 'SELECT'
 37 'SELECT' is pressed
```

The lines marked with an "*" have an associated screen image, as seen in the example of line 12 below. The screen image shows the previous value of the changed item.

```
PNS0001
                                          1/1
  1 J P[1] 100% CNT100
[END]
 POINT
```

General Limitations

Note the following general limitations for Log Book:

- Log Book does not support certain *i*Pendant operations such as selecting a link on a web page.
- Operations are not recorded in controlled start just after initial start. Cold start must be performed to begin logging events.
- Operations in the configuration menu are not recorded
- If you change the size of a log book, all data in the log book are lost.
- Some events are cyclically monitored for logging. Some of the following operations may not be recorded if they occur faster than the monitoring cycle:
  - Override ("Override x%")
  - Coordination (JOINT coordinate, User Coordinate etc.)
  - Single step (Single step ON/OFF)
  - Motion group ("Motion group x")
  - Sub group ("Sub group ROBOT/EXT")
  - User frame number ("User frame x")
  - Tool frame number ("Tool frame x")
- If you change the dictionary, some entries in the LOG BOOK screen cannot be read. To read such log, you must use the language that was used when the log was recorded.
- If passwords are enabled and $LOGBOOK.$LOG_ENT is a valid book number when you log in, your password input is recorded as "'x' is entered". Passwords for other functions are also recorded. To prevent passwords from being recorded, you can use screen filtering to filter out the screens that contain passwords.
- "'x' is selected in 'y' window" is not recorded for SpotTool+ warning windows.
- If you press a function key that has no label, the key number followed by empty parentheses will be recorded.

  Example If the function key line is as follows:

```
[TYPE]                          ON     OFF
```

F3 has no label. If you press F3 and $LOGBOOK $LOG_FNKEY is a valid value, “F3( )” is pressed” is recorded.

- If you change the remote TCP number in remote TCP jog mode, this is recorded as a change of user frame.
- If you changed current JOG coordination by parameter instruction, the change is recorded when the group is selected.
- Screen image doesn’t support double height font, which is used in the “on the fly” screen of ArcTool. It is recorded as two lines which have the same characters.

Example

0.0 Amps

in “On the fly” screen is recorded as

```
   0.0 Amps
0.0 Amps
```

- For KAREL read instructions, screen image is recorded just after input.
- KCL is not supported
- If the TP and CRT are used at the same time, log book records the operations of both of them. It might be difficult to distinguish between TP and CRT operations in a log.
- Screen changes caused by the automatic backup function are not recorded.

Refer to Filtering Alarms and Screens .

10.32.2 Recorded Events

The following is the list of all events that can be recorded by the Log Book function.

Recording of individual events can be disabled by setting the system variable specified in the ’Parameter’ column. If it is 0, the event is not recorded. Some of them are disabled by default. If the system variable is not 0, the value indicates the Log Book No. to which the event is recorded.

Some events have a screen image. This information helps to analyze the record. However, screen images take a lot of space in the Log Book; you can disable the recording of screen images by setting the system variable specified in the ’Save screen’ column.

Refer to Filtering Alarms and Screens later in this section.

Table 10–66. Events Recorded by Log Book

Message	Event	Parameter (Default)	Save Screen (Default)
Alarm history	When an alarm occurs, the alarm message is recorded. This record is the same as the record of the alarm history menu. To choose the alarms to be recorded, the 'filtering' function is provided.	$LOGBOOK. $LOG_ER (1)	None
'x' is entered	When a value or a word is entered, this is recorded. The 'x' in the message is the entered value or word. If screen image is enabled for this event, it will contain the previous value. Values will be recorded even if they are invalid. If you cancel an input, it is not recorded.	$LOGBOOK. $LOG_ENT (1)	$LOGBOOK. $IMG_ENT (TRUE)
'x' is selected 'x' is selected in 'y' menu	When a menu item is selected, this is recorded. The 'x' in the message is the selected item. If the menu has a title, the 'y' in the message shows the title. If screen image is enabled for this input, it will show the screen just before opening the menu.	$LOGBOOK. $LOG_SEL (1)	$LOGBOOK. $IMG_SEL (TRUE)
'x' is selected in 'y' window	When an item is selected in warning window, this is recorded. The 'x' in the message is the selected item. The 'y' in the message is the beginning of the message in the warning window. If screen image is enabled for this event it will show the warning window.	$LOGBOOK. $LOG_WIN (1)	$LOGBOOK. $IMG_WIN (TRUE)
'x' is selected in MENU 'x' is selected in FCTN	When an item is selected by MENU key or FCTN key, this is recorded. The 'x' in the message is the selected item	$LOGBOOK. $LOG_MENU (1)	None

Table 10–66. Events Recorded by Log Book (Cont'd)

Message	Event	Parameter (Default)	Save Screen (Default)
JOG menu TOOL 'x' JOG menu USER 'x' JOG menu JOG 'x' JOG menu GROUP 'x' JOG menu ROBOT JOG menu EXT	Operations of the JOG menu are recorded with these messages.	$LOGBOOK. $LOG_JGMN (1)	None
Menu changed 'x'	When a menu is changed, this is recorded. The 'x' in the message is the title of the new menu. Changes to sub menus such as Config or Detail are not recorded.	$LOGBOOK. $LOG_MNCHG (1)	None
'x' is pressed	When a key is pressed, this message is recorded with the key name. It is not recorded when the key is released. If SHIFT is held when the key is pressed, the word 'SHIFT,' is added to the key name(except SELECT, EDIT, DATA). The keys are grouped as follows, and you can disable recording of each group.		

Table 10–66. Events Recorded by Log Book (Cont'd)

Message	Event	Parameter (Default)	Save Screen (Default)
	F1, F2, F3, F4, F5	$LOGBOOK. $LOG_FNKEY (1)	$LOGBOOK. $IMG_FNKEY (TRUE)
	+X(J1), -X(J1), +Y(J2), -Y(J2), +Z(J3), -Z(J3), +X^(J4), -X^(J4), +Y^(J5), -Y^(J5), +Z^(J6), -Z^(J6)	$LOGBOOK. $LOG_JGKY (1)	None
	SELECT, EDIT, DATA,	$LOGBOOK. $LOG_PRGKEY (1)	None
	UF1, UF2, UF3, UF4, UF5, UF6, UF7	$LOGBOOK. $LOG_UFKY (1)	None
	+%, -%, COORD	$LOGBOOK. $LOG_OVRKY (1)	None
	FWD, BWD	$LOGBOOK. $LOG_FWDKY (1)	None
	HOLD	$LOGBOOK. $LOG_HLDKY (1)	None
	STEP	$LOGBOOK. $LOG_STPKY (1)	None
	PREV	$LOGBOOK. $LOG_PRVKY (1)	None
	ENTER	$LOGBOOK. $LOG_ENTKY (1)	None
	ITEM	$LOGBOOK. $LOG_ITMKY (1)	None
	RESET	$LOGBOOK. $LOG_RSTKY (1)	None
Override x%	When override is changed, this is recorded. The 'x' in the message is the new override value. This records all change of override by any method. For example, '+%' key, TP is enabled, override instruction of program.	$LOGBOOK. $LOG_OVR (1)	None

Table 10–66. Events Recorded by Log Book (Cont'd)

Message	Event	Parameter (Default)	Save Screen (Default)
JOINT coordinate USER coordinate TOOL coordinate JOG coordinate PATH coordinate	When coordinate is changed, this is recorded. This records all change of coordinate by any method.	$LOGBOOK. $LOG_CRD (1)	None
Single step ON Single step OFF	When single step is changed, this is recorded. This records all change of single step by any method.	$LOGBOOK. $LOG_STEP (1)	None
Motion group x	When motion group is changed, this is recorded. The 'x' in the message is the new motion group This records all change of motion group by any method.	$LOGBOOK. $LOG_GRP (1)	None
Sub group ROBOT Sub group EXT	When sub group is changed, this is recorded. This records all change of sub group by any method.	$LOGBOOK. $LOG_SGRP (1)	None
User frame x	When user frame number is changed, this is recorded. The 'x' in the message is the new user frame number This records all change of user frame number by any method.	$LOGBOOK. $LOG_UF (1)	None
Tool frame x	When tool frame number is changed, this is recorded. The 'x' in the message is the new tool frame number This records all change of tool frame number by any method.	$LOGBOOK. $LOG_UT (1)	None
Save file x Load file x	File x is saved File x is loaded These events are ONLY for file save/load in following screens • File screen • Program list screen	$LOGBOOK. $LOG_FILE (1)	None
WAIT is released (x, y)	The waiting WAIT instruction is canceled by WAIT RELEASE in program x line y.	$LOGBOOK. $LOG_WTRLS (0)	None

Table 10–66. Events Recorded by Log Book (Cont'd)

Message	Event	Parameter (Default)	Save Screen (Default)
Create program x Delete program x Write line x of y Delete line x of y	TP program x is created. TP program x is deleted. Line x of TP program y is written. Line x of TP program y is deleted.	$LOGBOOK. $LOG_PGCHG (0)	None
Write P[x] of y	Position data P[x] of TP program y is written. These are also recorded when program is changed internally.	$LOGBOOK. $LOG_SETPOS(0)	None
TP 'x' ON TP 'x' OFF	This records the low level key operation. All key operations of Teach Pendant are recorded. When a key is pressed, 'TP x ON' is recorded. When a key is released, 'TP x OFF' is recorded. ENABLE switch and E-STOP are also recorded. SHIFT key is treated as normal key. CRT operations are not recorded by this event. 'x' in the message is key name, the following keys are recorded. ENABLE, ESTOP PREV, F1, F2, F3, F4, F5, NEXT SHIFT, MENU, SELECT, EDIT, DATA,FCTN UP, DOWN, LEFT, RIGHT, DISP RESET, STEP, BACKSPACE, ITEM, HOLD, FWD, BWD, COORD, +%, -% 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, -/,, ., ENTER, HELP/DIAG, UF1, UF2, UF3, UF4, UF5, UF6, UF7, +X(J1), -X(J1), +Y(J2), -Y(J2), +Z(J3), -Z(J3), +X^(J4), -X^(J4), +Y^(J5), -Y^(J5), +Z^(J6), -Z^(J6),	$LOGBOOK. $LOG_TPKY (0)	None
Cold start Power failure recovery Control start	Start mode is recorded.	$LOGBOOK. $LOG_STMD(1)	None

10.32.3 Operations

Displaying the Log Book Screen

1. Press MENUS.
2. Select 4, ALARM.
3. Press F1, TYPE.
4. Select Log Book. You will see a screen similar to the following:

```
Log book
Operation        (20.0k)          1/790
  1*SHIFT,F5(TOUCHUP) is pressed, line 2
  2 '+Y^(J6)' is pressed
  3 SHIFT+FWD is pressed
  4*Select 'YES' in 'You are in the diff
  5 SHIFT+FWD is pressed
  6 SHIFT+FWD is pressed
  7 Single step ON
  8 'STEP' is pressed
  9 'LINER' is selected
 10*SHIFT,F1(POINT) is pressed line 2/2
```

5. **To change to a different Log Book** , press F2, BOOK.
6. **To view the screen image** for a line that is marked with a "*", select the line and press F3, DETAIL. You will see a screen similar to the following:

```
Log book                      JOINT  10 %
                                   1/790
*SHIFT,F5(TOUCHUP) is pressed, line 2/3
    00/06/02 14:17:36
--Screen image -------------------------
PNS0001
                                     2/3
   1 J P[1] 100% CNT50
   2 J P[2] 100% FINE
 [END]
```

7. **To view details** of an entry, select the entry and press F3, DETAIL. You will see a screen similar to the following:

```
Log book
                                    5/790
SRVO-001 Operator panel E-stop
    00/06/02 14:17:36
```

8. **To clear the log** , press F5, CLEAR, then press F4, YES.

Saving Log Book data as a Text File

Log Book data is saved as text file LOGBOOK.LS. There are two ways to save data:

- **At the FILE menu**, press F4, BACKUP, and select “Error Log”.

  LOGBOOK.LS is saved along with the error log files.
- **In the Log Book menu**, press the FCTN key and select SAVE.

  LOGBOOK.LS is saved to the selected device.

See the following example of a LOGBOOK.LS file.

```
Operation
============================================================
*SHIFT,F5(TOUCHUP) is pressed, line 2/3
00/06/02 14:17:36
--Screen image------------------------------
PNS0001\\\\\\\\\\\\\\\\\\\\\\\\JOINT\\10\%
   1 J P[1] 100% CNT59
   2 J P[2] 100% FINE
 [END]
 POINT                          TOUCHUP>
============================================================
'+Y^(J6)' is pressed
00/06/02 14:17:20
============================================================
SHIFT+FWD is pressed
00/06/02 14:17:12
============================================================
Select 'YES' in 'The cursor is on a different' window
00/06/02 14:16:40
--Screen image------------------------------
PNS0001\\\\\\\\\\\\\\\\\\\\\\\\JOINT\\10\%
     +------------------------------+
     |The cursor is on a different  |
     |line from where the program   |
   1 |PAUSED [1].                   |
```

```
   2 |Are you sure you want to run  |
 [END|from this line ?               |
     |                               |
     |   \\\\YES\\\\         NO      |
     |                               |
     +-------------------------------+
 POINT                          TOUCHUP>
============================================================
SHIFT+FWD is pressed
00/06/02 14:16:30
============================================================
SHIFT+FWD is pressed
00/06/02 14:16:28
============================================================
```

Filtering Alarms and Screens

You can set system variables to filter specific alarms and screens.

By default, Log Book filters out warning errors. You can filter errors to be logged by severity, type, and item number. Refer to Table 10–67

Table 10–67. System Variables used for Filtering

System Variable Name	Type	Default	Range	Description
$LOG_ER_SEV	INTEGER	6	-2147483648 2147483647	Filter by severity of error. When a bit in this is True, the corresponding errors are logged. • Bit 0 (1): Log warning errors. • Bit 1 (2): Log pause errors. • Bit 2 (4): Log abort errors. Ex. 6 = pause and abort errors are logged. Priority of this is lower than $LOG_ER_TYP and $LOG_ER_ITM.

Table 10–67. System Variables used for Filtering (Cont'd)

System Variable Name	Type	Default	Range	Description
$LOG_ER_TYP[1-n]	INTEGER	0	-2147483648 2147483647	Filter by type of error. If a positive value is specified, alarms of the specified type are logged. Ex. 11 = SRVO alarms are logged. If a negative value is specified, alarms of the specified type are not logged. Ex. -11 = SRVO alarms are not logged. Priority of this is higher than $LOG_ER_SEV but lower than $LOG_ER_ITM.
$LOG_ER_ITM[1-n]	INTEGER	0	-2147483648 2147483647	Filter by individual error. If a positive value is specified, the specified alarm is logged. Ex. 11001 = SRVO-001 is logged. If a negative value is specified, the specified alarm is not logged. Ex. -11001 = SRVO-001 is not logged. Priority of this is higher than $LOG_ER_SEV and $LOG_ER_TYP.

Control error filtering by setting the following system variables.

- $LOG_ER_ITM[1-n],
- $LOG_ER_SEV
- $LOG_ER_TYP[1-n]

The priority of the settings is: $LOG_ER_SEV < $LOG_ER_TYP < $LOG_ER_ITM

See the following for an example of error filtering:

```
SRVO-001 "Operator panel E-stop"   Severity=PAUSE, Type=11 Item=11001
SRVO-038 "Pulse mismatch"          Severity=ABORT Type=11 Item=11038
TPIF-104 "Teach pendant is disabled"Severity=WARN Type=9 Item=9104
```

Table 10–68. Alarm Filtering Example

$LOG_ER_SEV	$LOG_ER_TYP	$LOG_ER_ITM	SRVO-001	SRVO-038	TPIF-104
6	0, 0, ..	0 ,0, ..	Logged	Logged	Not logged
6	0, 0, ..	9104, 0, ..	Logged	Logged	Logged
6	0, 0, ..	9104, -11001, ..	Not logged	Logged	Logged
7	-11, 0, ..	11001, 0, ..	Logged	Not logged	Logged
0	11, 9, ..	-11001	Not logged	Logged	Logged

Screen filtering enables Logbook to log UIF events that occur in the screens you specify. Screen filtering supports the following events (UIF events).

- 'x' is entered
- 'x' is pressed
- 'x' is selected
- 'x' is selected in 'y' menu
- 'x' is selected in 'y' window
- 'x' is selected in MENU
- 'x' is selected in FCTN
- JOG menu TOOL 1 etc

Note TP'x' ON/OFF is not supported.

By default, screen filtering is disabled. Screen filtering is enabled/disabled by $LOGBOOK.$SCRN_FL.

- If $LOGBOOK.$SCRN_NO_ENT is TRUE, events in registered screens are recorded. If FALSE, events in NOT registered screens are recorded.
- To register screens, softpart ID and screen ID must be set to $LOG_SCRN_FL[ ].$SP_ID and $LOG_SCRN[ ].$SCRN_ID. By default, no screen is registered for filter.

Table 10–69. System Variables for Screen Filtering

System Variable Name	Type	Default	Range	Description
$LOGBOOK.$SCRN_FL	BOOLEAN	FALSE	FALSE/TRUE	Screen filter is disabled/enabled
$LOGBOOK.$SCRN_NO_ENT	BOOLEAN	TRUE	FALSE/TRUE	TRUE: Events on registered screens are not recorded. Events on other screens are recorded. FALSE: Events on registered screens are recorded Events on other screens are not recorded.
$LOGBOOK.$NUM_SCRN_FL	Integer	20	1,200	Number of $LOG_SCRN_FL
$LOG_SCRN_FL[ ].$SCRN_ID	ULONG	0	0,4294967295	Screen ID of screen for filter
$LOG_SCRN_FL[ ].$SP_ID	ULONG	0	0,4294967295	Softpart ID of screen for filter

To filter screens so that passwords are not recorded, set the following:

```
$LOGBOOK.$SCRN_FL = TRUE
$LOG_SCRN_FL[1].$SP_ID = 935
 $LOG_SCRN_FL[1].$SCRN_ID = 1
```

If you have an *i*Pendant, you can find the softpart ID and screen ID for the menus currently loaded on your system. Go to the DISP MENU, select HELP/DIAGNOSTICS, then select Menu Help from the flyout menu.

10.33 ORIGINAL PATH RESUME

Original Path Resume is a feature that, after Hold or Emergency Stop, allows the robot to return to its stop position before resuming the program, even if the robot has moved since the hold occurred. The stop position is the last program commanded position when the Hold or Emergency Stop occurred. The move from the current position to the stop position is called the “Restart Move”.

For example, if a Stop severity alarm occurs, the robot will decelerate to a stop and pause the TP program. The user might then jog the robot to another position to correct the cause of the alarm. When the TP program is resumed with Original Path Resume, the robot will first execute a Restart move to the Stop position, and then resume the program.

Figure 10–116. Hold Without Original Path Resume — Path Resumes from Current Position

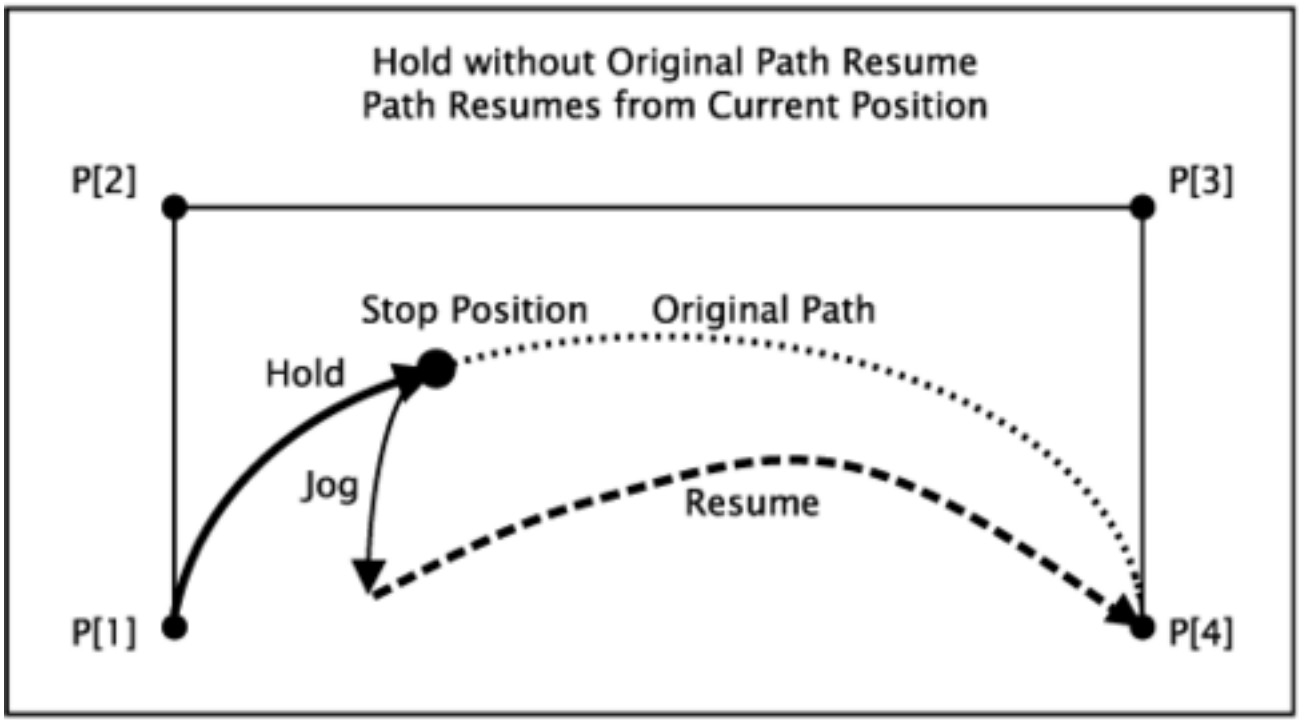
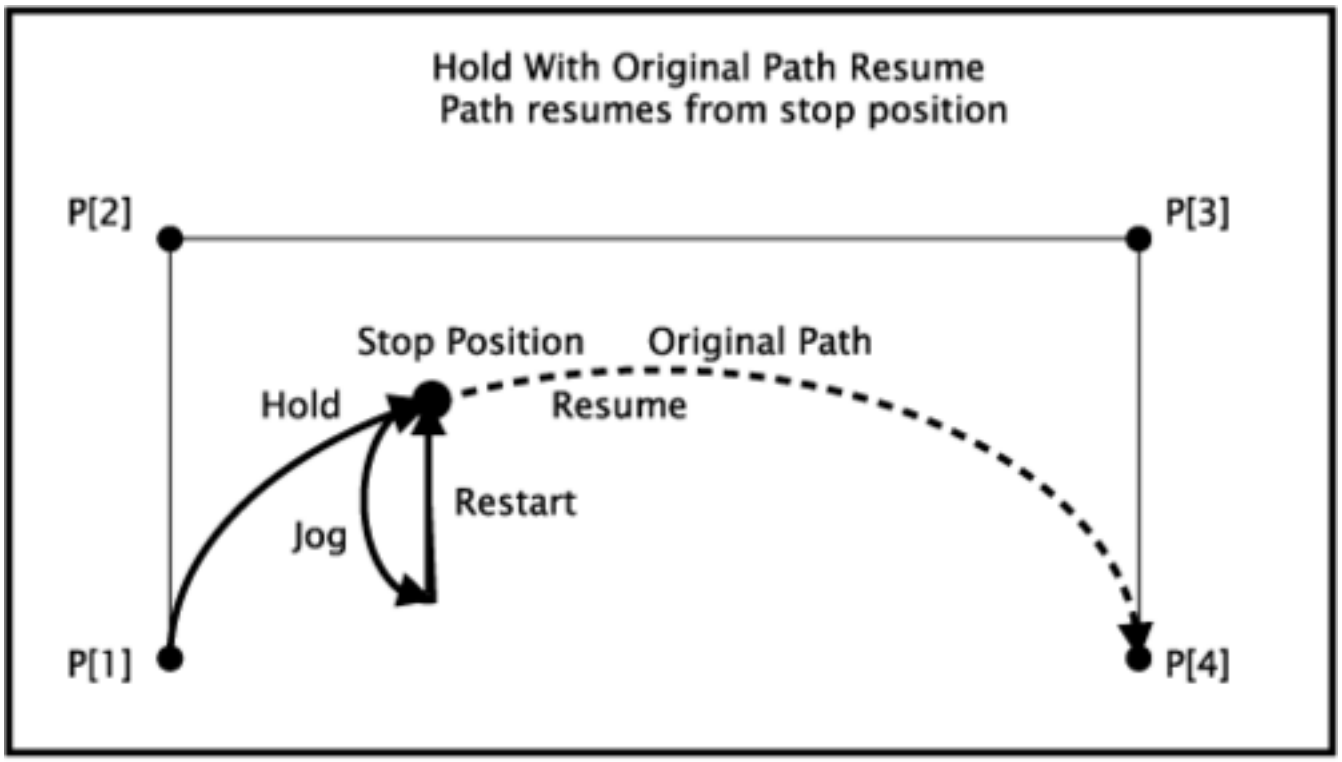
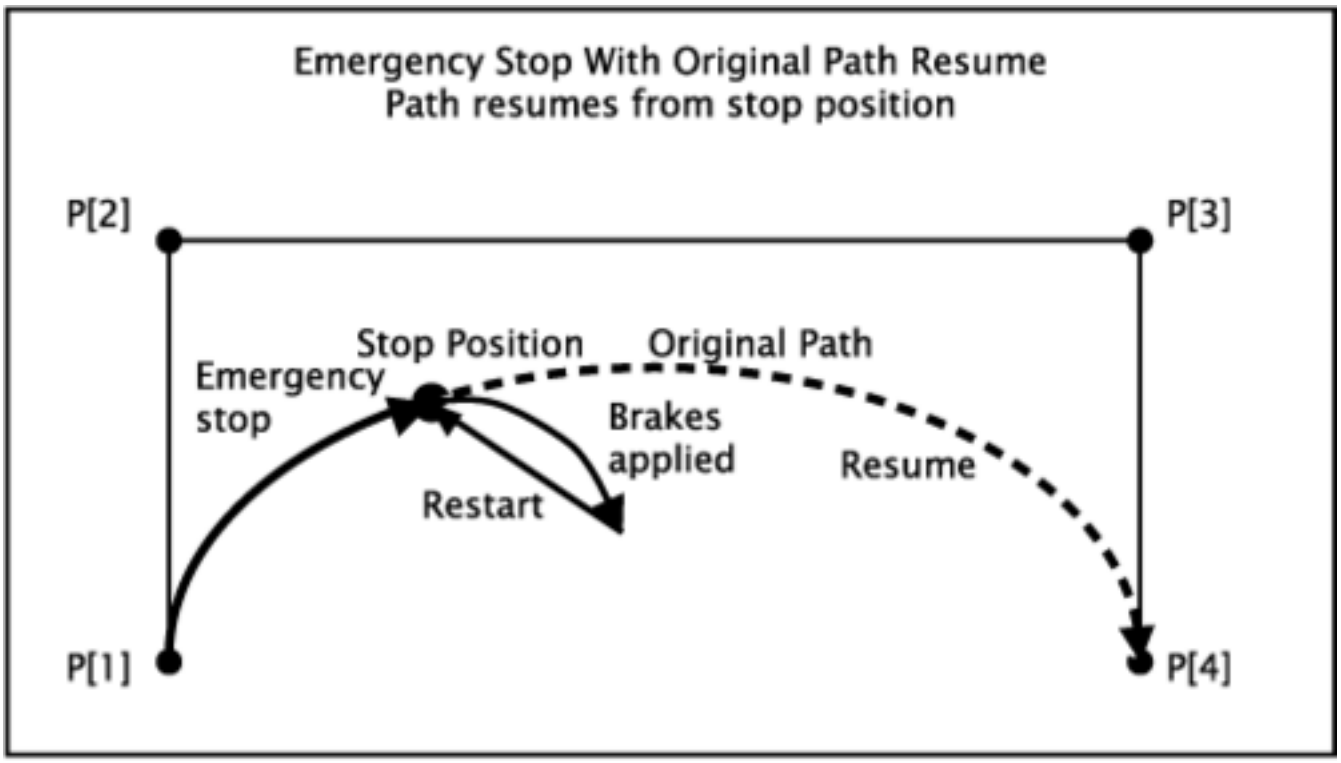


Figure 10–117. Hold With Original Path Resume — Path Resumes from Stop Position



When an Emergency Stop occurs, servo power is removed and the brakes are applied. Depending on the payload mass, speed, and direction of gravity, the robot will typically depart from the normal path. When the TP program is resumed with Original Path Resume, the robot will first execute a Restart move to the Stop position, and then resume the program.

Figure 10–118. Emergency Stop With Original Path Resume — Path Resumes from Stop Position



System Variables

Original Path Resume is enabled using $SCR.$ORG_PTH_RSM:
$SCR.$ORG_PTH_RSM = TRUE (Original Path Resume Enabled)
$SCR.$ORG_PTH_RSM = FALSE (Original Path Resume Disabled)

Motion type of the restart move is set by $MCR_GRP[].$RSM_MOTYPE

1. Joint
2. Linear
3. Program (Joint when hold segment Joint, Linear when hold segment Linear, Circular, or Arc)

Speed of the restart move is set by $MCR_GRP[].$RSM_SPEED, in mm/s or % as appropriate to motion type.

Orient type of the restart move is set by $MCR_GRP[].$RSM_ORIENT

1. ORNT
2. WJNT

Limitations of Original Path Resume

- Without the Constant Path Option, the resumed path may not exactly match the position and orientation of the original path. The stop position orientation may be close to, but not exactly, on the original path.
- With the Constant Path Option, the resumed path will match the position and orientation of the original path.
- The Restart move is subject to the same limitations as any other move. It is possible for the robot to get to a position in which the Restart move may cause alarms such as “Axis Limit” or “In Singularity”, in which case operator intervention will be necessary to resume the program.
- Because the Restart move starts from the current position, it is possible for the robot to hit something if the path is not clear between the current position and the stop position. For example, the robot can be jogged to a position that will cause a collision with a fixture when the Restart move executes.

10.34 PAYLOAD CONFIRM

10.34.1 Overview

Payload confirm consists of two functions:

- Payload Setting Confirm
- Payload Over Confirm

Payload setting confirm evaluates payload setting accuracy with little robot motion.. When this function is used, the robot has to be in the position where where J5 and J6 have maximum moment loading. Only axes J5 and J6 have to be at the maximum moment position for the current payload. The other axes have no limitation.

When using Payload Confirm, J5 and J6 axes move between the current position +1[degree] and the current position –1[degree] when F4, PLSET and F3, YES are selected. The payload setting error will be shown after robot moved. Payload setting error of payload [1]-[10] is stored.

Payload over confirm disables the overload warning when the motor is able to move in actual program with the payload that is over specification.

The program to be evaluated must be executed beforehand. When F5, OVLD is selected, this function evaluates the last executed program. The maximum torque of the last executed program and rated torque is compared. The overload warning is disabled when the maximum torque of the checked program is lower than the rated torque. Note that the overload warning is not shown even if the maximum torque of the other program is higher than the rated torque.

10.34.2 Payload Setting and Payload Over Confirm

Procedure 10-52 Using Payload Setting Confirm

1. Press MENUS.
2. Select SYSTEM.
3. Press F1, [TYPE].
4. Select PayLoad Check. You will see a screen similar to the one shown in Figure 10–119 .

Figure 10–119. Payload Check Screen

```
PAYLOADSET/OVLD              JOINT 10%
                                   1/9
 group1 Payload[10] Payload Setting Err.
  J4J5:0.0%  J6:0.0% [NOT DONE]
  Push 'PLSET' to check payload setting.

  Payload Setting Status [OVER LOAD!]
  run Prog in low OVR and push 'OVLD'key.

 Trq Chk[push 'OVLD' to check overload]
  J1:  0.00%   J2:  0.00%  J3:  0.00%
  J4:  0.00%   J5:  0.00%  J6:  0.00%
```

5. **To check the payload setting**, press F4, PLSET.
6. **To check the overload**, press F5, OVLD.

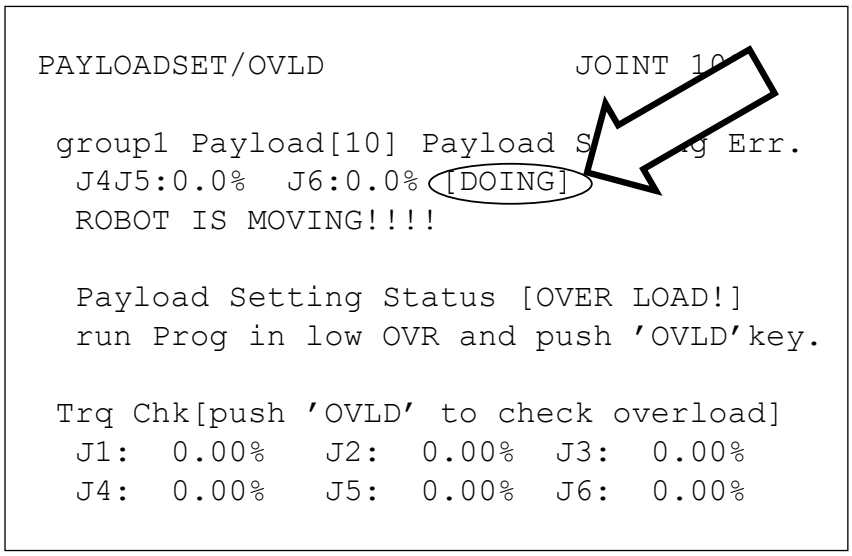
Procedure 10-53 Using Payload Setting Confirm when Payload is Changed

1. Abort all programs that are running or paused.
2. Move the robot to the position where the J5 and J6 axes are loaded almost maximum moment with current payload. **Confirm that robot does not interfere if J5 and J6 axes move between the current position+1[degree] and current position–1[degree].**
3. Turn off the teach pendant enable switch.
4. Display the Payload Confirm screen:
   a. Press MENUS.
   b. Select SYSTEM.
   c. Press F1, [TYPE].
   d. Select PayLoad Check.

5. Press F4, PLSET.

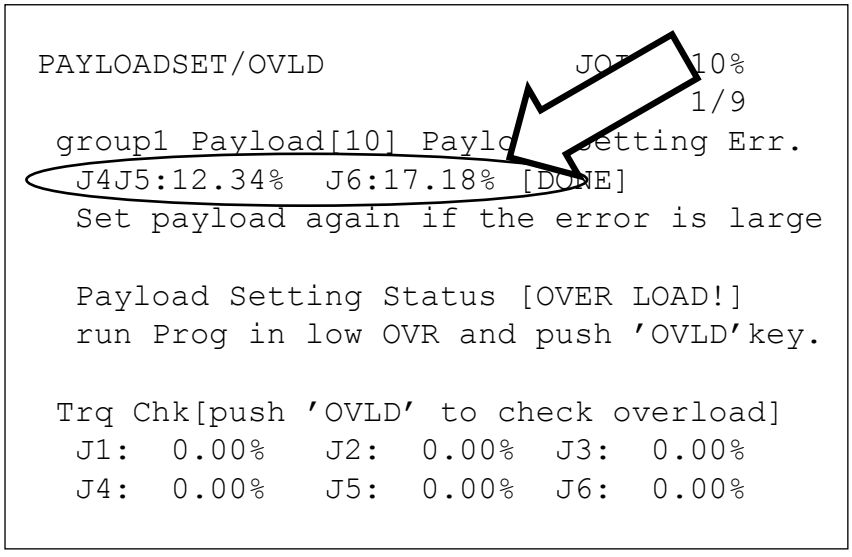
6. Press F3, YES. Wait until the right column of the line below Payload Setting Err. changes from DOING to DONE. This takes about 30 seconds. See Figure 10–120 .

Figure 10–120. Payload Setting Screen



7. If "Payload Setting Err." is large, payload setting might be incorrect. Threshold is 10% when payload identification is ordered and 20% when payload identification is not ordered. (The torque margin is not calibrated and error becomes larger if payload identification is not ordered). See Figure 10–121 .

Figure 10–121. Payload Setting Error



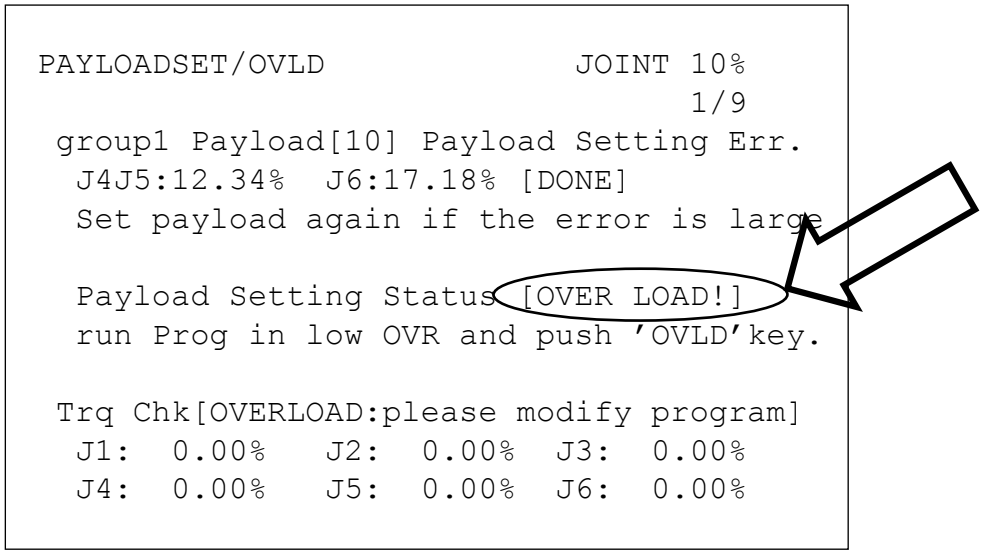
Note You can not execute payload setting confirm when other program is running or paused.

Procedure 10-54 Using Payload Over Confirm When the Program or Payload is Changed

Conditions

- Payload Setting Status is set to OVER LOAD. See the screen shown in Figure 10–122 .

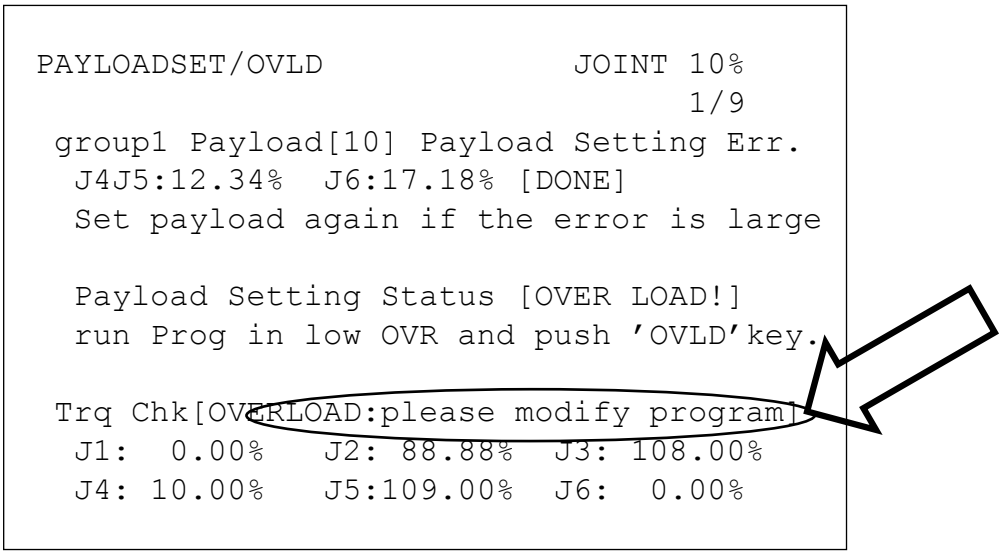
Figure 10–122. Modify Program Screen



- The program to be payload over confirmed has been executed. Elect the program to be payload over confirmed (press SELECT, select the program, and press ENTER.).

1. Set the override to 10% or lower, and execute the program.
2. Display payload over and payload setting confirm screen:
   a. Press MENUS.
   b. Select SYSTEM.
   c. Press F1, [TYPE].
   d. Select PayLoad Check.
3. Press F5, OVLD. You will see a screen similar to the one shown in Figure 10–123 .

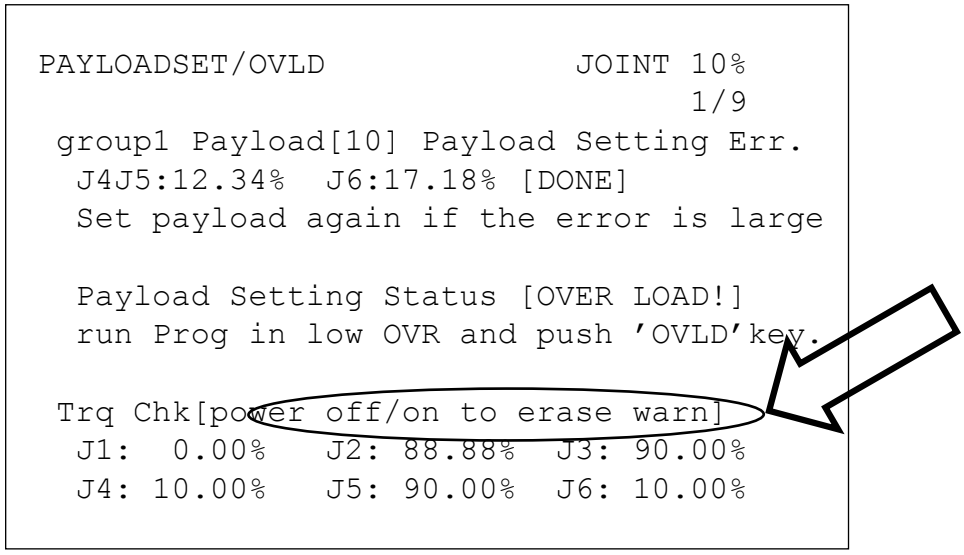
Figure 10–123. Modify Program Screen



4. Modify the program.

5. When Trq Chk becomes OK:power off/on to erase warn! See the screen shown in Figure 10–124 .

Figure 10–124. Power On/Off screen



Note The Overload warning is disabled when the maximum torque of the checked program is lower than rated torque. Note that overload warning is not shown even if maximum torque of the other programs are higher than rated torque. You must confirm that the maximum torque is lower than rated torque in all programs you use.

To enable the overload warning again, change the setting of MOTN-171 in the setup/error table screen from “NODISP” to “DEFAULT.”

6. The result shown in the screen that is displayed when the program running is not correct. Display the screen again after the program to evaluate has ended.

10.35 POSITION REGISTER LOOK-AHEAD EXECUTION FUNCTION

10.35.1 Overview

While the robot is executing a program, it reads the lines ahead of the line currently being executed (look-ahead execution). The position register look-ahead execution function enables look-ahead execution for position registers. To understand fully the features of the position register look-ahead function, it is helpful to understand some of the details of program execution.

Program Execution without the Position Register Look-Ahead Function

Conventionally, look-ahead execution is performed for motion instructions that have normal positional data (do not use position registers). Look-ahead execution can not be performed for motion instructions that use position registers for their positional data.

Motion instructions that use position registers cannot be read in advance because the values in the position registers could be changed by the program, data transfer function, and so forth.

If the robot reads a motion instruction that uses a position register prior to its execution, the value of the position register might yet be changed by a program or another function (such as data transfer). Such a change is not reflected in the motion instruction that has already been read by the robot. Consequently, the robot’s operation might be unpredictable.

Motion instructions that use position registers can be classified into two types:

- Motion instructions with the target position specified by a position register
- Motion instructions with an offset instruction where an offset is given by a position register

Even when a target position or offset is calculated during program execution, and a position register holding this calculation result is used with a motion instruction, look-ahead execution is not performed for the instruction, for the reason explained above.

Program Execution with the Position Register Look-Ahead Function

The position register look-ahead execution function enables look-ahead execution for position registers. For this purpose, an instruction to lock position registers and an instruction to unlock

position registers are provided. Using these instructions, you can explicitly specify a program portion. Then, for the specified program portion, even when it contains motion instructions that use position registers, look-ahead execution can be performed.

The position registers can be locked to prevent their contents from being changed after they are read. When an attempt is made to execute an instruction to change a locked position register (for example, an assign instruction for the position register, or an application instruction to set data in the position register), the following error message is issued:

```
INTP-128 Pos reg is locked
```

When a function other than the program (such as the data transfer function) attempts to change the value of a locked position register, the following error message is issued, and the attempt fails:

```
VARS-037 Position register is locked
```

Position registers are generally locked and unlocked with instructions taught in a program. When a program that has locked the position registers terminates, the position registers are unlocked automatically.

All position registers are locked simultaneously. While the position registers are locked, access to any position register is disabled, even in a different motion group.

10.35.2 Program Instructions

The following program instructions have been added for the position register look-ahead function:

- LOCK PREG
- UNLOCK PREG

LOCK PREG

Locks all position registers. This instruction prevents any change being made to any position register.

UNLOCK PREG

Unlocks the position registers.

These are control instructions, not motion instructions. They can be taught in the same way as other control instructions.

10.35.3 Program Example

Position Register Look-Ahead Program Example shows how to use the LOCK PREG and UNLOCK PREG instructions in a program.

Position Register Look-Ahead Program Example

```
1: J P[1] 100% FINE
 2:   PR[1]=PR[2]
 3:   PR[2]=PR[3]
 4: LOCK PREG
 5: L P[2] 100mm/sec Cnt100
 6: L P[3] 100mm/sec Cnt100
 7: L PR[1] 100mm/sec Cnt100
 8: L P[4] 100mm/sec Cnt100 offset, PR[2]
 9: L P[5] 100mm/sec FINE
10: UNLOCK PREG
```

When line 4 of this sample program has been executed, the position registers are locked. They are unlocked when line 10 has been executed. Therefore, the motion instructions with position registers in lines 7 and 8, which are executed with the position registers locked, are subject to look-ahead execution.

If the program is terminated between lines 4 and 10, the locked position registers are unlocked automatically. If the program is paused between lines 4 and 10, the cursor is moved manually, then the program is restarted, the locked position registers are unlocked. In this case, look-ahead execution is not performed for the instructions in lines 7 and 8.

When backward execution is performed, then normal execution is restarted, the position registers are unlocked. For example, suppose that program execution is paused during the execution of line 6, backward program execution is performed up to line 5, then forward program execution is restarted. In this case, the position registers are unlocked. So, look-ahead execution is not performed for lines 7 and 8.

When program execution is started from a line located after line 4, the position registers are not locked. So, look-ahead execution is not performed for lines 7 and 8.

A LOCK PREG instruction can be executed even when the position registers are already locked. Nothing occurs, however, when the LOCK PREG instruction is executed for a second time. Similarly, the UNLOCK PREG instruction can be executed even when the position registers are not locked. Nothing occurs, however, when the UNLOCK PREG instruction is executed for a second time.

10.35.4 Execution

When executing position register look-ahead program instructions, be aware of the following:

- The LOCK PREG and UNLOCK PREG instructions are not executed in backward program execution mode.
- Look-a head execution is not performed for the LOCK PREG and UNLOCK PREG instructions. This means that when one of these instructions is encountered, look-ahead execution is stopped temporarily; after the instruction is executed, look-ahead execution is again enabled.

10.36 PROGRAM SHIFT UTILITY

The program shift utility allows you to offset the positions of an entire teach pendant program or a portion of a teach pendant program. This is an easy way to adjust a teach pendant program after a fixture or the physical location of a robot has been changed.

Note For PaintTool and paint robots, this is almost always accomplished using program adjust. The program shift utility has little functionality to handle extended axes (rails) and line tracking programs. Use of this utility is not recommended for these applications.

Shifting a program can be accomplished either as a

- Parallel shift
- Parallel and rotational shift

You can shift the positions of an entire program or a portion of a program from within a program or from one program to another. In this way, robot paths can be transferred from one program to another or one robot to another in order to perform backups. See Figure 10–125 and Figure 10–126 .

Figure 10–125. Shifting an Entire Program

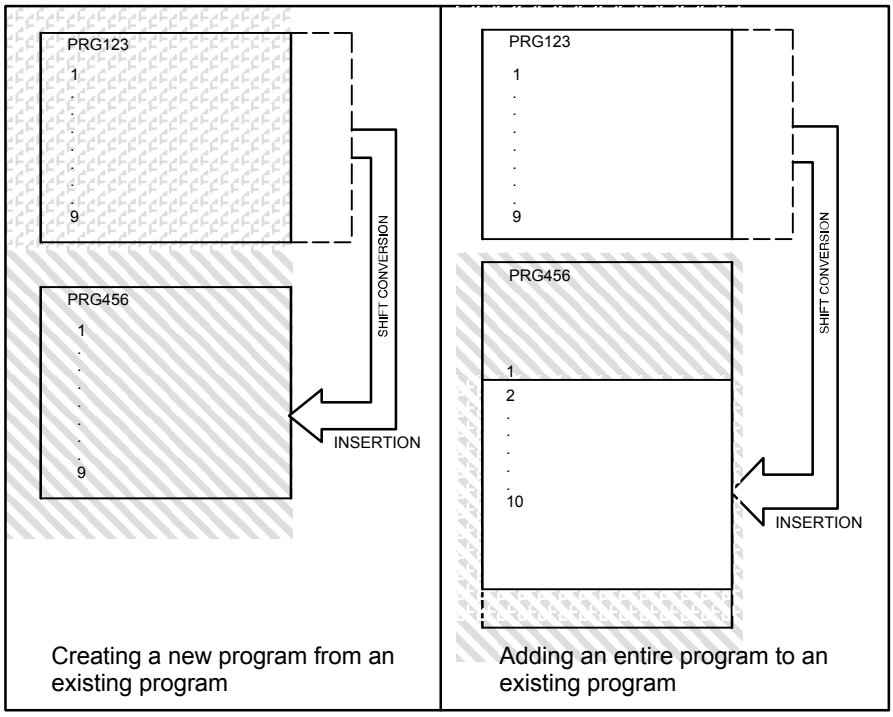
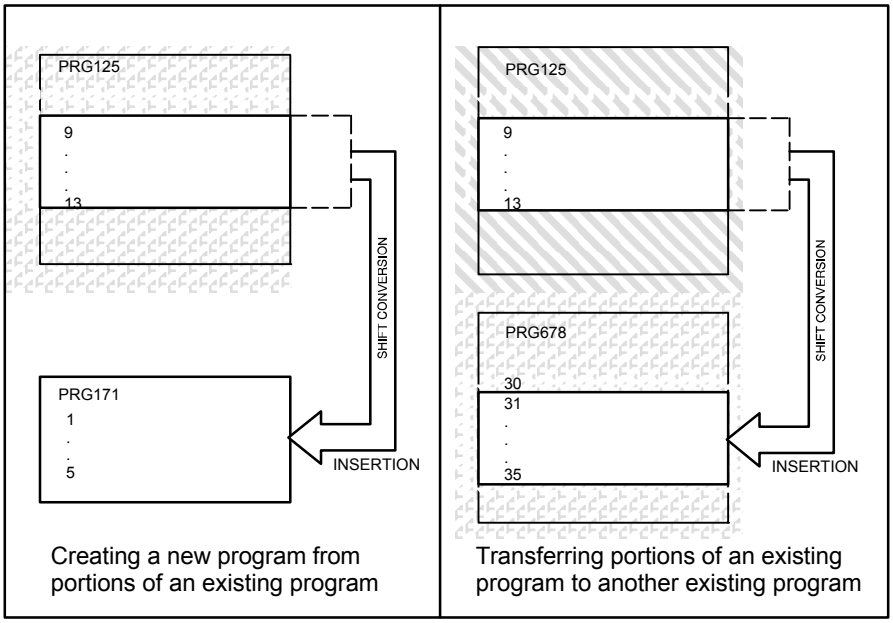


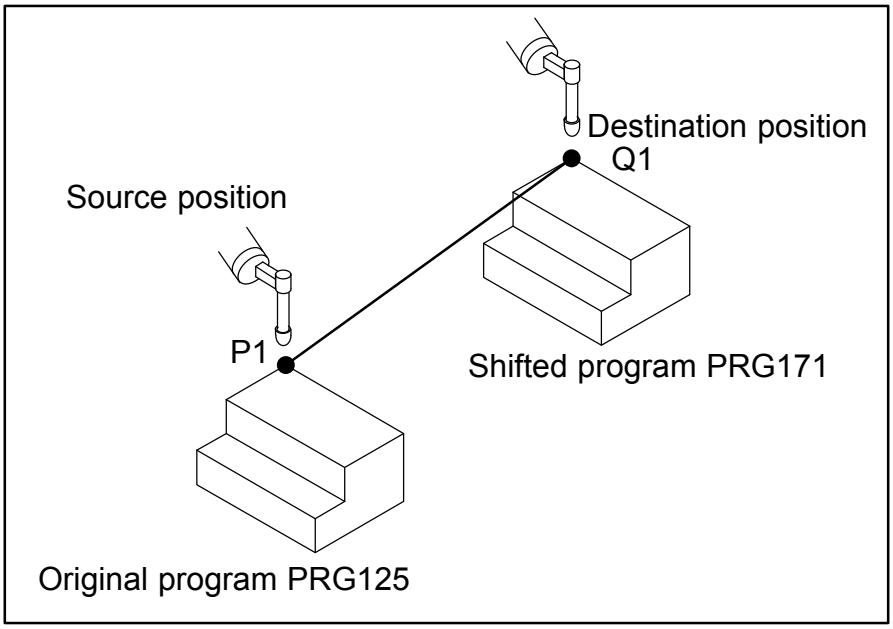
Figure 10–126. Shifting Portions of a Program



Parallel Shift

A parallel shift of a program is accomplished by reteaching the location of one point from the original (source) program, to the destination program. See Figure 10–127 .

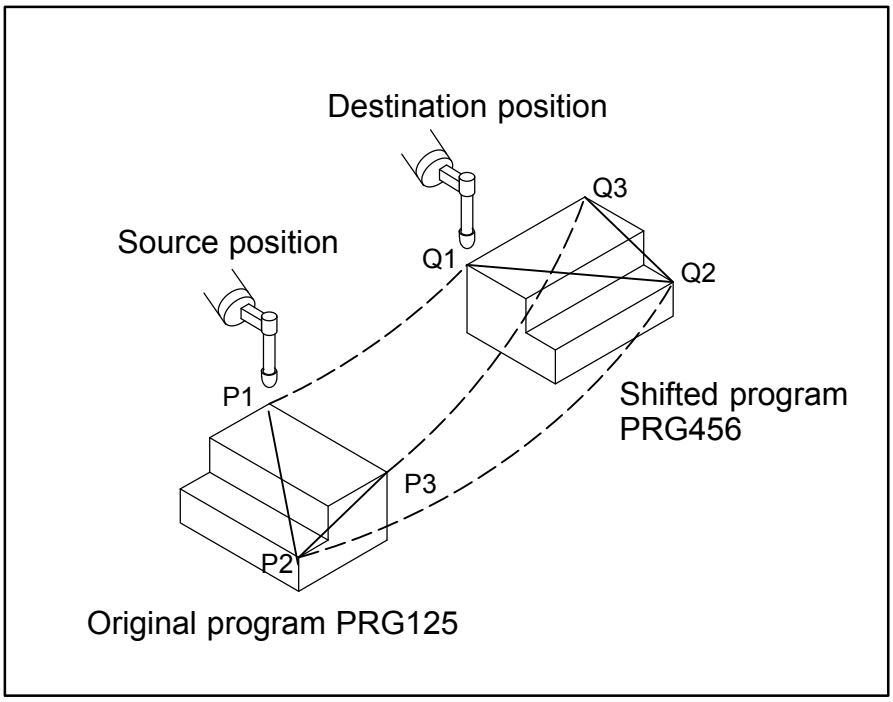
Figure 10–127. Parallel Shift



Parallel and Rotational Shift

A parallel and rotational shift is accomplished by reteaching the location of three points from the original (source) program (P1, P2 and P3) to the destination program (Q1, Q2 and Q3). See Figure 10–128 .

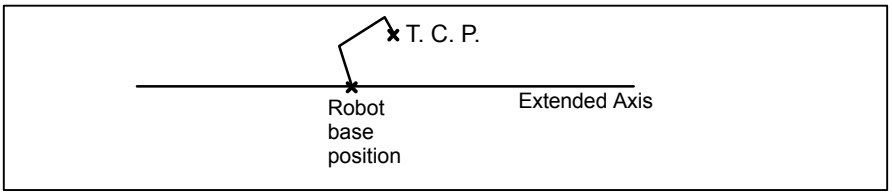
Figure 10–128. Parallel and Rotational Shift



Extended Axes

Extended (Ext) axes determines how the program shift function will translate the program when you are using extended axes. Figure 10–129 displays a key for use with Figure 10–130 through Figure 10–132

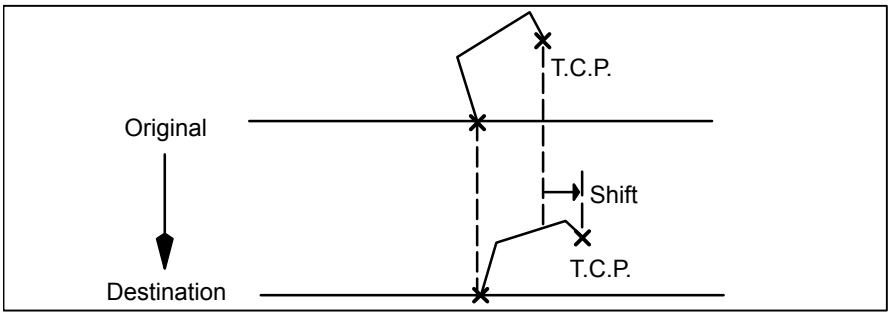
Figure 10–129. Program Shift Key



Possible extended axis types are:

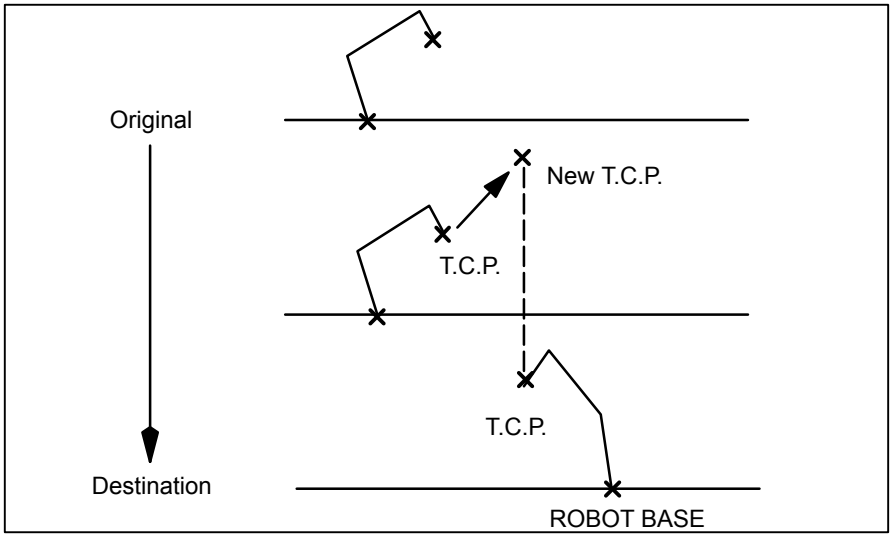
- **Robot axes only** - allows you to shift the axes of the robot without shifting a non-integrated extended axis such as a positioning table. The shift is calculated using the change in the robot tool center point (TCP). See Figure 10–130 .

Figure 10–130. Example of Robot Axes Only Shift



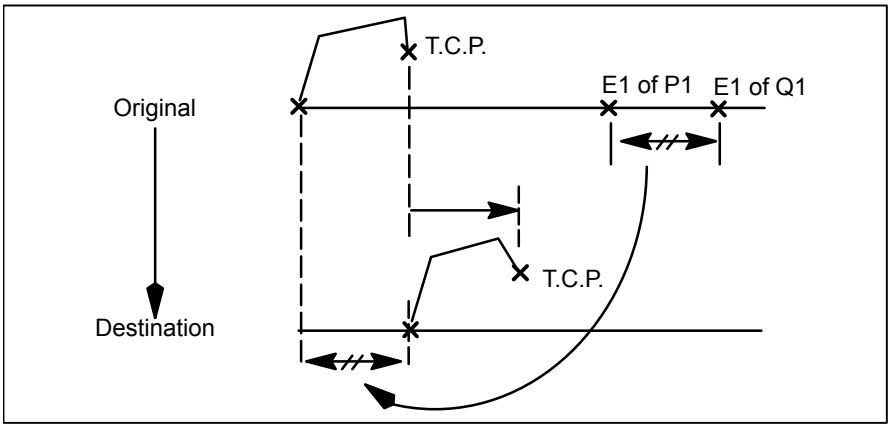
- **Extended integrated** - allows you to shift the axes of the robot and any integrated axes. The amount of shift for the robot and the extended axes is calculated using the change in the TCP. See Figure 10–131 .

Figure 10–131. Example of Extended Axes Integrated Shift



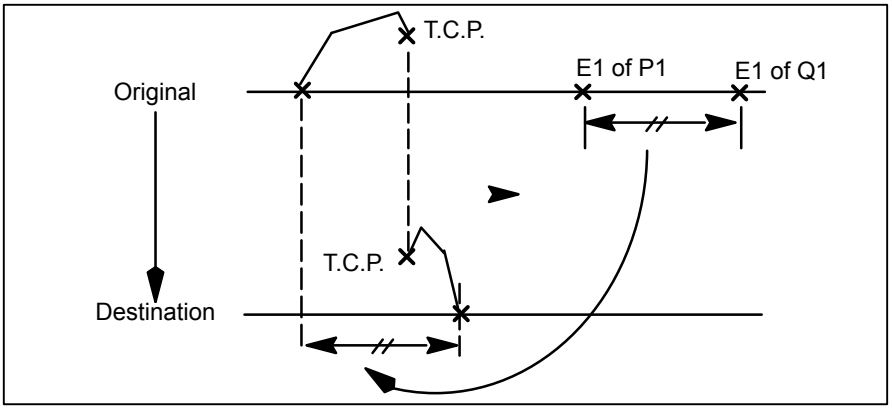
- **With extended axes** - allows you to shift positions for robot axes and any extended axes in your system. The amount of shift for the robot is calculated by using the change in the TCP. The amount of shift for the extended axes is calculated using the difference between an original position (P1) and a new position (Q1). See Figure 10–132 .

Figure 10–132. Example of With Extended Axes Shift



- **Extended axes only** - allows you to shift positions for the extended axes in your system while maintaining a constant TCP location. The amount of shift for the extended axes is calculated using the difference between an original position (P1) and a new position (Q1). See Figure 10–133 .

Figure 10–133. Example of With Extended Axes Only Shift



- **Replace Extended axes** - allows you to shift positions for only the extended axes in your system without affecting any robot angles. The shift amount for the extended axes is calculated using the difference between an original position (P1) and a new position (Q1). See Figure 10–134 .

Figure 10–134. Example of a Replace Extended Axes Shift

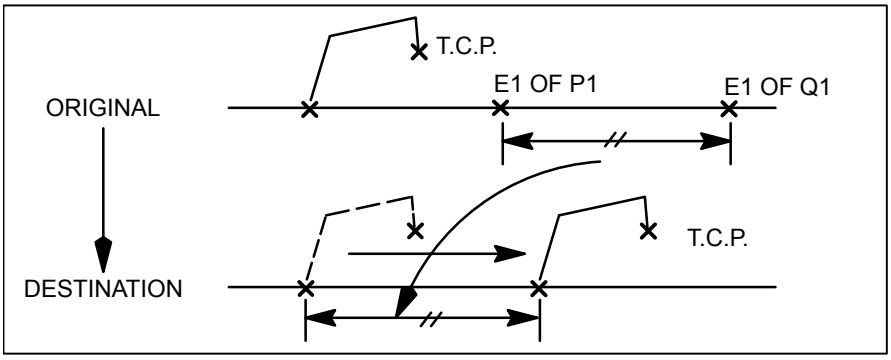


Table 10–70 lists the PROGRAM SHIFT setup items. Use Procedure 10-55 to perform a program shift.

Table 10–70. PROGRAM SHIFT Screen Items

ITEM	DESCRIPTION
Original Program	This item is the name of the program that you want to shift. To select a program, press ENTER, type the name of the program, and press ENTER again.
Range	This item indicates whether the whole program or a portion will be shifted.
Start line	This item is the first line in the program that will be shifted.
End line	This item is the last line in the program that will be shifted.
New Program	This item is the name of the program that results when you offset or shift the Original Program. If you want the resulting offset or shifted program to replace the Original Program, make the New Program name the same as the Original Program name.
Insert line	This item indicates the location in an existing program where new shifted position information will be inserted.
Position data	This item is the current x, y, and z position information.
Rotation	This item, if set to ON, indicates that you want to rotate three position about an axis. If set to OFF, you can shift a single source position to a single destination position.
Source position	This item is the original position that you will be shifting. If Rotation is set to ON, there will be three source positions.
Destination position	This item is the destination position to which you will be shifting. If Rotation is set to ON, there will be three destination positions.

Procedure 10-55 Using the Shift Utility

Conditions

- The program you want to shift has been created and contains recorded positions.

Steps

1. Press MENUS.
2. Select UTILITIES.
3. Press F1, [TYPE].
4. Select Program shift.
5. Move the cursor to Original Program. If the program you want to shift is not selected, press ENTER. Use the appropriate function keys to type the name of the program and press ENTER.
6. Move the cursor to Range and select to shift the WHOLE program or PART of the program.

   - **To shift the whole program press F5, WHOLE.** You will see a screen similar to the following.

```
PROGRAM SHIFT
Program
 1 Original Program:                    [STYLE37]
 2   Range:                              WHOLE
 3   Start line:  (not used)             ****
 4   End Line:    (not used)             ****
 5 New Program:                         [       ]
 6  Insert LINE:  (not used)             ****
 7 EXT axes :             Robot axes only
Use shifted up, down arrows for next page
```

   - **To shift part of the program press F4, PART.** You will see a screen similar to the following.

```
PROGRAM SHIFT
Program
 1 Original Program:                    [STYLE37]
 2   Range:                              PART
 3   Start line:                              0
 4   End Line:                                0
 5 New Program:                         [       ]
 6  Insert LINE:  (not used)                ****
 7 EXT axes :             Robot axes only
Use shifted up, down arrows for next page
```

7. If you selected to shift PART of a program,

- Move the cursor to Start line and type the starting line number.
- Move the cursor to End line and type the ending line number.

8. Move the cursor to New Program and press ENTER. Use the appropriate function keys to type the name of the new program and press ENTER. This is the program to which you will be shifting the positions.

- This can be the same as the original program, an already existing program, or a program that does not exist.

9. If you are inserting lines into an existing program, move the cursor to Insert line and type the line number at which you want to insert the shifted information.

10. Press and hold the SHIFT key while pressing the down arrow key. You will see the Shift Position screen.

11. Move the cursor to Rotation. You will see a screen similar to the following.

```
PROGRAM SHIFT
Shift amount/Teach
   Position data
   X :********  Y :********  Z :********
  1 Rotation:                     OFF
  2 Source position          P1:
  3 Destination position     Q1:
```

- **To rotate the positions,** press F4, ON. You will see a screen similar to the following.

```
PROGRAM SHIFT
Shift amount/Teach
  Position data
  X :********  Y :********  Z :********
  1 Rotation:                    ON
  2 Source position        P1:
  3                        P2:
  4                        P3:
  5 Destination position   Q1:
  6                        Q2:
  7                        Q3
```

Note You cannot type positional data directly if you are performing a rotation. If you attempt to type positional data directly, Rotation will be set to OFF automatically.

12. **To add the instructions directly** if you know the exact positional data,

    a. Press NEXT, >.

    b. Press F2, DIRECT. You will see a screen similar to the following.

```
PROGRAM SHIFT
Shift amount/Direct entry
  1 X (mm)            0.00
  2 Y (mm)            0.00
  3 Z (mm)          500.00
```

    c. Type each x, y, and z shift amount.

    d. Press F2, EXECUTE.

```
Execute transform  YES NO
```

    e. Press F4, YES.

    f. **When you are finished** , press PREV or F2, TEACH. You will see a screen similar to the following.

```
PROGRAM SHIFT
Shift amount/Teach
  Position data
  X :********  Y :********  Z :********
  1 Rotation:                      ON
  2 Source position       P1:
  3                       P2:
  4                       P3:
  5 Destination position  Q1:
  6                       Q2:
  7                       Q3:
```

13. Move the cursor to Source position (see Figure 10–127 and Figure 10–128).

14. Move the robot to the first source position (P1) and either record or specify the position:

- **To record a position,** jog the robot to the position you want, press and hold in the SHIFT key and press F5, RECORD.
- **To specify a previously recorded position or position register,** press F4, REFER. Select the position or position register, and press ENTER.
- **To clear a position setting** , press NEXT, >, and then press F1, CLEAR. At the prompt "Clear all data," press F4, YES, to confirm.

15. **If you are rotating the positions** , record or specify all the source positions.

16. Move the cursor to Destination position (see Figure 10–127 and Figure 10–128). See the following screen for an example.

```
PROGRAM SHIFT
PROGRAM SHIFT (POSITION)
  Position data
  X :********  Y :********  Z :********
  1 Rotation:                      ON
  2 Source position         P1:    P[1]
  3                         P2:    P[2]
  4                         P3:    P[3]
  5 Destiniation position   Q1:
  6                         Q2:
  7                         Q3:
```

- **To record a position**, jog the robot to the destination position (Q1). Press and hold in the SHIFT key and press F5, RECORD.

- **To specify a previously recorded position register**, press F4, REFER. Type the number of the position register and press ENTER.

```
Input position register number:
```

- **To clear a position setting**, press NEXT, >, and then press F1, CLEAR. At the prompt "Clear all data," press F4, YES, to confirm.

17. **If you are rotating the positions**, record or specify all the destination positions.
18. **To shift the program**, press F2, EXECUTE.

```
Execute transform  YES NO
```

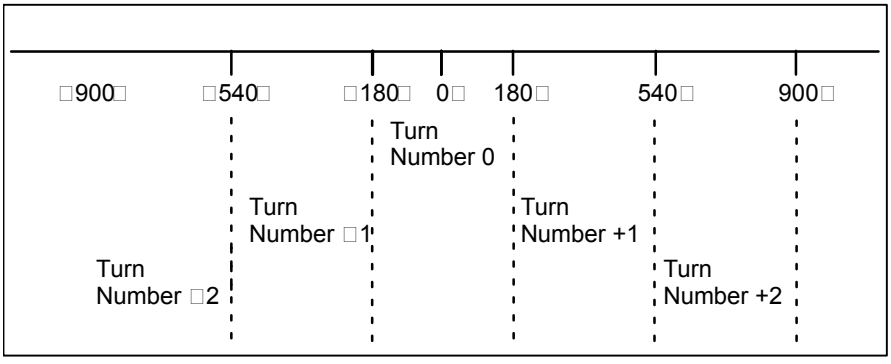
- **To execute the shift**, press F4, YES.
- **To not execute the shift**, press F5, NO.

Note For each position to shift, you will receive the message ‘‘Select P[n]: m angle (deg p)’’ (where n equals a position number of the destination program and m equals the amount of angle) only if the shift causes:

— A joint to wrap greater than 180 degrees.

— The turn number of the joint to change if the wrap is less than 180 degrees. See Figure 10–135 .

Figure 10–135. Turn Numbers



19. **Select the kind of angle to use** .

```
Select P[1]: J120 angle
deg p  deg q  *unint*  QUIT
```

- **To allow joints to wrap and/or turn number to change**, press F1, deg p. Normally, the angle change will be less than 180 degrees, but will have a different turn number.
- **To not allow any wrapping or a change in turn number**, press F2, deg q. The angle change will be greater than 180 degrees, but the turn number is the same.
- **To not allow any wrapping, a change in turn number**, **and any angle changes** , press F3, *unint*. The joint angles for that position will remain uninitialized. You will have to reteach the position after the transformation has completed.

> **Warning**
>
> **F5, QUIT, will insert the original position into the SHIFT or MIRROR program. Be sure this is what you want to do before you press F5, QUIT; otherwise, you could injure personnel or damage equipment.**

- **To stop the shift for each position**, press F5, QUIT.

20. Wait until the software has finished processing the shift.

```
-PROCESSING-
```

10.37 REFERENCE POSITION UTILITY

A *reference position* defines position limits within which an output signal will turn on. To use the reference position utility you specify a joint position for each axis of your robot, assign an output signal to the position, and, optionally, specify tolerance ranges for each axis joint position.

When the robot moves to within the tolerance range of the specified reference position joint axis locations, the assigned digital output (DO) or robot output (RO) signal turns on. When the robot moves out of the tolerance range, the DO or RO signal turns off. **If a tolerance range is not specified, every axis of the robot must be at the EXACT reference position joint axis location for the signal to turn on.**

To use a reference position, your program must contain a taught joint position whose axes locations match those of the reference position.

In a multiple motion group system, independent reference positions are available for each group, as follows:

- For **group 1**, three independent reference positions are available, and a digital output is available for each of these reference positions.
- For **groups 2–8**, one reference position and one digital output is available for each group.

Note You can increase the number of reference positions for each motion group by setting the system variable $REFPOSMAXNUM[n], where“n” is the number of the motion group. The maximum number of reference positions is 64. If you want to change the number of reference positions, change the value of $REFPOSMAXNUM[n] for the motion group you want to change. Turn off and then turn on the controller for the change to take effect. The maximum number of enabled reference positions at any one time is 10.

The Reference Position Output operates in all modes - T1, T2, and Auto.

Two screens are associated with the reference position utility: the **LISTING** screen and the **DETAIL** screen. The listing screen allows you to view limited information for all reference positions. The detail screen allows you to view all information for a single reference position.

Table 10–71 lists and describes each item on the LISTING screen. Table 10–72 lists and describes each item on the DETAIL screen.

Table 10–71. Reference Position LISTING Screen Items

ITEM	DESCRIPTION
No.	This item shows the number of available reference positions. Three is the maximum. **Note** In DispenseTool, Reference position #1 is reserved and will be automatically assigned as the last position in the MOV_HOME.TP program
Enb/Dsbl	When set to **Enable** , this item allows the system to check whether the robot is at the specific joint axes positions when no tolerance range is set, or within the specified tolerance range of the joint reference position. If the robot is in the range, the specified signal is turned on. If the robot goes out of the range, the signal turns off. This can be set from the LIST screen or the DETAIL screen. When set to **Disable,** this item allows the system to ignore the reference position check. This can be set from the LIST screen or the DETAIL screen.
@Pos	This item indicates whether the robot is currently at any reference position that is enabled.
Comment	This item allows you to enter a comment about the reference position. This can be set from the LIST screen or the DETAIL screen.

Table 10–72. Reference Position DETAIL Screen Items

ITEM	DESCRIPTION
Reference position number	This item indicates the reference position number the screen is currently displaying.
Comment	This item allows you to enter a comment about the reference position. This can be set from the LIST screen or the DETAIL screen.
Enable/Disable	When set to **Enable** , this item allows the system to check whether the robot is at the exact joint axes positions when no tolerance range is set, or within the specified tolerance range of the joint reference position. If the robot is in the range, the specified signal is turned on. If the robot goes out of the range, the signal turns off. This can be set from the LIST screen or the DETAIL screen. When set to **Disable** , this item allows the system to ignore the reference position check. This can be set from the LIST screen or the DETAIL screen.
Is a Valid HOME? If your system has home checking	This item allows you to specify whether this reference position will be checked as a home position, if your system has home checking. When Is a Valid HOME is set to TRUE, this reference position will be checked as a home position. **Note** In DispenseTool, Reference position #1 is reserved and will be automatically assigned as the last position in the MOV_HOME.TP program
Signal Definition	This item allows you to specify the digital output or robot output signal that turns on and off as the robot moves in and out of the specified tolerance range.
Joint Axis Location J1 through J[n]	This item allows you to enter the angle of each joint for your robot that together will form the reference position.
Tolerance Range +/- 0.000	This item allows you to specify the acceptable position limits, or tolerance range, of each joint. After the robot is within these position limits, the assigned digital output (DO) or robot output (RO) signal turns on. A tolerance of 0.5 to 1.0 per axis usually works well.

Procedure 10-56 Setting the Reference Position

Steps

1. Press MENUS.
2. Press SETUP.
3. Press F1, [TYPE].
4. Select Ref Position. You will see a screen similar to the following.

```
REF POSN
                          GROUP: 1
No.  Enb/Dsbl  @Pos       Comment
 1   DISABLE   FALSE  [              ]
 2   DISABLE   FALSE  [              ]
 3   DISABLE   FALSE  [              ]
```

5. Press F3, DETAIL, to select a reference position. You will see a screen similar to the following.

```
REF POSN
Reference Position     Group: 1
   Ref. Position Number:          1
 1 Comment:                [safe position  ]
 2 Enable/Disable:                    ENABLE
 3 Is a valid HOME:                   FALSE
 4 Signal definition:                 DO[  0]
 5  J1 :      0.000     +/-            0.000
 6  J2 :      0.000     +/-            0.000
 7  J3 :      0.000     +/-            0.000
 8  J4 :      0.000     +/-            0.000
 9  J5 :      0.000     +/-            0.000
10  J6 :      0.000     +/-            0.000
```

Note For more information about setting up a reference position, refer to Table 10–72 .

6. To enter a comment

a. Move the cursor to Comment and press ENTER.

b. Use the alphanumeric keys to enter the comment.

7. To enable or disable the reference position

a. Move the cursor to Enable/Disable.

b. Press either F4, ENABLE, or F5, DISABLE.

8. To specify whether the current reference position is a home position

a. Move the cursor to Is a valid HOME.

b. Press either F4, TRUE, for a home position, or F5, FALSE, if it is not a home position.

9. To define the signal

a. Move the cursor to Signal definition.

b. Press either F4, DO, for digital output or F5, RO, for robot output.

c. Move the cursor to the right and enter a value for the signal number.

10. To enter the joint axis location of a reference position

a. Move the cursor to a joint axis.

b. Press and hold the SHIFT key and use the arrows keys on the teach pendant, to jog the robot to the reference position.

Note If the position has already been recorded in a program, refer to Procedure 7-8 for instructions on how to move the robot to that position.

c. Press SHIFT, F5, RECORD to record the position. This records the current position of all the joint axes.

11. To enter a tolerance range (+/-0.000)

a. Move the cursor to each joint axis tolerance range.

b. Type a value for the tolerance range and press ENTER.

Note If a tolerance range is not specified, every axis of the robot must be at the EXACT reference position joint axis location for the signal to turn on.

12. If your controller is configured for multiple groups and you want to change the group number

a. Press F2, [GRP#].

b. Type the group number you want to use for the reference position and press ENTER.

13. Press PREV display the previous screen.

10.38 RESUME OFFSET

Resume Offset allows the Restart move to go to a previous position of the path, “offset” from the normal stop position. Processes that are continuous along the path, such as cutting, welding, or dispensing, use this feature. For example, sealant dispensing applications are sensitive to the TCP speed, so resume offset is used to allow time to accelerate to speed along the path before reaching the original stop position and resuming sealant flow.

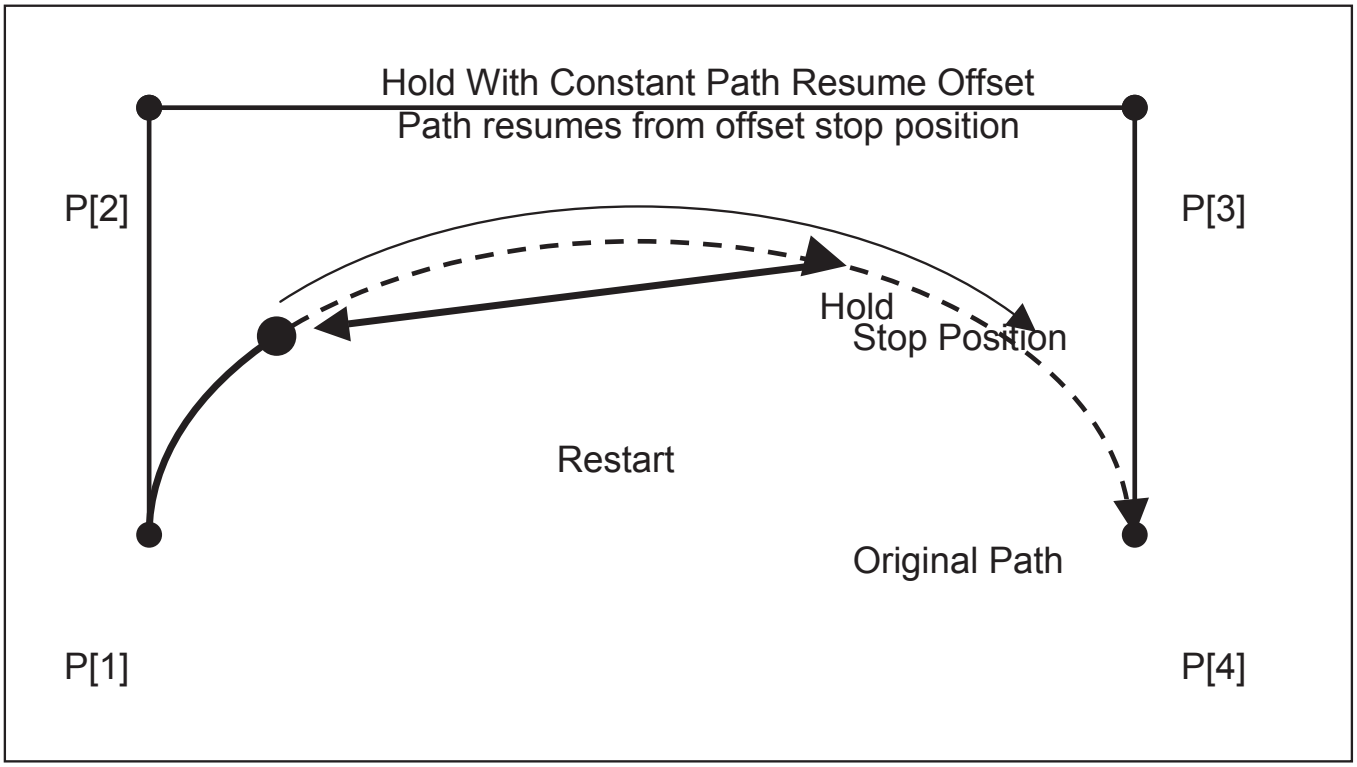
10.38.1 Limitations of Resume Offset

- Resume Offset requires Original Path Resume. Resume Offset is subject to the limitations of Original Path Resume.
- Resume Offset uses the same Restart move as Original Path Resume. The Restart move goes directly to the Resume Offset stop position, not backward along the original path.
- The actual Resume Offset position will be close to the desired offset, but usually not exact.

10.38.2 Constant Path Resume Offset

Constant Path Resume Offset works in conjunction with the Constant Path Option to maintain a "memory" of the TCP path, so that at the end of Hold the stop position can be offset to any position in recent memory. The Resume move will accelerate from the offset stop position, follow the identical path to the original stop position, and then continue execution normally.

Figure 10–136. Hold With Constant Path Resume Offset — Path Resumes from Offset Stop Position



10.38.3 Limitations of Constant Path Resume Offset

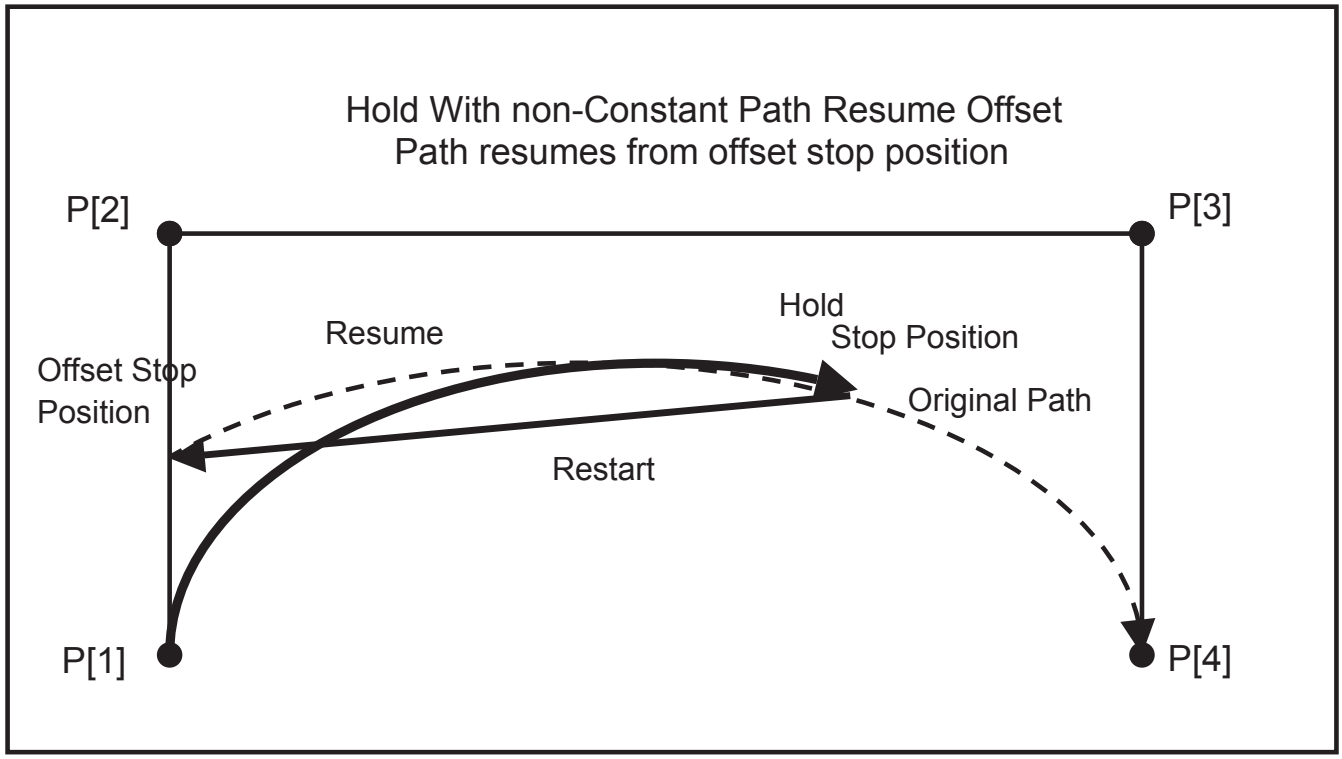
- Constant Path Resume Offset requires the Constant Path option. Constant Path Resume Offset is subject to the limitations of the Constant Path option.
- Constant Path Resume Offset requires the offset distance to be set before the robot has stopped due to Hold. Changing the offset distance after Hold but before Resume has no effect.
- Constant Path Resume Offset can't offset past the beginning of recent memory. Attempting to do so will only move to the earliest position in recent memory. The recent memory size can be increased using the Setup->Resume Offset menu.

10.38.4 Non-Constant Path Resume Offset

Non-Constant Path Resume Offset computes the offset on the current segment, not on the path. It is used in situations where Constant Path Resume Offset is not supported. The resumed path is generally not on the original path, but will be close for applications with slow program speeds or limited corner blending.

With Non-Constant Path Resume Offset, the offset distance can be changed any time until resume. There is no need to set the offset distance before Hold is complete.

Figure 10–137. Hold with non-Constant Path Resume Offset — Path Resumes from Offset Stop Position



10.38.5 Limitations of non-Constant Path Resume Offset

- Non-Constant Path Resume Offset records a stop position on the taught segment, not necessarily on the path. This can lead to path deviation at resume.
- Non-Constant Path Resume Offset can't offset past the start position of the current segment. Attempting to do so will only move to the start position of the current segment.

10.38.6 Selecting Resume Offset Type

Table 10–73 describes how the style of Resume Offset is selected:

Table 10–73. Selecting Resume Offset Type

$cpcfg.$resume_ofst.$ro_enable	Constant Path Resume	Non-Constant Path Resume
FALSE	No Resume Offset	Non-Constant Path Resume Offset
TRUE	Constant Path Resume Offset	Non-Constant Path Resume Offset

When Constant Path option is loaded and Constant Path Resume is used, the value of $CPCFG.$RESUME_OFST.$RO_ENABLE determines whether or not Resume Offset is applied.

When Constant Path option is not loaded, non-Constant Path Resume Offset is used.

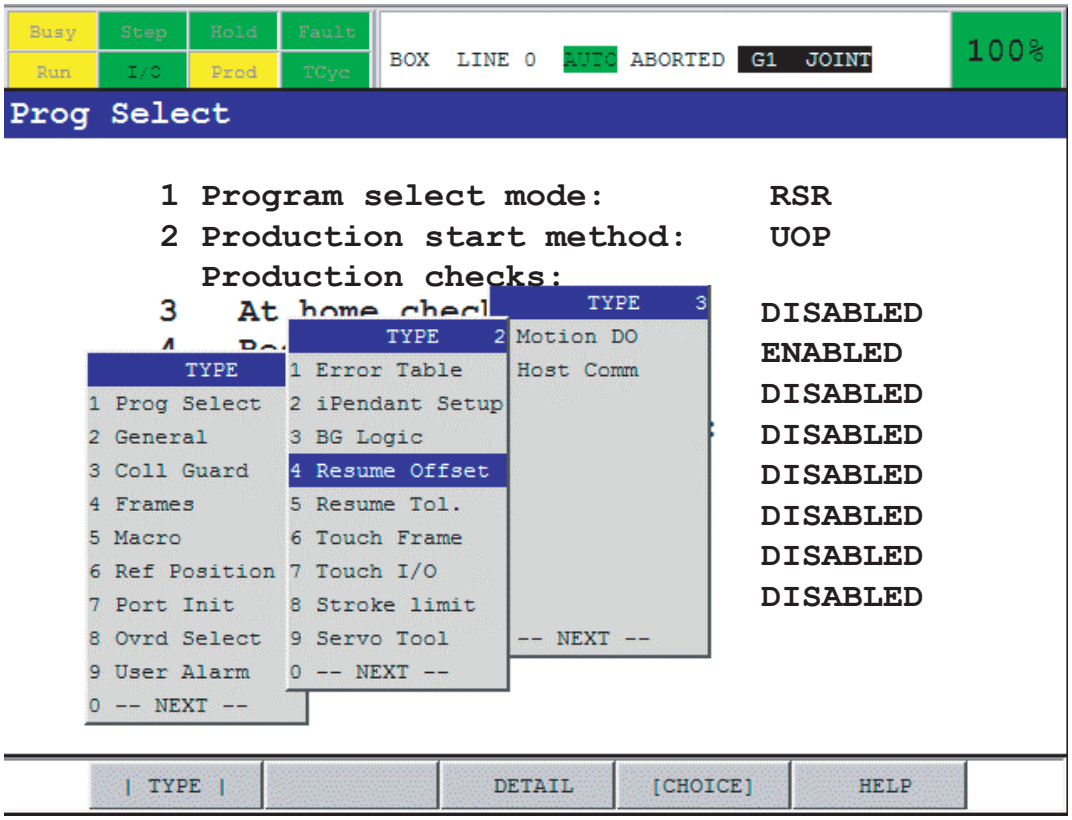
Some options do not support Constant Path Resume. When these options are used, Non-Constant Path Resume is used. These options include:

- Touch Sensing
- Line Tracking
- ServoGun (in SPOT motion option)
- Weave

10.38.7 Resume Offset Setup Menu

This menu can be found in menu->setup->Resume Offset.

Figure 10–138. Resume Offset Setup Menu



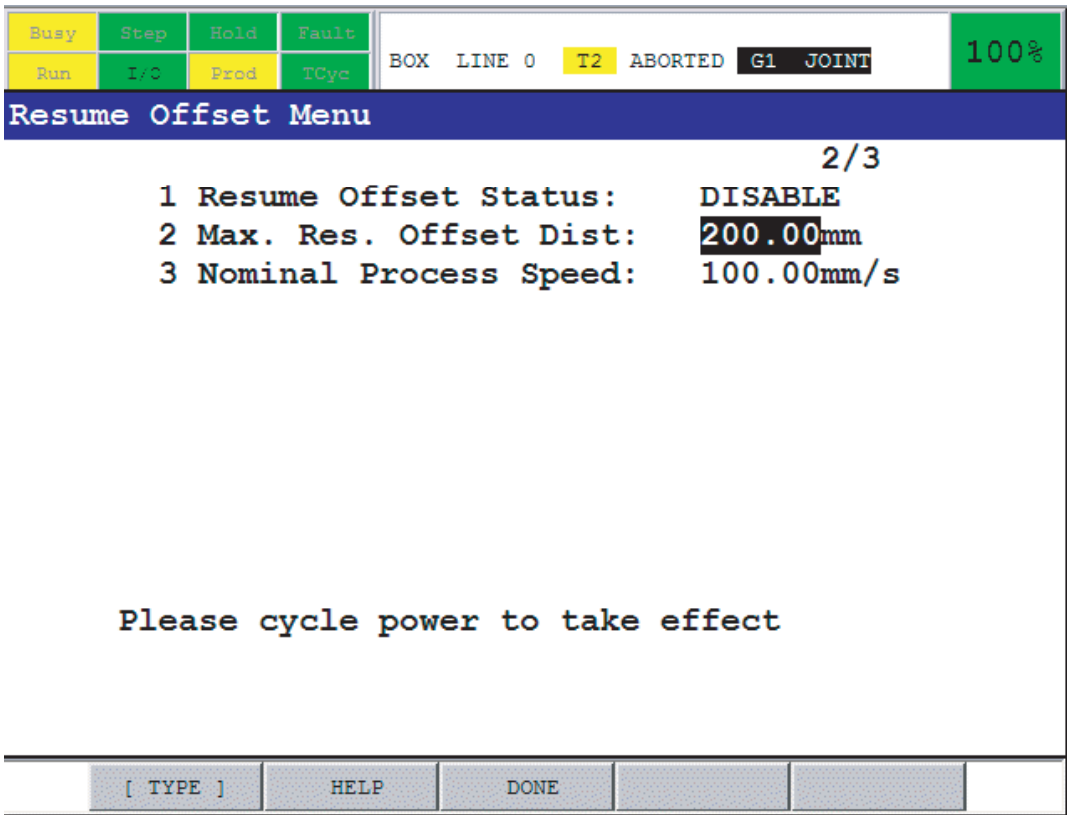
The Resume Offset Setup Menu is used to configure the recent memory for Constant Path Resume Offset. This configuration has a direct effect on system memory usage. Changes to the configuration require a power cycle to take effect.

Table Table 10–74 lists and explains the Resume Offset Setup Menu items.

Table 10–74. Resume Offset Setup Menu Screen Items

Item	Description
Resume Offset Status	This item indicates whether Constant Path Resume Offset is enabled. This is the same as the sysvar $CPCFG.$RESUME_OFST.$RO_ENABLE.
Max. Res. Offset Dist	This item is the maximum resume offset you expect to use. Larger values require more memory.
Nominal Process Speed	This item is the nominal resume offset speed: the program speed at which you want your process to execute. Smaller values require more memory.

Figure 10–139. Resume Offset Setup Menu Screen Items



If the configuration is changed, you will see a screen similar to the following.

```
Is it ok to cycle power now?
Please note change required
cycling power to take effect
Answer NO will remove change
     [ YES ]          NO
```

You must cycle power for changes to the configuration to take effect. You can either cycle power immediately to apply the changes, or reject the changes.

10.38.8 System Variables

$CPCFG.$RESUME_OFST.$RO_ENABLE
Enables Constant Path Resume Offset when Constant Path Option is loaded.

$MCR_GRP[].$RSM_OFFSET
Distance of Resume Offset to use (mm)

$MOR_GRP[].$OGDST_RATIO
Ratio of actual resume offset achieved to requested resume offset.
1.0 = 100%
0.0 = 0%
-1.0 = uninit

10.39 ROBOT-PC COMMUNICATIONS MACROS

10.39.1 Overview

A teach pendant program can coordinate with a program running on a PC using the macros described in this section. These macros "wake up" the PC application by raising an Event and optionally Send Data that can be used immediately by the PC program.

The robot to PC communication macros require you to have the PC communication option loaded, and requires that you have a program developed under PCDK running on a host PC. Refer to the PC Developer's Kit Help file for details on writing this program.

The five communication macros are listed below.

- Send Event( )
- Send Data( )
- Send SysVar( )
- Get Data( )
- Request Menu( )

The following sections describe the syntax and parameters for the communication macros.

10.39.2 Send Event ( )

The Send Event macro is used to send (or signal) an Event to the PC. To receive this Event, the PC must be running an application based on the FANUC Robot Server and have an FRCPacketEvent Event Handling active, waiting for this request code.

Table 10–75 lists and describes the Send Event macro. Send Event Sample Teach Pendant Program shows a sample teach pendant program that uses the Send Event macro.

Table 10–75. Send Event Macro Description

MACRO ITEM	DESCRIPTION
Syntax:	Send Event(request_code[, wait_sw[, status_reg]])
Parameters:	**request_code** - This item determines which event is triggered on the PC. It can be a value between 0 and 254.
	Wait_sw - This item is a wait switch. Valid values are either 0 or 1. **If 0,** do not wait for an answer from the PC. **If 1,** wait for an answer from the PC. The default value of **wait_sw** is 0. This parameter is optional if the parameter **status_reg** is also not entered.
	Status_reg - This item is a register which stores the value of either the answer or the error status for which the teach pendant program is waiting. Valid values are either 0 or a register number. . **If 0,** no register is used, and no answer from the PC is reported back to the teach pendant program **If wait_sw = 1** , then the teach pendant program waits for the PC to answer, but the answer value is not placed into a teach pendant register. If an error occurs in the macro, the teach pendant program is aborted. **If status_reg is not 0,** this is the register number used to report the PC's answer or error status to the calling teach pendant program. **If wait_sw = 0,** the teach pendant program will not wait for the PC to answer and the register will only be used to report errors. R[number] is set to 9999 if an error occurs. The default value of **status_reg** is 0. This parameter is optional.

Send Event Sample Teach Pendant Program

```
PROG TEST1
   1: ! Send Event 5
   2: ! No answer from PC
   3: Send Event (5)
PROG  TEST2
   1: ! Send Event 7
   2: ! Wait for PC answer
   3: ! Answer in Reg 5
   4: Send Event (7,1,5)
   5: If R[5]<9999,JMP LBL[10]
   6: ! Error in macro
   7: !
   8: LBL[10]
```

10.39.3 Send Data( )

The Send Data macro is used to send up to seven values to a PC application that is waiting for the event. The request code tells the PC application what data is being sent and what the PC is supposed to do with it.

Table 10–76 lists and describes the Send Data macro. Send Data Sample Teach Pendant Program shows a sample teach pendant program that uses the Send Data macro.

Table 10–76. Send Data Macro Description

MACRO ITEM	DESCRIPTION
Syntax:	Send Data(request_code, wait_sw, status_reg, val1[,val2[,val3[,val4[,val5[,val6[,val7]]]]]])
Parameters:	**request_code** - This item determines which event is triggered on the PC. It can be a value between 0 and 255.
	Wait_sw - This item is a wait switch. Valid values are either 0 or 1. **If 0,** do not wait for an answer form the PC. **If 1,** wait for an answer from the PC. The default value of **wait_sw** is 0.
	Status_reg - This item is a register which stores the value of either the answer or the error status for which the teach pendant program is waiting. Valid values are either 0 or a register number. **If 0,** no register is used, and no answer from the PC is reported back to the teach pendant program. **If wait_sw = 1** , then the teach pendant program waits for the PC to answer, but the answer value is not placed into a teach pendant register. If an error occurs in the macro, the teach pendant program is aborted. **If status_reg is not 0,** this is the register number used to report the PC's answer or error status to the calling teach pendant program. **If wait_sw = 0,** the teach pendant program will not wait for the PC to answer and the register will only be used to report errors. R[number] is set to 9999 if an error occurs. The default value of **status_reg** is 0. This parameter is optional.
	Val1 through Val7 - These items are data that are sent to the PC. Valid items are: INTEGER constants, REAL constants, the values of registers, and STRING constants. At least one value must be sent, but the others are optional. **If no data is to be sent,** use the Send Event( ) macro instead.

Send Data Sample Teach Pendant Program

```
PROG TEST1
   1: ! Send Event 42 and Data
   2: ! No answer from PC
   3: Send Data(42,0,0,'PARTS_ITEM3',R[1],5,6,7.1,8,
```

```
               'TOOL_ITEM1')
PROG TEST2
   1: ! Send Event 9 and data
   2: ! Wait for PC answer
   3: ! Answer in REG 3
   4: Send data (9,1,3,'TOOLING',R[1],5,'TEST')
   5: IF R[3]<9999,JMP LBL[10]
   6: ! Error in macro
   7: !
   8: LBL[10]
```

10.39.4 Send SysVar( )

The Send SysVar macro is used to send the value of a system variable (whether integer, real, or string) to the PC's Event Handler. The request code determines what data is being sent and what the PC is supposed to do with it.

Table 10–77 lists and describes the Send Data macro. Send SysVar Sample Teach Pendant Program shows a sample teach pendant program that uses the Send Data macro.

Table 10–77. Send SysVar Macro Description

MACRO ITEM	DESCRIPTION
Syntax:	Send SysVar(request_code, wait_sw, status_reg, "name1" ["name2" ["name3"]])

Table 10–77. Send SysVar Macro Description (Cont'd)

MACRO ITEM	DESCRIPTION
Parameters:	**request_code** - This item determines which event is triggered on the PC. It can be a value between 0 and 255.
	Wait_sw - This item is a wait switch. Valid values are either 0 or 1. **If 0,** do not wait for an answer form the PC. **If 1,** wait for an answer from the PC. The default value of **wait_sw** is 0.
	Status_reg - This item is a register which stores the value of either the answer or the error status for which the teach pendant program is waiting. Valid values are either 0 or a register number. **If 0,** no register is used, and no answer from the PC is reported back to the teach pendant program. **If wait_sw = 1** , then the teach pendant program waits for the PC to answer, but the answer value is not placed into a teach pendant register. If an error occurs in the macro, the teach pendant program is aborted. **If status_reg is not 0,** this is the register number used to report the PC's answer or error status to the calling teach pendant program. **If wait_sw = 0,** the teach pendant program will not wait for the PC to answer and the register will only be used to report errors. R[number] is set to 9999 if an error occurs.
	Name1 through Name3 - These items are STRING values which are used to create one system variable name whose contents are then sent to the PC. Each string can be up to 16 characters long; however, since some system variable names are longer than 16 characters, the optional parameters **Name2** and **Name3** can be supplied. Send SysVar appends them to Name1 to form a single system variable name.

Send SysVar Sample Teach Pendant Program

```
PROG TEST1
   1: ! Send Event 5 and value of system variable
   2: ! No answer from PC
   3: Send SysVar(5,0,0,'$FAST_CLOCK')
PROG TEST2
   1: ! Send Event 9 and value of system variable
   2: ! Wait for PC answer
   3: ! Answer in REG 3
   4: Send data (9,1,3,'$SCR_GRP[1]', '$FAST_CLOCK')
   5: IF R[3]<9999,JMP LBL[10]
   6: ! Error in macro
   7: !
   8: LBL[10]
```

10.39.5 Get Data( )

The Get Data macro is used to request that certain data (depending on the request code) be placed into the teach pendant register. Get Data always waits for the data to be received from the PC. This macro is implemented using the Send Data macro. It is possible to receive up to eight types of data in eight registers.

Note Only REAL and INTEGER data can be sent this way.

Table 10–78 lists and describes the parts of the Get Data macro. Get Data Sample Teach Pendant Program shows a sample teach pendant program that uses the Get Data macro. Refer to Section 10.39.3 for more information on the Send Data macro.

Table 10–78. Get Data Macro Description

MACRO ITEM	DESCRIPTION
Syntax:	Get Data(request_code, status_reg, reg1[, reg2[, reg3[, reg4[, reg5[, reg6[, reg7[, reg8]]]]]]])
Parameters:	**request_code** - This item determines which event is triggered on the PC. It can be a value between 0 and 255.
	Status_reg - This item is a register which is used to store either the answer or the error status from the PC. **If 0,** no register is used and no answer from the PC is reported to the calling teach pendant program. The teach pendant program **always** waits for the PC to answer, but the answer is not placed into a teach pendant register. If an error occurs, the teach pendant program is aborted. **If status_reg is not 0,** this is the teach pendant register that is used to report either the PC's answer or error status back to the teach pendant program. R[number] is set to 9999 if an error occurs. The default value of **status_reg** is 0.
	Reg1 through Reg8 - These items are registers which are used to accept data from the PC. The PC must be programmed to know what data is being requested. Valid items are: INTEGER and REAL. At least one data item must be requested.

Get Data Sample Teach Pendant Program

```
PROG TEST1
   1: ! Send Event 42
   2: ! Wait for Data put into R[1] and R[5]
   3: ! No other answer from PC
   4: Get Data(42,0,1,5)
PROG TEST2
   1: ! Send Event 42
   2: ! Wait for data put into R[1] and R[8]
   3: ! Other answer or error in R[10]
   4: Get Data(42,10,1,8)
```

```
   5: IF R[10]<9999,JMP LBL[10]
   6: ! Error in macro
   7: !
   8: LBL[10]
```

10.39.6 Request Menu( )

The Request Menu macro is used to send a request to place a menu onto the PC. The controller automatically waits until the event flag is set to a non-zero number before continuing. The PC will place the response into R[status_reg] so that the teach pendant program can use the value.

Table 10–79 lists and describes the Request Menu macro. Request Menu Sample Teach Pendant Program shows a sample teach pendant program that uses the Request Menu macro.

Table 10–79. Request Menu Macro Description

MACRO ITEM	DESCRIPTION
Syntax:	Request Menu(request_code, status_reg)
Parameters:	**request_code** - This item determines which event is triggered on the PC. It can be a value between 0 and 255.
	Status_reg - This item is a register which is used to store either the answer or the error status from the PC. **If 0,** no register is used and no answer from the PC is reported to the calling teach pendant program. The teach pendant program **always** waits for the PC to answer, but the answer is not placed into a teach pendant register. If an error occurs, the teach pendant program is aborted. **If status_reg is not 0,** this is the teach pendant register that is used to report either the PC's answer or error status back to the teach pendant program. R[number] is set to 9999 if an error occurs. The default value of **status_reg** is 0. This parameter is optional.

Request Menu Sample Teach Pendant Program

```
PROG TEST1
   1: ! Send Event 42
   2: ! The PC knows which menu we want
   3: ! Automatically wait for PC
   4: Request Menu(42)
PROG TEST2
   1: ! Send Event 42
   2: ! The PC knows which menu we want
   3: ! Wait for the operator to select
   4: ! something on the menu.  The Pc
```

```
 5: ! reports what was selected in R[6]
 6: Request Menu(43,6)
 7: IF R[6]<9999,JMP LBL[10]
 8: ! Error in macro
 9: !
10: LBL[10]
```

10.40 SAFETY SPACE

10.40.1 Overview

Safety Space allows you to specify strict limits for detecting robot collisions. You can determine the parts of the motion path where collisions are more likely to occur and increase the Collision Guard sensitivity to collisions for those particular areas. The Safety Space settings do not affect Collision Guard global settings.

10.40.2 Requirements

The Collision Guard option must be installed and active for the Safety Space function to work properly. Refer to the *Software Installation Manual* for more information.

Note Accurate specifications of the payload characteristics are required to maximize the accuracy of this function and to avoid false detections.

In the sample program shown in Sample Safety Space Program , the path defined by points P[1], P[2], and P[3] is considered the safety space.

Sample Safety Space Program

```
1:  J P[1] 100% FINE
2:  CALL SFSPC
3:  L P[2] 200mm/sec CNT100
4:  L P[3] 200m/sec FINE
5:  CALL SFSPCEND
6:  L P[4] 2000mm/sec FINE
```

10.40.3 Operation

Use Procedure 10-57 to call the Safety Space function.

In the SFSPC macro program, the lines that assign values to $SBR system variables should set the values to the default settings for the first run. After running the program, the actual values for maximum and minimum disturbance torque are obtained from $MOR_GRP system variables in Step 7 and Step 8 . When the actual disturbance is known, the SFSPC macro program can be changed according to the formulas in Step 7 and Step 8 .

Procedure 10-57 Calling the Safety Space Function

Conditions

- The Collision Guard option is installed and active. Refer to the *Software Installation Manual* for more information.

Steps

1. Insert the CALL SFSPC instruction where you want the safety space function to begin.
2. Insert the CALL SFSPCEND instruction where you want the safety space function to end.
3. Delete any unnecessary groups and joints within the SFSPC and SFSPCEND macro programs.

   Note Step 4 through Step 8 assume that you want to detect collisions that affect axis 1 the most. If you want to monitor axes in motion groups other than motion group 1, change the index in the system variable to the appropriate number. For example, to monitor axis 1 of the second group, refer to $SCR_GRP[2].$axisorder[1]. The value of this variable should be used as the index for $SBR. Assuming that $SCR_GRP[2].$axisorder[1] = 7, SFSPC can be changed to reference $SBR[7].$param[119].

4. Set the value of the system variable $SBR[1].$param[119] to a default value (for example, $SBR[1].$param[119] = 30000).
5. Set the value of the system variable $SBR[1].$param[120] to a default value (for example, $SBR[1].$param[120] = -30000).
6. Execute the Safety Space portion of the program to determine the values for SFSPC.
7. Replace the value 30000 with ($MOR_GRP[1].$MAX_DIS_TRQ[1] + 728)
8. Replace the value -30000 with ($MOR_GRP[1].$MIN_DIS_TRQ[1] - 728)

10.41 SERVO TOOL (Option)

Servo Tool is an option that simplifies and improves programming for servo grippers or other servo-driven axes.

This option provides a schedule-based user interface that allows the user to specify torque limits for specified auxiliary or robot axes. The Servo Tool TP instruction allows the user to call a particular servo tool schedule during TP program execution.

Also, this option provides high sensitive part detecting function that can be used in any section of regular program flow or in jogging conditions.



Warning

Enabling Servo Tool on robot axes could cause damage to equipment or personnel due to low motor torque and reduced error checking.

10.41.1 Benefits

ServoTool provides the following benefits:

- Gives the user a way to program the amount of torque needed without any slowdown, hesitation, or faulty position reporting.
- Automatically adjusts the error checking level to be suitable with the system when torque is low and the specified axis is not responsive.
- Automatically updates the program position to the contact position when Servo Tool is used. No extra programming logic is required.
- Provides a schedule-based user interface to allow users to assign torque limits for a specified auxiliary or robot axis.
- Provides a special TP instruction to call a particular servo tool schedule during TP program execution.
- Allows the robot to maintain the position where contact is made and increase actual torque to overcome the motion dynamics limitation without crushing the part.
- Provides high sensitive part detecting functions with various detecting methods including disturbance torque, torque, and position.

Warning

Regular error checking will be reduced or disabled when Servo Tool is enabled. Verify the path with extra caution.

10.41.2 Limitations

Servo Tool has the following limitations:

- When used in Single Step mode or when the program is paused, the ServoTool [ ] TP instruction will only activate the torque limit servo setting. No position update or offset will be applied. Longer execution time for motion instructions can be expected.
- When a program aborts, all torque limits will be reset to 100% value and error checking limits will be reset to their default values.
- As with other TP instructions, if the program resumes or skips to the motion instructions between **ServoTool []** and **ServoTool End**, no ServoTool schedule will be used.

Warning

Motor torque is low and regular error checking is disabled when a program is paused while Servo Tool is enabled.

10.41.3 Program Instruction

The SERVOTOOL[ ] and SERVOTOOL END TP instructions can be used in TP programs whenever needed.

SERVOTOOL[schedule number]

When the robot executes this TP instruction, the following occurs:

- A special set of servo parameters is updated in real time to make sure the specified torque limit is used.
- Error checking is open.
- Quick stop is used at the moment when contact is made (for example, at gripper contact with the part). This prevents hesitation that could affect cycle time.
- If Auto Position Update is enabled, the controller remembers the contact position for the specific axis, and applies it to the next moves until SERVOTOOL END is executed.

Torque limits and position offsets can be kept in the subsequent motion lines until SERVOTOOL END is reached.

SERVOTOOL END

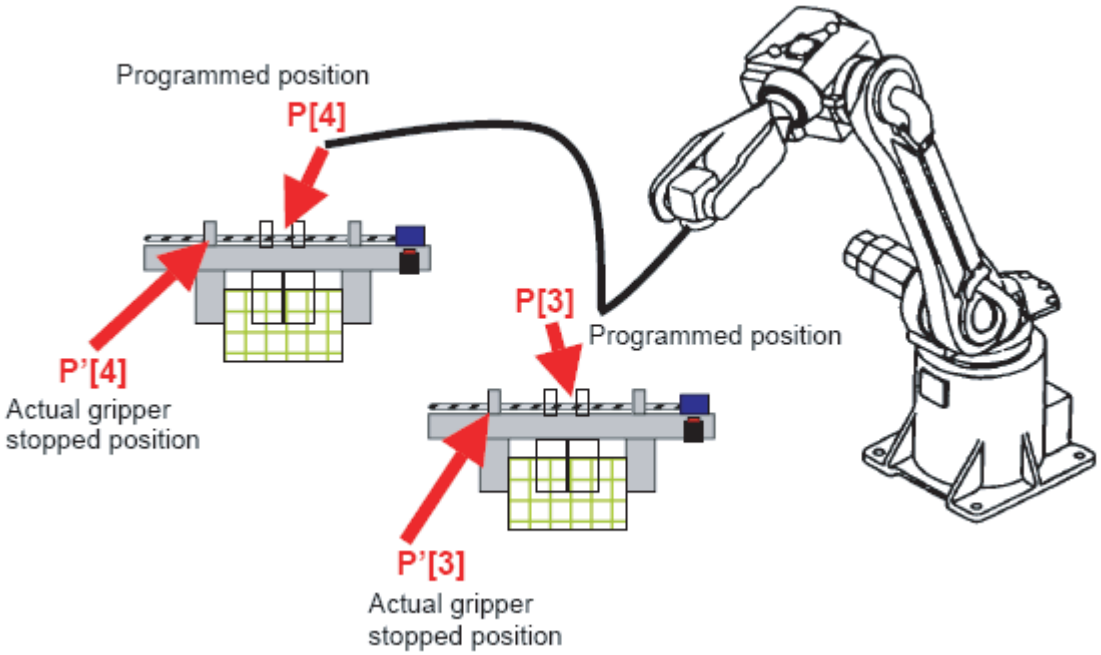
When the program reaches the SERVOTOOL END command, there is no delay or hesitation before the next motion. The following occurs:

- Torque limit is reset to its default value.
- Error checking is reset to normal.
- The internal position is updated.

Example Program Fragment

```
:
:
4: ServoTool [1]
5: L P[3] 1500mm/s FINE
6: J P[4] 100% CNT100
7: ServoTool End
```

Figure 10–140. Servo Tool



10.41.4 Setup

You can set up Servo Tool for high sensitive detecting function in many different conditions. $SV_DETECT[group].$DETECTED can be used to monitor detecting condition after $SV_DETECT[group].$TRIGGER has been set any number greater 0. For example, High Speed Skip standard TP instruction can be used to monitor $SV_DETECT[group].$DETECTED system variable to switch program flow when collision detected.

Table 10–80.

Item		Description
Setting menu:	Schedule No [ ]	This item identifies the schedule you will set up or modify.
	Group Number	This item identifies the group number of the axes to be controlled.
	Axis Number	This item identifies the axis number whose torque you want to specify.
	Torque Limit Range: 1% to 100% Default: 100%	This item is the percentage torque limit for the specified axis.
	Contact Sensitivity	This item identifies the duration that the servo axis remains idle in system ITP time. When the servo axis remains idle over this time period, the contact between the servo axis and the part will be determined, and the next action will be executed.
	Contact Tolerance	This item identifies the idle condition of the servo axis in encoder counts. For example, if this item is 2, then this servo axis is idle as long as the encoder reading for this servo axis maintains within 2 counts each time.
	Auto Position Update Default: YES	If set to Yes, position data of the specified axis between the SERVOTOOL[ ] and SERVOTOOL End instructions is updated automatically. You do not need to use an offset command or PR[ ] value when teaching the Servo Tool program.
Settings after contact: These items specify the amount of force to be exerted after the end effector contacts the workpiece.	Offset This item is not used if Auto Position Update is set to NO.	This item specifies the position offset at which contact is made. A positive value means the specific axis will be commanded to move an offset in the positive axis direction. This item can be used to control the tightness between end effector and part. If Auto Position Update is set to NO, this item will be disabled and the original taught point will be used.
	Torque Limit	This item specifies the new torque limit to be applied to the workpiece after contact is made. This value can be used to control the tightness between end effector and part.

10.41.5 Detection

You can set up Servo Tool for high sensitive detecting function in many different conditions.

$SV_DETECT[group].$DETECTED can be used to monitor detecting condition after $SV_DETECT[group].$TRIGGER has been set any number greater 0. For example, High Speed Skip standard TP instruction can be used to monitor $SV_DETECT[group].$DETECTED system variable to switch program flow when collision detected.

System Variables

All detecting required system variables are in group based $SV_DETECT system variable.

- $SV_DETECT[].$ENABLE[]: These axis based variables are used to determine which axes will be used to detect.
- $SV_DETECT[].$TOL[]: These axis based variables are used to determine the detecting tolerance for each enabled axis. The default value is 5% of max torque (if torque is used to detect) or standard disturbance torque margin value (if disturbance torque is used to detect).
- $SV_DETECT[].$DET_TYPE: This variable determines the detecting type. 0 is disturbance torque (default) and 1 is motor torque.
- $SV_DETECT[].$ALGO: This variable determines the detecting algorithm.0: the upper and lower limits in $SV_DETECT[].$DET_LIMIT[1].$UPLIM[axis] and in $SV_DETECT[].$DET_LIMIT[1].$LOWLIM[axis] will be used to determine detection occurred or not. 1: only positive value of $SV_DETECT[].$DET_TYPE will be used to set detection flag. 2:: only negative value of $SV_DETECT[].$DET_TYPE will be used to set detection flag. 4: absolute value of $SV_DETECT[].$DET_TYPE over $SV_DETECT[].$DET_LIMIT[1].$UPLIM[axis] will be used to set detection flag.
- $SV_DETECT[].$DET_LIMIT[1].UPLIM[]: These axis based variables are upper limits for detection when $SV_DETECT[].$ALGO is 0 or 4.
- $SV_DETECT[].$DET_LIMIT[1].LOWLIM[]: These axis based variables are lower limits for detection when $SV_DETECT[].$ALGO is 0.
- $SV_DETECT[].$LEARN: This variable is used for automatically learning the upper and lower limits when $SV_DETECT[].$ALGO is 0. Please set $SV_DETECT[].$DET_LIMIT[1].UPLIM[] to large negative values, for example, -10000 and $SV_DETECT[].$DET_LIMIT[1].LOWLIM[] to large positive values, for example, 10000, to make this learning works.
- $SV_DETECT[].$TRIGGER: This variable is used for start detecting or learning (if $LEARN is 1). To bypass the region which easy to cause mis-detection, this variable can be set to any value greater than 1 to start counting down. The recommended value is 20.
- $SV_DETECT[].$DETECTED: This variable is used for start detecting or learning (if $LEARN is 1). To bypass the region which easy to cause mis-detection, this variable can be set to any value greater than 1 to start counting down. The recommended value is 20.This variable will be set when trigger starts (also $LEARN is 0) and the detection criterion met.

Program Example (with high speed skip)

For example, the original program is:

```
1: J P[1] 100% FINE
```

```
2: L P[2] 2000mm/s CNT0
3: L P[3] 800mm/s CNT100
4: L P[4] 2000mm/s CNT100
5: J P[5] 100% FINE
```

If extra protection is required for line 3. $SV_DETECT[].$DETECTED can be used combined with High Speed Skip

```
1: J P[1] 100% FINE
2: SKIP CONDITION $SV_DETECT[1].$DETECTED = 1
2: L P[2] 2000mm/s CNT0
3: $SV_DETECT[1].$TRIGGER = 20
3: L P[3] 800mm/s CNT100, SKIP LBL[1], PR[1] = LPOS
4: L P[6] 2000mm/s CNT100
5: LBL[1]
6: $SV_DETECT[1].$TRIGGER = 0
7: L P[4] 2000mm/s CNT100
8: J P[5] 100% FINE
```

Operation

After modifying the program, turn on $SV_DETECT[].$ENABLE[] for the desired axes. Set $SV_DETECT[].$DET_LIMIT[1].UPLIM[] to large negative values and $SV_DETECT[].$DET_LIMIT[1].LOWLIM[] to large positive values. Then executing the modified program with $SV_DETECT[].$LEARN = 1. Please be careful to only set $SV_DETECT[].$LEARN = 1 when the program is ready to be played back completely from start to end and no more program logic or position changes are required. Verify that $SV_DETECT[].$DET_LIMIT[1].UPLIM[] and $SV_DETECT[].$DET_LIMIT[1].LOWLIM[] have been set automatically with proper values. Then turn off $SV_DETECT[].$LEARN and start normal program execution. If mis-detection happens, $SV_DETECT[].$TOL[] can be used to fine tune sensitivity.

10.42 SHAPE GENERATION II (OPTION)

10.42.1 Overview

The Shape Generation II option simplifies cutting two dimensional shapes. The software also reduces online programming time by reducing the number of required taught robot positions and providing features such as shape cut interface.

The following shapes can be cut using the Shape Generation II option:

- Circles

- Hexagons
- Rectangles
- Slots
- Key slots
- Pommels
- Custom

To use the Shape Generation II option, you must do the following:

1. Set up default shape information - Section 10.42.3
2. Generate shape programs for the shapes you want to cut - Section 10.42.7
3. Include process instructions within the generated shape program - Procedure 10-61



Warning

Make sure your program stops the process (such as laser or water jet cutting) when an error occurs. The Shape Generation II option does not automatically turn off the process equipment when an error occurs. For example, if you are cutting shapes using a laser or water jet, you must write the program so that the process shuts off if an error occurs. If you do not, you could injure personnel or damage equipment.

4. Create a production program that calls the generated shape programs - Step 9

10.42.2 Quick Start Hints

Setup

Under SETUP, you will find “Shapes”. This is where you can change the default data for “Step to center” as well as the data for each shape. This is the default data and all items can be changed at the time of shape creation.

Note “Step to center” is programmable, so setting it here is not critical. Setting or clearing FLAG 101 in the main or calling program, will dictate this behavior when running a shape program.

Shape Gen

Shape Generation II creates complete independent programs; it does not create these shapes “on the fly” like the original Shape Generation. All shapes must be completed before any “production” start is initiated.

Create or Edit

Under UTILITIES, you will find" Shape Gen". This is where you will choose which shape to work with. Most of these parameters are taken from the original Shape Generation Feature, so they will function the same way.

Edit

If you have created a program with this utility already, simply go to the type of shape created and press DETAIL; then from the second item (Program Name) press F4 (CHOICE) to select the program to modify

10.42.3 Shape Setup

You can set up detailed information before you use the Shape Generation II option. You will specify this information in the SETUP Shapes DETAIL screens. **The information you specify in these screens will become the default information that applies to all shapes you generate using the Shape Generation II option**.

You can set up data to define how you want to generate the following shapes:

- Circle
- Hexagon
- Rectangle
- Slot
- Keyhole
- Pommel
- Custom

You can display and set detailed default information for each shape from the shape-specific DETAIL Setup screens. You will be able to adjust the data for each individual shape from the UTILITIES Shape Gen screen beyond what you see here.

Table 10–81 lists and describes the items in the SETUP Shapes screen. Use Procedure 10-58 to set up shape information.

Table 10–81. Shape Setup Information

ITEM	DESCRIPTION
Step to shape center	This item defines whether the robot steps to the shape center (Enabled) or steps directly to the pierce position (Disabled).
First Move Speed	This indicates the move speed of the robot to the first position in the program.

Table 10–81. Shape Setup Information (Cont'd)

Process Copy/Paste	**Note** AVC must be associated with motion. If you are using AVC, the normal CALL PROCESS_ONx.TP and CALL PROCESS_OFx.TP can not be used from the main shape TP program. The information contained in the PROCESS_ONx.TP and PROCESS_OFx.TP programs must be moved into the shape program for this to work correctly. When this is enabled, the information contained in the PROCESS_ONx.TP and PROCESS_OFx.TP programs is copied to the shape program. When it is disabled, the CALL to the process programs is used.	
Upward Z+ for Utool	This is used when the standard ARC Tool Utool defines the tool Z direction to go into the TCP, instead of away from the TCP.	
On Path Term Type	This is used to control the term type used to get onto the initial part of the shape, where the choice will be CR or CNT. If CNT is specified, the "On Path CNT value" will be used. If CR is specified, "On Path CNT value" is ignored.	
On Path CNT value	This value is used if the On Path Term Type CNT is specified. If CR is specified, this value is ignored.	
Off Path CNT VALUE	This is used to adjust the CNT value as the motion leaves the shape path.	
Write Protection	This item automatically sets the shape to be write protected when the shape is created. It is set to TRUE by default.	
Shape data	**Circle**	Selecting this item allows you to set up default shape information for Circles.
	Slot	Selecting this item allows you to set up default shape information for Slots.
	Hexagon	Selecting this item allows you to set up default shape information for Hexagons.
	Rectangle	Selecting this item allows you to set up default shape information for Rectangles.
	Keyhole	Selecting this item allows you to set up default shape information for Keyholes.
	Pommel	Selecting this item allows you to set up default shape information for Pommel shapes.
	Custom	Selecting this item allows you to set up default information for custom shapes.

Procedure 10-58 Setting Up Default Shape Information

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Shapes. You will see a screen similar to the following.

```
SETUP Shapes
 1 Step to shape center:        ENABLE
 2 First Move Speed:            2000 mm/sec
 3 Process Copy/Paste:          ENABLE
 4 Upward Z+ for Utool:         ENABLE
 5 On Path Term Type:           CR
 6 On Path CNT value:            100
 7 Off Path CNT VALUE:             10
 8 Write Protection:            TRUE
   Shape data:
 9  Circle:
10  Slot:
11  Hexagon:
12  Rectangle
13  Keyhole:
14  Pommel:
15  Custom:
```

5. Select the kind of shape you want (Circle, Hexagon, Rectangle, Slot, Keyhole, Pommel or Custom) and press DETAIL. Refer to Section 10.42.5 and Section 10.42.6 for detailed information for standard and custom shapes.
6. Set the items as desired.
7. **To save the data** you entered, press SAVE.
8. **To display help information**, press F5, HELP.
9. **To display the previous screen**, press PREV.
10. **To toggle the speed units**, press F3, UNITS. The speed units will change between mm/sec to cm/min to in/min.

10.42.4 Setting Up Default Shape Data

You can set up data to define how you want to generate the following shapes:

- Circle
- Hexagon
- Rectangle
- Slot
- Keyhole
- Pommel
- Custom

Using the SETUP Shape DETAIL Screens

You can display and set detailed default information for each shape from the shape-specific DETAIL Setup screens. You will be able to adjust the data for each individual shape from the UTILITIES Shape Gen screen beyond what you see here.

All the items in Table 10–82 appear in both the SETUP Shape screen and the UTILITIES Shape Gen screen, except where noted. These items apply to *all* shapes. Additional items specific to each shape type are provided with each shape discussion in Section 10.42.5 .

Table 10–82. SETUP DETAIL Screen and UTILITIES DETAIL Screen Common Items

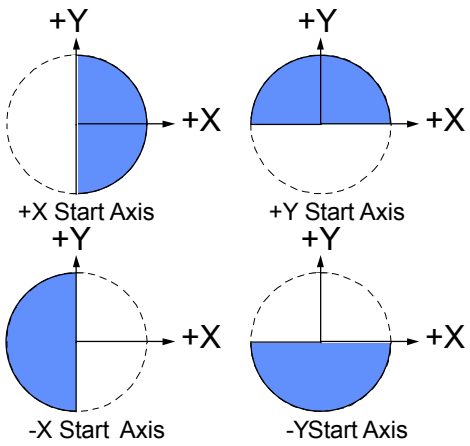
ITEM	DESCRIPTION
Start Axis(+/-X or +/-Y)	This item is defined as the axis on the shape at which that the robot will begin its cut. Each of the four (4) start axis choices are shown. Note: Half circle orientation depends on which start axis is implemented. 

Table 10–82. SETUP DETAIL Screen and UTILITIES DETAIL Screen Common Items (Cont'd)

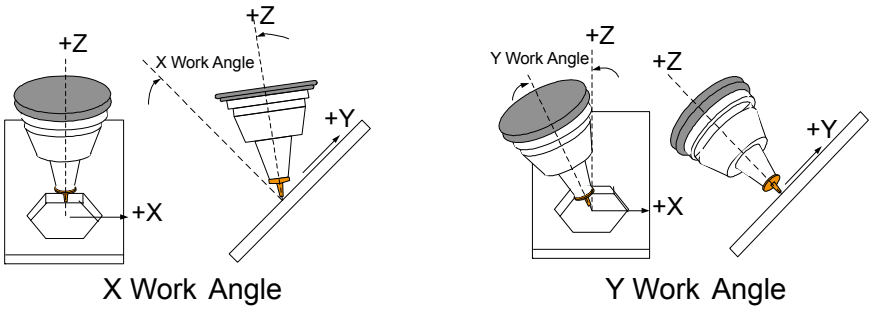
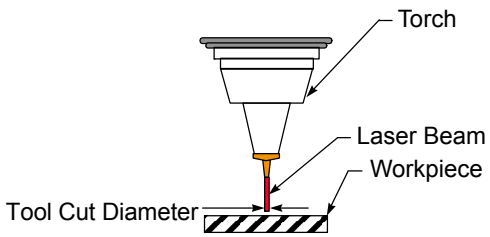
X-Work angle Range: +/- 45° Y-Work angle Range: +/- 45°	The X and Y Work angles are provided to allow you to program beveled cuts. The angles are measured from the normal around the corresponding shape axis (X or Y).  X Work Angle Y Work Angle
Blend-in angle Range: ± 45°	This item is the angle from the start axis to the start position.
Blend-in distance Range: ± 20 mm	This item is the distance from the shape edge to start position.
Overlap angle Range: ± 45°	This item is the angle between the start axis and end position.
Blend-off angle Range: ± 45°	This item is the angle from the start axis to the end position inside the shape.
Blend-off dist Range: ± 20mm	This item is the distance from the shape edge to the end position.
Num of circ points This item appears only in Circle, Slot, Keyhole, Rectangle, and Pommel shapes	This item is the number of points you choose to define around the circle. For rectangle shapes, normally this is set to 0 and CR is used with a single point in the corner. If this is a non 0 value, the corner will be turned into circular motion and the number indicates how many points will be used in each corner.
Cut direction	This item is the direction the TCP travels around the shape.
Rotation	This item allows the shape to rotate around the TCP.
Tool cut diameter	
Plunge distance	This item is the distance the tool travels down into the workpiece.

Table 10–82. SETUP DETAIL Screen and UTILITIES DETAIL Screen Common Items (Cont'd)

Plunge speed	This item is the speed of the tool as it travels down into the workpiece.
Retract distance	This item is the distance the tool travels up out of the workpiece when the cut is completed.
Retract speed	This item is the speed of the tool as it travels up out of the workpiece when the cut is completed.
Speed units	This item is the speed measurement in mm/sec, cm/min, or in/min.

Procedure 10-59 describes how to use the SETUP Shape and DETAIL screens.

Procedure 10-59 Using the SETUP Shape and DETAIL Screens

Steps

1. Press MENU.
2. Select SETUP.
3. Press F1, [TYPE] and select Shapes.
4. Select the kind of shape you want (Circle, Hexagon, Rectangle, Slot, Key Hole, Pommel, or Custom) and press DETAIL. Refer to Section 10.42.5 and Section 10.42.6 for detailed information on standard and custom shapes.
5. Set the items as desired.
6. **To save the data** you entered, press SAVE.
7. **To display help information**, press F5, HELP.
8. **To display the previous screen**, press PREV.
9. **To toggle the speed units**, press F3, UNITS. The speed units will change from mm/sec to cm/min to in/min.

10.42.5 Shape-Specific Shape Data for Standard Shapes

Use Procedure 10-58 to set up data for the shape you want.

Refer to Table 10–82 for information on items common to all shapes.

This section provides detailed shape-specific information about default data for the following standard shapes:

- Circles
- Hexagons

- Rectangles
- Slots
- Keyholes
- Pommels

Circle Detail

Figure 10–141 shows the SETUP Circle DETAIL screen. The circle-specific items on this screen are listed and described in Table 10–83 .

Refer to Table 10–82 for information on items common to all shapes.

Figure 10–141. SETUP Circle DETAIL Screen

```
SETUP Circle
  1  Diameter:            55.00 mm
  2  Circular Speed:     390 cm/min
  3  Linear Speed:       390 cm/min
  4  Circle type:         FULL
  5  Edge distance:       0.00 mm
  6  Start axis:          +X
  7  X Work angle         0.00 deg
  8  Y Work angle:        0.00 deg
  9  Blend-in angle:      20.00 deg
 10  Blend-in distance     3.00 mm
 11  Overlap angle:       10.00 deg
 12  Blend-off angle:     —45.0 deg
 13  Blend-off dist:      20.00 mm
 14  Num of circ points:      2
 15  Cut direction:        CW
 16  Rotation:            0.00 deg
 17  Tool cut diameter:   0.00 mm
 18  Plunge distance:     0.00 mm
 19  Plunge speed:        390 cm/min
 20  Retract distance:    12.00 mm
 21  Retract speed:       390 cm/min
```

Table 10–83. Circle DETAIL Items (Circle-Specific Items)

ITEM	DESCRIPTION
Diameter Range: 1 - 500 mm	This item is the diameter in mm of the current circle/half circle.
Circular Speed Range: 1 - 9999 cm/sec	This item is the cutting speed of the circle/half circle.

Table 10–83. Circle DETAIL Items (Circle-Specific Items) (Cont'd)

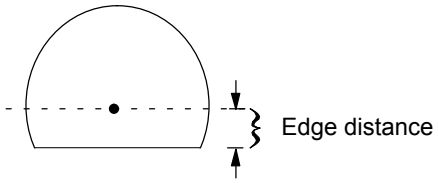
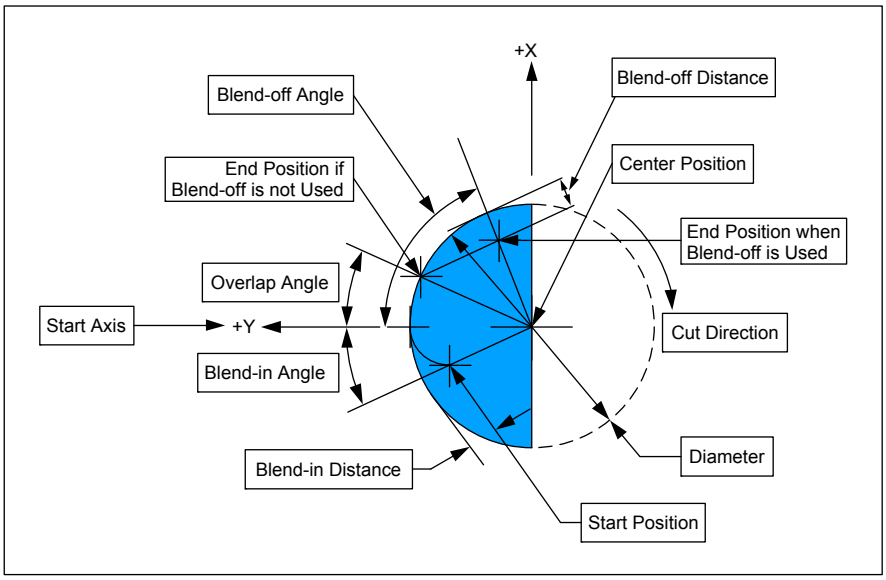
Linear Speed Range: 1 - 9999 cm/sec	This item is the cutting speed of the circle/half circle.
Circle Type Full/D_Shape	This item specifies whether the circle is a full circle or D-shape circle.
Edge Distance Range: -250 - 250 mm	This item specifies the distance from the circle center to where the D-shape edge will be. Edge distance is used only when the circle Type is set to D_Shape. 

Figure 10–142. Circle Detail Terminology



Hexagon Detail Data

Figure 10–143 shows the SETUP Hexagon DETAIL screen. The items on these screens are listed and described in Table 10–84 .

Refer to Table 10–82 for information on items common to all shapes.

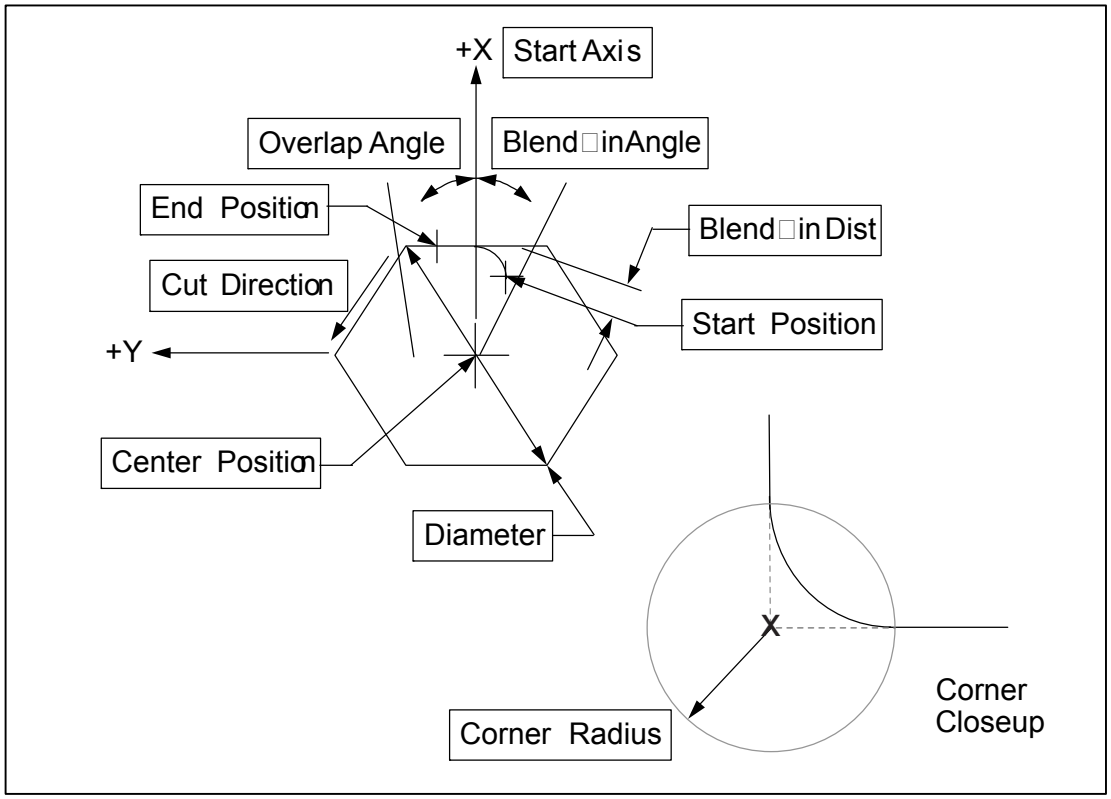
Figure 10–143. SETUP Hexagon DETAIL Screen

```
SETUP Hexagon
                              1/18
  1  Diameter:            15.00 mm
  2  Corner Radii:        2.00 mm
  3  Speed:                65  mm/sec
  4  Start axis:           +X
  5  X Work angle:         5.00 deg
  6  Y Work angle:        15.00 deg
  7  Blend-in angle:      20.00 deg
  8  Blend-in distance:    5.00 mm
  0  Overlap angle:       20.00 deg
 10  Blend-off angle:      0.00 deg
 11  Blend-off dist:       0.00 deg
 12  Cut direction:        CW
 13  Rotation:             0.00 deg
 14  Tool cut diameter:    0.00 mm
 15  Plunge distance:      0.00 mm
 16  Plunge speed:         65 mm/sec
 17  Retract distance:    12.00 mm
 18  Retract speed:        65 mm/sec
```

Table 10–84. Hexagon DETAIL Items (Hexagon-Specific Items)

ITEM	DESCRIPTION
Diameter Range: (1 - 500 mm)	This item is the diameter in mm across the corners of the current hexagon.
Corner Radii	This item specifies the radius of the corners (all corners are identical).
Speed Range: (1 - 9999 cm/sec)	This item is the cutting speed of the current hexagon.

Figure 10–144. Hexagon Detail Terminology



SETUP Rectangle DETAIL Data

Figure 10–145 shows the SETUP Rectangle DETAIL screen. The items specific to rectangles are listed and described in Table 10–85 .

Refer to Table 10–82 for information on items common to all shapes.

Figure 10–145. SETUP Rectangle DETAIL Screen

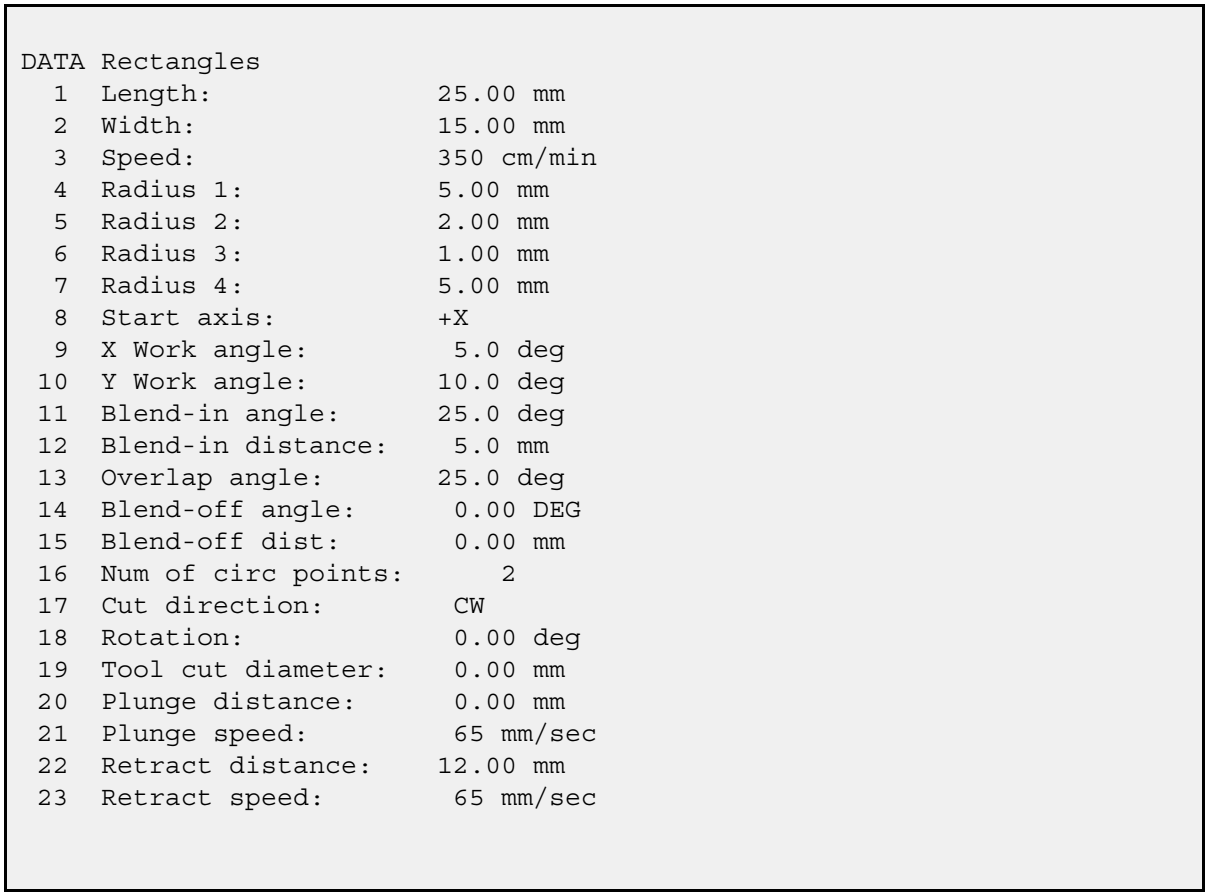
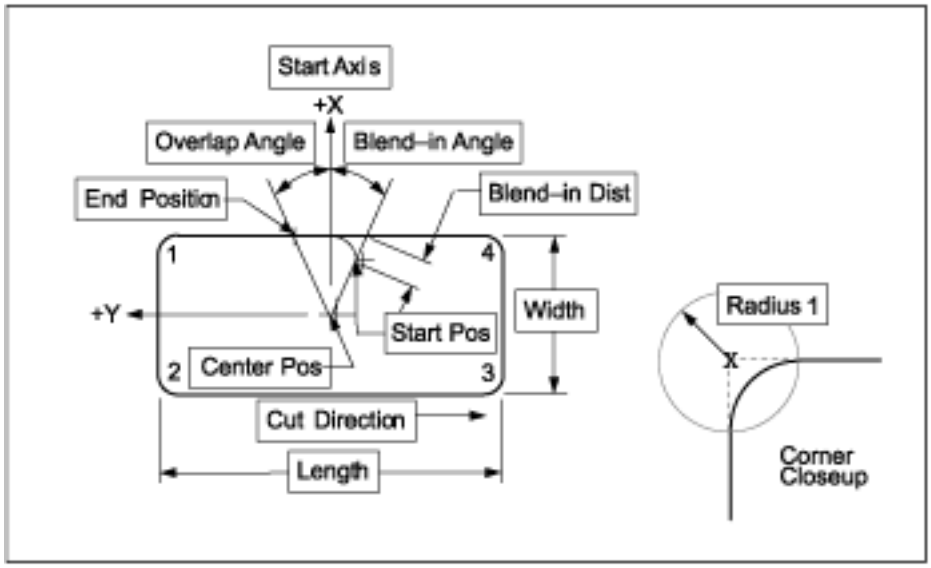


Table 10–85. Rectangle DETAIL Items (Rectangle-Specific Items)

ITEM	DESCRIPTION
Length Range: (1 - 500 mm)	This item is the length in mm across the longer side of the current rectangle.
Width Range: (1 - 500 mm)	This item is the width in mm across the shorter side of the current rectangle.
Speed Range: (1 - 9999 cm/sec)	This item is the cutting speed of the current rectangle.
Radius 1	This item specifies the radius of the first corner encountered in the rectangle depending on the cut direction.
Radius 2	This item specifies the radius of corner 2.
Radius 3	This item specifies the radius of corner 3.
Radius 4	This item specifies the radius of corner 4.

Figure 10–146. Rectangle Detail Terminology



SETUP Slot DETAIL Items

Figure 10–147 shows the SETUP Slot DETAIL screen. The items on these screens are listed and described in Table 10–86 .

Refer to Table 10–82 for information on items common to all shapes.

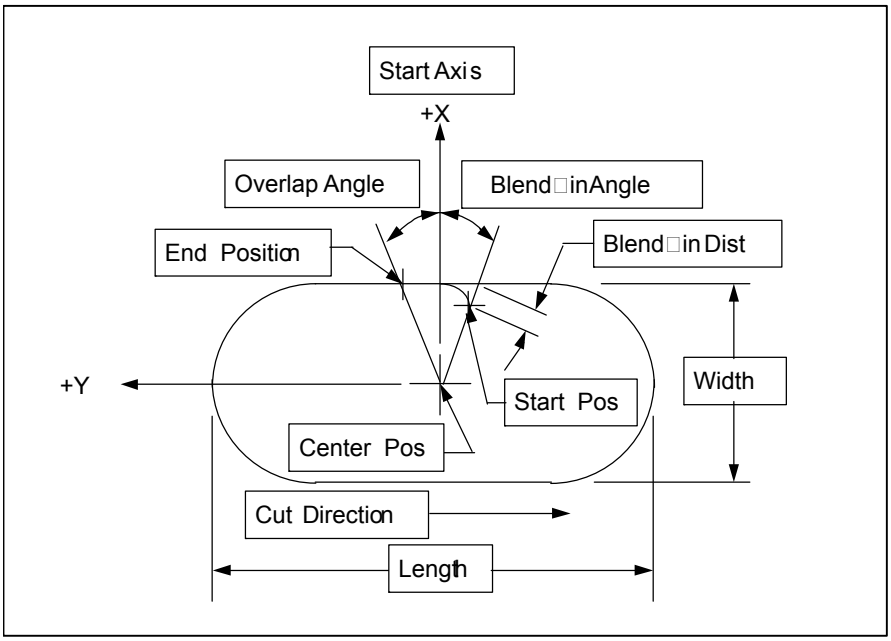
Figure 10–147. SETUP Slot DETAIL Screen

```
DATA Slot
  1  Schedule #: 1        [Slot 1         ]
  2  Length:              15.00 mm
  3  Width:               10.00 mm
  4  Speed:               350 cm/min
  5  Start axis:          +X
  6  X Work angle:         5.0 deg
  7  Y Work angle:        10.0 deg
  8  Blend-in angle:      25.0 deg
  9  Blend-in distance:     5.0 mm
 10  Overlap angle:       25.0 deg
 14  Blend-off angle:      0.00 DEG
 15  Blend-off dist:       0.00 mm
 13  Num of circ points:      2
 16  Cut direction:        CW
 17  Rotation:             0.00 deg
 18  Tool cut diameter:    0.00 mm
 19  Plunge distance:      0.00 mm
 20  Plunge speed:         65 mm/sec
 21  Retract distance:    12.00 mm
 22  Retract speed:        65 mm/sec
```

Table 10–86. Slot DETAIL Items (Slot-Specific Items)

ITEM	DESCRIPTION
Length Range: (1 - 500 mm)	This item is the length in mm across the longer side of the current slot.
Width Range: (1 - 500 mm)	This item is the width in mm across the shorter side of the current slot.
Speed Range: (1 - 9999 cm/sec)	This item is the cutting speed of the current slot.

Figure 10–148. Slot Detail Terminology



SETUP Keyhole DETAIL Items

Figure 10–149 shows the key hole schedule DETAIL screen. The items on these screens are listed and described in Table 10–87 .

Refer to Table 10–82 for information on items common to all shapes.

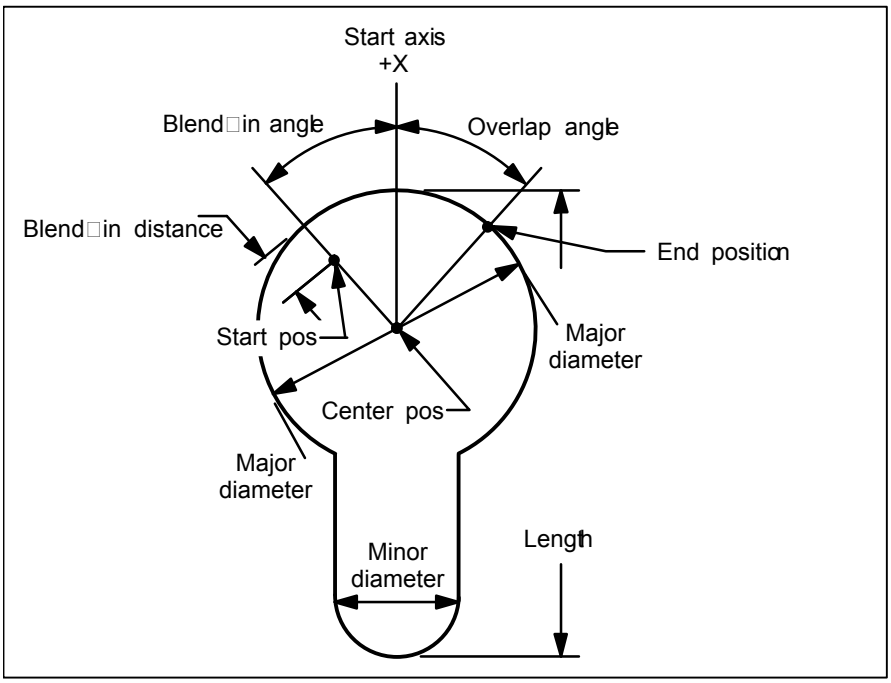
Figure 10–149. SETUP Keyhole DETAIL Screen

```
SETUP Keyhole
  1  Major Diameter:     100.00 mm
  2  Minor Diameter:      50.00 mm
  3  Length:             250.00 mm
  4  Circular Speed:      65 mm/sec
  5  Linear Speed:        65 mm/sec
  6  Start axis:          +X
  7  X Work angle:         5.0 deg
  8  Y Work angle:        10.0 deg
  9  Blend-in angle:      25.0 deg
 10  Blend-in distance:    5.0 mm
 11  Overlap angle:       25.0 deg
 14  Blend-off angle:      0.00 DEG
 15  Blend-off dist:       0.00 mm
 14  Num of circ points:      2
 16  Cut direction:        CW
 17  Rotation:             0.00 deg
 18  Tool cut diameter:    0.00 mm
 19  Plunge distance:      0.00 mm
 20  Plunge speed:         65 mm/sec
 21  Retract distance:    12.00 mm
 22  Retract speed:        65 mm/sec
```

Table 10–87. Keyhole DETAIL Items (Keyhole-Specific Items)

ITEM	DESCRIPTION
Major Diameter Range: (0 - 500 mm)	This item specifies the diameter of the large circle.
Minor Diameter Range: (0 - 500 mm)	This item specifies the diameter of the small radius, which is also the width of the slot.
Length Range: (0 - 500 mm)	This item specifies the overall length of the keyhole.
Circular Speed Range: (1 - 9999 cm/sec)	This item is the cutting speed of the current keyhole.
Linear Speed Range: (1 - 9999 cm/sec)	This item specifies the weld speed. You can also specify the speed units: mm/sec, cm/min, and in/min. NOTE: In the CUT_TML.TP program, the mm/sec speed units are always used, after converting the specified units to mm/sec.

Figure 10–150. Keyhole Detail Terminology



SETUP Pommel DETAIL Items

Figure 10–151 shows the pommel schedule DETAIL screen. The pommel-specific items on this screen are listed and described in Table 10–88 .

Refer to Table 10–82 for information on items common to all shapes.

Figure 10–151. SETUP Pommel DETAIL Screen

```
SETUP Pommel
 1 Diameter:              0.00 mm
 2 Object diameter:       0.00 mm
 3 Tool-Face toward:      Center
 4 Circular Speed:      65 mm/sec
 5 Linear Speed:        65 mm/sec
 6 Start Axis           X
 7 X Work Angle         0.000 deg
 8 Y Work Angle         0.000 deg
 9  Blend-in angle:      25.0 deg
10  Blend-in distance:    5.0 mm
11  Overlap angle:       25.0 deg
12  Blend-off angle:      0.00 DEG
13  Blend-off dist:       0.00 mm
14  Num of circ points:       2
15  Cut direction:         CW
16  Rotation:             0.00 deg
17  Tool cut diameter:    0.00 mm
18  Plunge distance:      0.00 mm
19  Plunge speed:         65 mm/sec
20  Retract distance:    12.00 mm
21  Retract speed:        65 mm/sec
```

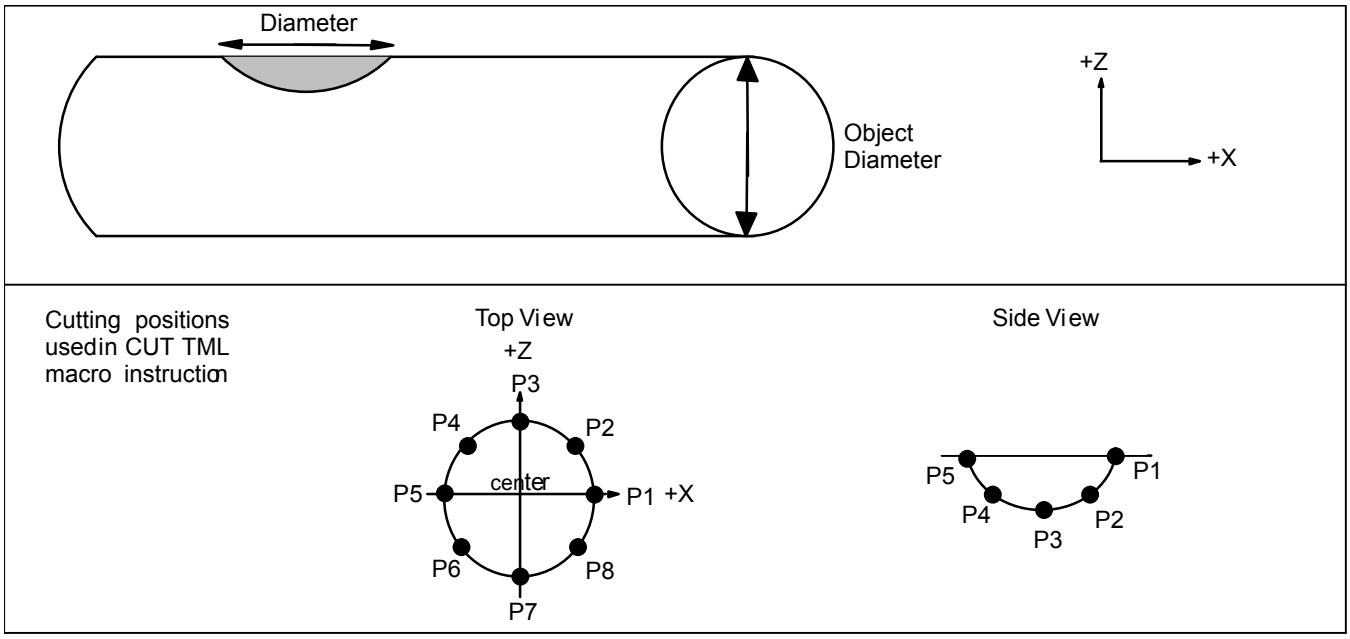
Table 10–88. Pommel DETAIL Items (Pommel-Specific Items)

ITEM	DESCRIPTION
Diameter Min.: 0.00 mm Default: 0.00 mm Max: 500.00 mm	This item specifies the diameter of the circle to process on the pipe.
Object Diameter Min.: 0.00 mm Default: 0.00 mm Max: 500.00 mm	This item specifies the diameter of the pipe. This must be larger than the diameter of the circle.

Table 10–88. Pommel DETAIL Items (Pommel-Specific Items) (Cont'd)

Tool-Face Default: Parallel	This item specifies the direction the face of the tool will point during rotation: • **Parallel** keeps the face of the tool parallel to the pipe's x-y plane. • **Center** directs the face of the tool to the origin of the pipe's x-y plane. Side View Parallel Tool Face Center Tool Face 
Circular Speed Min: 0 Default: 390 cm/min, 154 in/min, 65 mm/sec Max: 9999 cm/min, 9999 in/min, 9999 mm/sec	This item specifies the weld speed. You can also specify the speed units: mm/sec, cm/min, and in/min. **NOTE:** In the CUT_TML.TP program, the mm/sec speed units are always used, after converting the specified units to mm/sec.
Linear Speed Min: 0 Default: 390 cm/min, 154 in/min, 65 mm/sec Max: 9999 cm/min, 9999 in/min, 9999 mm/sec	This item specifies the weld speed. You can also specify the speed units: mm/sec, cm/min, and in/min. **NOTE:** In the CUT_TML.TP program, the mm/sec speed units are always used, after converting the specified units to mm/sec.
Circle type Default: FULL	This item specifies the kind of circle: • **FULL** is a full circle. **D_SHAPE** is a half circle. When you select D_SHAPE, 1 is substituted for R[7].

Figure 10–152. Pommel Detail Terminology



10.42.6 Setting Up Custom Shape Data

To create a custom shape you must perform the following steps:

1. Define the path.
2. Create the custom shape template program.
3. Set up custom shape detail data.

Figure 10–153 shows the SETUP Cust Shape DETAIL screen. The items on this screen are listed and described in Table 10–89 .

Figure 10–153. SETUP Cust Shape DETAIL Screen

```
SETUP Cust Shape
1 Program name:   TRIANG60
2 X Work angle:   0.00 deg
3 Y Work angle:   0.00 deg
4 Rotation:       0.00 deg
```

Table 10–89. Cust Shape DETAIL Screen Items

ITEM	DESCRIPTION
Program Name	This item requires you to press F4, [CHOICE], to select the program to be used for the custom shape.
Template Name This item appears only on the UTILITIES Cust Shape screen.	
X Work Angle	This item provides a bias angle relative to the X axis for a bevel cut.
Y Work Angle	This item provides a bias angle relative to the Y axis for a bevel cut.
Rotation	This item allows the shape to rotate about the TCP.

Limitations

The following limitation applies to Custom shapes.

Table 10–90. Limitations for Custom Shapes

ITEM	DESCRIPTION
Tool Cut Diameter	This item does not apply to Custom Shapes.

Procedure 10-60 Setting Up a Custom Shape

Steps

1. **Define the path:** Create a template teach pendant program that will define a special shape with the x, y and z coordinates of the shape to be cut from a reference point. The data must include the reference point, Cut Start Point (with lead in if desired), End Point (with overlap or runoff if required) and all intermediate points.

   The custom shape template program is flexible and can have any TP instruction that is needed for the shape. These templates will be duplicated and the points updated for the TCP location at the time of shape creation. The following screen shows an example template program.

```
TRIANG60.TP
   1:  !Template for 60deg triangle ;
   2:L P[1] 2000mm/sec FINE    ;
   3:L P[2] 100mm/sec CNT100    ;
   4:  CALL PROC_ON1    ;
   5:L P[3] 100mm/sec CNT100    ;
   6:L P[4] 100mm/sec CNT100    ;
   7:  DO[1]=OFF ;
   8:L P[5] 100mm/sec CNT100    ;
   9:  WAIT    .50(sec) ;
```

2. **Create the custom shape template program**.

> **Caution**
>
> You must take precautions when selecting the TERM type. Select FINE, CNTXXX or CR. Otherwise, the shapes will not be accurate.

Note Only L (Linear) and C (Circular) type motions can be used. Use only x, y and z for entering point data. W, P, and R are always 0.

Note You can add as many points as needed.

> **Warning**
>
> **You must not perform SHIFT TOUCHUP on any of the points because it is not intended to be executed as a stand-alone teach pendant program.**

3. **Set up Custom Shape Detail data :** After you create the teach pendant program, set up the Custom Shape detail data.
   a. Press MENU.
   b. Select SETUP.
   c. Press F1, TYPE and select Shapes.
   d. Select Custom and press DETAIL. You will see a screen similar to the following.

```
DATA Cust Shape
1 Program name:   TRIANG60
3 X Work angle:   0.00 deg
4 Y Work angle:   0.00 deg
4 Rotation:       0.00 deg
```

e. Set custom shape details as desired. Refer to Table 10–89 for information on custom shape details.

10.42.7 Generating a Shape Program from UTILITIES Shape Gen

Using Shape Generation II requires a minimum of two teach pendant programs:

- **Generated Shape Programs.** One or more TP programs *created by the Shape Generation II option*, using information you enter in the UTILITIES Shape Gen DETAIL screens. These programs define the paths to generate the shapes; they automatically include process-on and process-off instructions but they do not include additional motion statements.
- **Production Program.** A production TP program *that you create* to call the generated programs. This includes all other logic and motion statements required by your application.



Warning

In your production program, you must be aware of the calling sequence of the shape programs for the proper execution of your application program. Otherwise, you could injure personnel or damage equipment.

Make sure the production program stops the process (such as laser or water jet cutting) when an error occurs. The Shape Generation II option does not automatically turn the process equipment off when an error occurs. For example, if you are cutting shapes using a laser or water jet, you must write the production program so that the process shuts off if an error occurs. If you do not, you could injure personnel or damage equipment.

Generated Shape Programs

Instructions and information you provide are used by the Shape Generation II feature to generate the shapes you specify.

You will use the UTILITIES Shape Gen DETAIL screens to enter data for the particular shape you are defining; this data will override the default data you previously set up in the SETUP Shapes DETAIL screens.

In addition to the data you enter in the UTILITIES Shape Gen DETAIL screens, the system will use the following information to generate a shape program:

- Currently selected jog GROUP (for multi group systems)
- Current TOOL frame of the selected group
- Current USER frame of the selected group
- The location of the TCP of the GROUP that is selected. This will be the center point.
- PROC_ONx (where x is the GROUP number)
- PROC_OFx (where x is the GROUP number)

Note The generated shape programs will automatically include a process-on and a process-off CALL statement during the creation process. If these programs do not already exist, they will be created automatically. Once they are created, they will not be touched again by the system and you can add instructions for the process requirements.

UTILITIES Shape Gen DETAIL Information

The UTILITIES Shape Gen DETAIL screens include the same information as the SETUP Shape DETAILS screens, with the addition of three items.

Changing the data on these screens will affect *only the current shape* you are working on. When you press CREATE, the shape will be created with that data and you can immediately start using that program.

- Table 10–91 provides information about the items that appear only on the UTILITIES Shape Gen DETAIL screens.
- Refer to Table 10–82 for information about the items on the UTILITIES Shape Gen DETAILS screen that are common to all shapes.

Table 10–91. UTILITIES Shape Gen DETAIL Screens Additional Items

Item	Description
Create or Edit Program	This item allows you to create a new program name for the shape program that will be created. At this item you can also select an existing program to edit. Press F4, [CHOICE] to bring up a list of existing programs.

Table 10–91. UTILITIES Shape Gen DETAIL Screens Additional Items (Cont'd)

Item	Description
Center position	This item appears near the end of the table. It indicates whether a center position has been recorded. By moving to this item you can move to the recorded center position or touch up the position. If Step to shape center (set up on the SETUP Shape screen) is enabled, the robot will step to the center position.
Servo schedule This item appears only if the Small Circle Accuracy option is installed.	This item is the number of the servo schedule that will be used to cut this shape. You can define up to 20 servo schedules to improve the cut quality if required. The default value is 0 and the maximum value is 20.

Note The data you see in a specific shape UTILITIES Shape Gen DETAIL screen is populated with information from the SETUP Shapes DETAIL screen. If the default data is not correct for your normal operation, modify it at the SETUP Shapes DETAIL screen. Refer to Section 10.42.4 .

Figure 10–154. UTILITIES Shape Gen DETAIL Screen Example

```
UTILITIES Circle
                                1/24
 1 Create or Edit Program:
 2 Diameter:                  20.00 mm
 3 Circular speed:             60 mm/sec
 4 Linear speed:               60 mm/sec
 5 Circle type:                FULL
 6 Edge distance:              0.00 mm
 7 Start axis:                  +X
 8 X Work angle:               0.00 deg
 9 Y Work angle:               0.00 deg
10 Blend-in angle:            20.00 deg
11 Blend-in distance:          3.00 mm
12 Overlap angle:             10.00 deg
13 Blend-off angle:          −45.0 deg
14 Blend-off dist:            20.00 mm
15 Num of circ points:            2
16 Cut direction:              CW
17 Rotation:                  0.00 deg
18 Tool cut diameter:         0.00 mm
19 Plunge distance:           0.00 mm
20 Plunge speed:            390 cm/min
21 Retract distance:          12.00 mm
22 Retract speed:           390 cm/min
23 Center position:          Not Rec
```

Procedure 10-61 Generating or Modifying a Shape Program

Steps

1. Press MENUS.
2. Select UTILITIES.
3. Press F1, TYPE.
4. Select Shape Gen and press ENTER. You will see a screen similar to the following:

Figure 10–155. UTILITIES Shape Gen Screen

```
UTILITIES Shape
          Shape select:
        1 Circle:
        2 Slot:
        3 Hexagon:
        4 Rectangle
        5 Keyhole:
        6 Pommel:
        7 Custom:
```

5. Select the shape you want to create or modify and press DETAIL.
6. Start a new program, select an existing program to edit, or copy an existing program:

- **To start a new program**,
  1. Move the cursor to Create or Edit Program and press ENTER.
  2. Type a name for the shape program that will be generated and press ENTER. The name of the new program will move to Program Name.

  If all the default shape data is correct for the new program, go to Step 8 to generate the program.

  If you want to modify the default data for the new program, go to Step 7 .

  If the original default data is not correct for your normal operation, you can modify it at the SETUP Shapes DETAIL screens. Refer to Section 10.42.4 .

- **To edit an existing program**,
  3. Move the cursor to Create or Edit Program and press F4, [CHOICE].
  4. Select the name of the program you want to modify and press ENTER.

- **To copy an existing program** if a generated shape program that you want to use for another shape of the same type already exists,

  5. Move the cursor to Program Name and press CHOICE.
  6. Select the name of the program you want to copy and press ENTER.
  7. Move the cursor to Create or Edit Program and enter the new program name. The new program data will be copied from the existing one.

  You can now go to Step 7 to modify data or go directly to Step 8 to generate the new program.

7. Modify shape data as desired. The modifications you make here will affect only the current shape. The default shape data will not change.



Caution

If you leave the DETAIL screen before pressing CREATE to generate the program, when you return it will be re-populated from the default data. Any changes you made to data will be lost.

- For information about DETAIL screen items that are common to all shapes, refer to Table 10–82 .
  For information about DETAIL screen items that appear only on the UTILITIES DETAIL screen, refer to Table 10–91 .
- For information about Circle-specific DETAIL screen items, refer to Table 10–83 .
- For information about Slot-specific DETAIL screen items, refer to Table 10–86 .
- For information about Hexagon-specific DETAIL screen items, refer to Table 10–84 .
- For information about Rectangle-specific DETAIL screen items, refer to Table 10–85 .
- For information about Keyhole-specific DETAIL screen items, refer to Table 10–87 .
- For information about Pommel-specific DETAIL screen items, refer to Table 10–88 .
- For information about Custom-specific DETAIL screen items, refer to Table 10–89 .

8. **To generate the program** when all the information is correct, press CREATE. This will create the new shape program or re-create the existing program you edited.

Note When you press CREATE to generate a new program, the system will always use the current location of the group that is selected. However, if you are editing an existing program, the TCP that was recorded in the existing program will always be used. Changes you make to the TCP while editing the existing program will not take effect. If you want to move the center point, you must move the TCP to the new location and use TOUCHUP to re-record the TCP.

9. Add your application and process-specific instructions to the PROC_ON and PROC_OFF instructions.

   Note The instructions you enter in the PROC_ON and PROC_OFF instructions are never changed by the Shape Generation II option. Any other modifications made to the program logic will be lost when you press CREATE.

10. **To run the program**

    a. Enable the Teach Pendant.

    b. Press and hold the deadman switch.

    c. Press and hold SHIFT.

    d. Press TEST.

Note **Single step** is not supported from the TEST key. If you need to single step the program, go to the SELECT program screen and run it from there.

This will leave the TCP at the last point in the program. If you want to move back to the center point, move the curser to Center position (located near the bottom of the screen). Press F4, TOUCHUP or F5, MOVETO to re-teach the center position or move to the center position.

If the shape is not exactly the way you want it to be, you can change the data in the UTILITIES Shape Gen DETAIL screen, then press CREATE again to re-create that shape. You will see a screen similar to the following:

```
This will DELETE the program
file and all of the
information that it contains.
Are you sure you want to
continue?
```



Caution

Pressing CREATE again will completely erase the current program and re-create it with the same name, using the data you set at the UTILITIES Shape Gen DETAIL screen. The instructions you enter in the PROC_ON and PROC_OFF instructions are never changed by the Shape Generation II option. Any other modifications made to the program logic will be lost when you press CREATE.

Examples of shape programs generated by the Shape Generation II option are shown in Figure 10–156 and Figure 10–157 .

Figure 10–156. Shape Generated Program Example ACIR1

```
/PROG  ACIR1
/ATTR
OWNER  = KAREL;
COMMENT  = "Cut Circle GP1";
PROG_SIZE = 1169;
CREATE  = DATE 07-04-16  TIME 14:33:56;
MODIFIED = DATE 07-04-16  TIME 14:33:56;
FILE_NAME = ;
VERSION  = 0;
LINE_COUNT = 17;
MEMORY_SIZE = 1593;
PROTECT  = READ;
TCD:  STACK_SIZE = 0,
      TASK_PRIORITY = 50,
      TIME_SLICE = 0,
      BUSY_LAMP_OFF = 0,
      ABORT_REQUEST = 0,
      PAUSE_REQUEST = 0;
DEFAULT_GROUP = 1,*,*,*,*,*,*;
CONTROL_CODE = 00000000 00000000;
/APPL
/MN
   1:  !SHAPE GENERATED CIRCLE ;
   2:  UTOOL_NUM=1 ;
   3:  UFRAME_NUM=0 ;
   4:  IF (F[101]=OFF),JMP LBL[1] ;
   5:L P[1] 60mm/sec FINE    ;
   6:  LBL[1] ;
   7:  IF (F[102]=ON),JMP LBL[2] ;
   8:L P[2] 60mm/sec FINE    ;
   9:  CALL PROC_ON1    ;
  10:L P[3] 60mm/sec FINE    ;
  11:L P[4] 60mm/sec CNT100    ;
  12:C P[5]
     :  P[6] 60mm/sec CNT100    ;
  13:C P[7]
     :  P[8] 60mm/sec CNT100    ;
  14:L P[9] 60mm/sec CNT10    ;
  15:L P[10] 60mm/sec FINE    ;
  16:  CALL PROC_OF1    ;
  17:  LBL[2] ;
```

Figure 10–157. Shape Generated Program Example ACIR2

```
/PROG  ACIR2
/ATTR
OWNER  = KAREL;
COMMENT  = "Cut Circle GP1";
PROG_SIZE = 1169;
CREATE  = DATE 07-04-19  TIME 08:48:00;
MODIFIED = DATE 07-04-19  TIME 08:48:00;
FILE_NAME = ;
VERSION  = 0;
LINE_COUNT = 17;
MEMORY_SIZE = 1593;
PROTECT  = READ;
TCD:  STACK_SIZE = 0,
      TASK_PRIORITY = 50,
      TIME_SLICE = 0,
      BUSY_LAMP_OFF = 0,
      ABORT_REQUEST = 0,
      PAUSE_REQUEST = 0;
DEFAULT_GROUP = 1,*,*,*,*,*,*;
CONTROL_CODE = 00000000 00000000;
/APPL
/MN
   1:  !SHAPE GENERATED CIRCLE ;
   2:  UTOOL_NUM=1 ;
   3:  UFRAME_NUM=0 ;
   4:  IF (F[101]=OFF),JMP LBL[1] ;
   5:L P[1] 60mm/sec FINE    ;
   6:  LBL[1] ;
   7:  IF (F[102]=ON),JMP LBL[2] ;
   8:L P[2] 60mm/sec FINE    ;
   9:  CALL PROC_ON1    ;
  10:L P[3] 60mm/sec FINE    ;
  11:L P[4] 60mm/sec CNT100    ;
  12:C P[5]
     :  P[6] 60mm/sec CNT100    ;
  13:C P[7]
     :  P[8] 60mm/sec CNT100    ;
  14:L P[9] 60mm/sec CNT10    ;
  15:L P[10] 60mm/sec FINE    ;
  16:  CALL PROC_OF1    ;
  17:  LBL[2] ;
```

Production Program

Shape Generation II only creates shape programs. You must write a production application program that does cell operations, required startup checks, and calls the generated shapes for the specific operation.

Warning

In your production program, you must be aware of the calling sequence of the shape programs for the proper execution of your application program. Otherwise, you could injure personnel or damage equipment.

Make sure the production program stops the process (such as laser or water jet cutting) when an error occurs. The Shape Generation II option does not automatically turn the process equipment off when an error occurs. For example, if you are cutting shapes using a laser or water jet, you must write the production program so that the process shuts off if an error occurs. If you do not, you could injure personnel or damage equipment.

Figure 10–158 shows a simple example of a production program.

Figure 10–158. Production Program Example

```
 1:  !Move robot to home ;
 2:J P[1] 35% FINE    ;
 3:  !Move above first circ point ;
 4:J P[2] 100% FINE    ;
 5:  CALL ACIR1    ;
 6:  !Move above second circ point ;
 7:J P[3] 100% FINE    ;
 8:  CALL ACIR2    ;
 9:  !Move away for next part ;
10:   ;
11:  !ETC... ;
```

10.42.8 Servo Schedules

Up to 20 servo schedules are available, if the Small Circle Accuracy option is loaded along with the Shape Generation II option.

You can display a table summary of all data servo schedules from the DATA Servo screen.

Table 10–92. DATA Servo Screen Items

ITEM	DESCRIPTION
Schedule #	This item is the current schedule to which the defined data corresponds.
Coulomb	This item is the Coulomb friction gain which compensates for mechanical friction present in each joint of the robot.
Backlash	This item is the Backlash compensation gain which is used for fine tuning short motions and low speeds used for small shape cutting.
Comment	This item contains a description of the schedule.

See Figure 10–159 .

Figure 10–159. Servo Schedules

```
DATA Servo
#   Coulomb   Backlash   Comment
1   120%      55%        [  Circle 5  ]
2   100%      100%       [  Circle 6  ]
3   100%      100%       [  Default  ]
4   100%      100%       [  Default  ]
5   100%      100%       [  Default  ]
6   100%      100%       [  Default  ]
7   100%      100%       [  Default  ]
8   100%      100%       [  Default  ]
9   100%      100%       [  Default  ]
10  100%      100%       [  Default  ]
```

Each servo schedule can be used to fine tune cut quality if needed. Tuning small shapes involves optimizing the Coulomb friction compensation gain and the backlash compensation gains.

The Coulomb friction gain compensates for mechanical friction present in each joint of the robot. The default value of Coulomb friction compensation gain is tuned for general motion, in which case the effects of friction are minimal. However, for small shape cutting, the effects of friction on motion performance are more significant because the motion segments are generally very short and the motion speeds are very low as well. Fine tuning the Coulomb friction compensation gain might help improve motion performance of the robot.

As for Coulomb friction, the effect of backlash on motion performance becomes significant for the short motions and low speeds used for small shape cutting. Although the backlash compensation gain is tuned for general motion, the default value might not be optimal for small shape cuts. Fine tuning the backlash compensation gain might also contribute to better cut quality.

The default values for the Coulomb friction compensation gain and the backlash compensation gain are 100%. The allowed values for those gains range from 50% to 200%.

Warning

When using the servo schedules you must have substantial knowledge of the system. You must have enough knowledge to understand the changes made to the system when modifying the Coulomb friction compensation and Backlash compensation gains. Otherwise, you could injure personnel or damage equipment.

Warning

Changing the values of the Coulomb friction compensation gain and the backlash compensation gain will affect robot motion and behavior in such a way that it might present potential risk to you. Use extreme caution; otherwise, you could injure personnel or damage equipment.

The servo schedules can be used with any shape schedule. In this case, no additional programming is required.

The servo schedule feature can also be used without the Shape Generation II cut programs. In this case, a system macro is provided to manually access the servo schedules from a teach pendant program: SET_SRVO.TP.

Warning

When using the servo adjust macro to manually change the values of the servo gains, it is very important to reset the servo gains to their default values after the cut is performed. This ensures that the custom values that have been tuned for a specific cut are not used throughout the rest of the programmed motion. Otherwise, you could injure personnel or damage equipment.

10.42.9 Tuning Guidelines

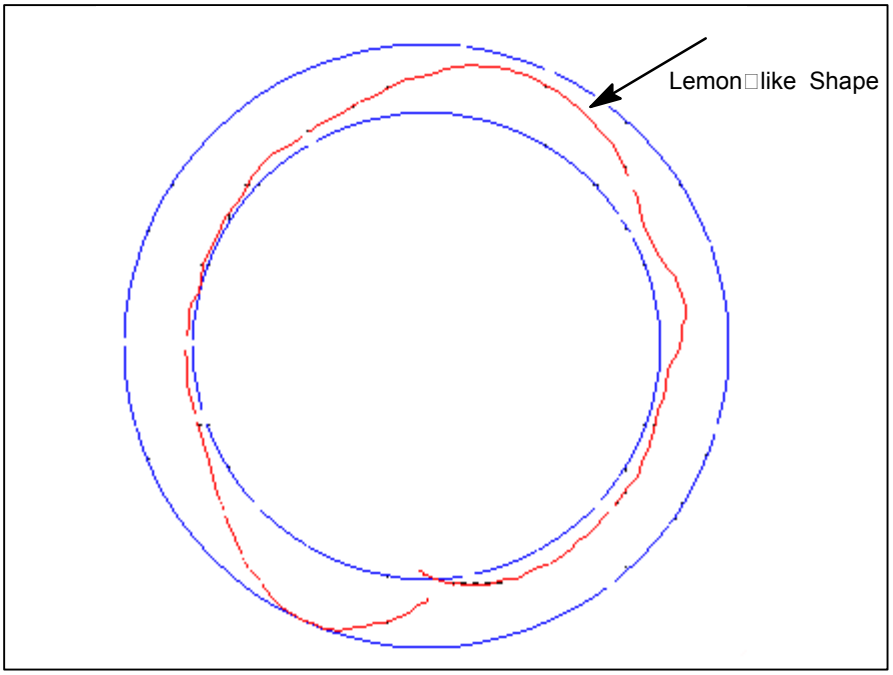
Use the following guidelines for the shapes listed below:

- Lemon shape circle
- Lemon shape circle with ripples
- Circle with flat spot
- Circle with inward bump

Lemon Shape Circle

To improve the *lemon-like* shape of the circle, increase the backlash compensation gain. See Figure 10–160 .

Figure 10–160. Lemon Shape with a Bump on the Right Side



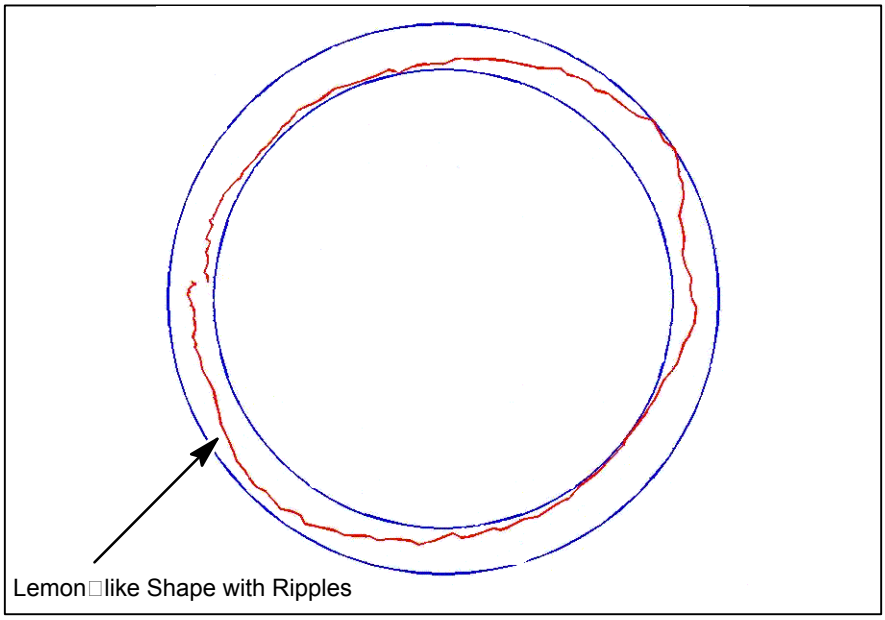
Lemon Shape Circle with Ripples

To avoid a lemon shape circle with ripples:

- Lower the backlash compensation gain.
- Adjust the Coulomb friction compensation gain up or down.

See Figure 10–161 .

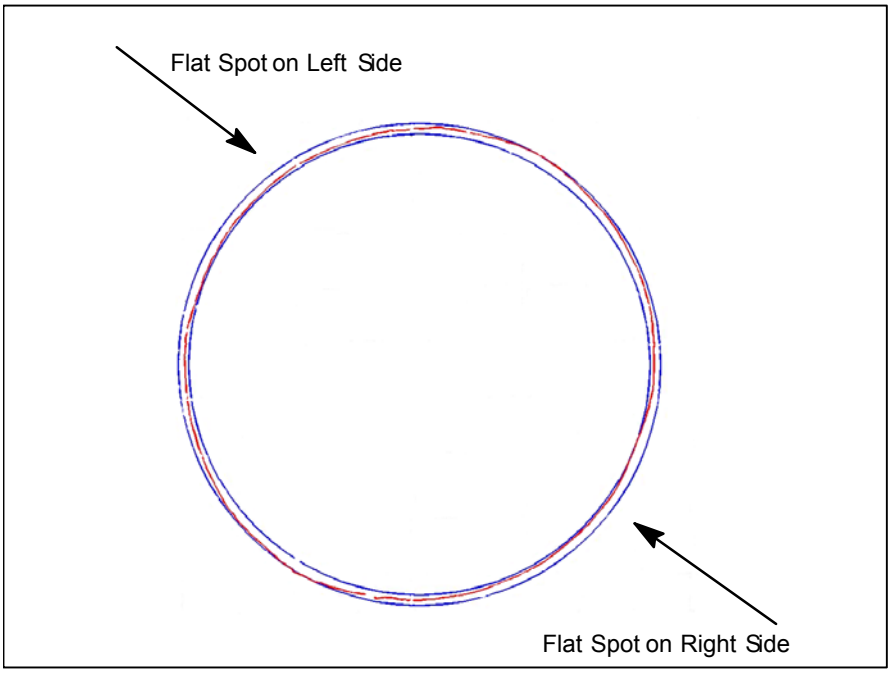
Figure 10–161. Lemon Shape with Outward Bump and Ripples



Circle with Flat Spot

To avoid a circle with a flat spot, lower the Coulomb friction compensation gain to reduce or eliminate the flat spots. See Figure 10–162 .

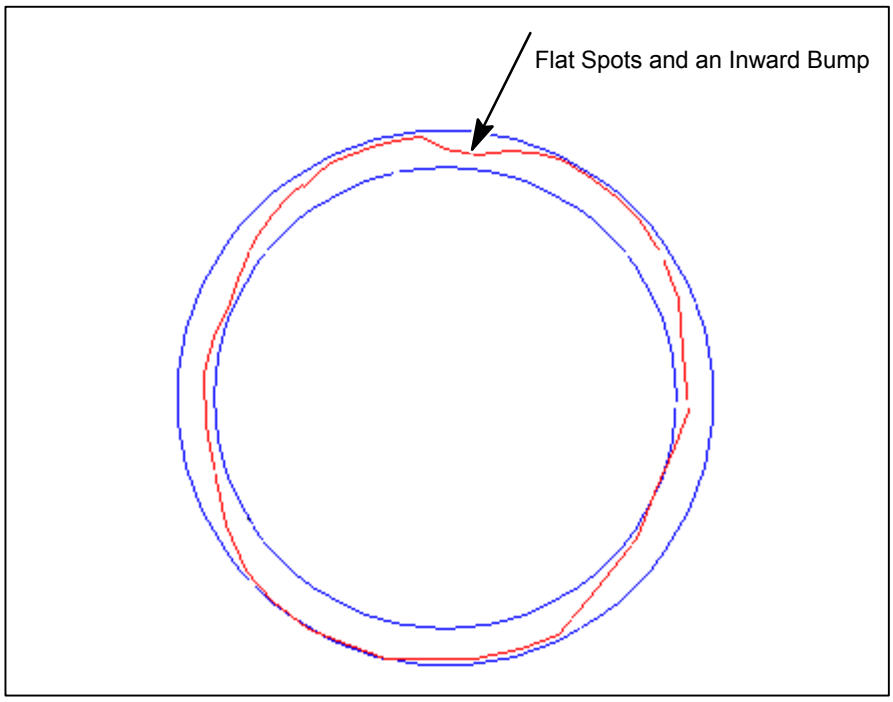
Figure 10–162. Flat Spot on Each Side



Circle with Inward Bump

To avoid this shape, lower the backlash compensation gain to reduce or eliminate the inward bump. See Figure 10–163 .

Figure 10–163. Two Flat Spots and an Inward Bump



10.43 SIMPLE TEACH SUITE

10.43.1 Overview

The Simple Teach Suite is a set of functions that make it simpler for you to teach a program.

You can touch up points at any time and run the Simple Teach Suite functions again to verify results without affecting production.

10.43.2 Auto Path Smoothing

10.43.2.1 Overview

Auto Path Smoothing makes user-taught robot paths smooth while allowing the tool orientation to maintain process-required angles. Auto Path Smoothing also makes position transition smooth. If you want to shift the positions of taught points, Auto Path Smoothing can automatically shift the entire path proportionally.

10.43.2.2 Limitations

Auto Path Smoothing cannot be used with coordinated motion or multi-group systems.

In order to obtain accurate results, you must have an accurate tool center point defined.

Only single-step motion is allowed in the teach pendant program editor “Simple Teach” mode.

10.43.2.3 How It Works

Auto Path Smoothing – Orientation

Orientation smoothing can be applied to a user-defined range of teach pendant programming instructions. The smoothing algorithm is based on the orientation of start and end points in this range of teach pendant instructions. In other words, the orientation of start and end points will not be changed after conversion. Therefore, these two points need to be taught correctly. Also, you can specify other fixed points between start and end points to avoid an obstacle or fine adjust the smoothing results.

There are two smoothing algorithms from which you can choose: two-angle approach (RS control) and three-angle approach (AES control).

If it is important that the tool stays on one side of the path, the three-angle approach is the better choice. In normal cases, such as corners, the two-angle approach will be sufficient.

Auto Path Smoothing – Position (Rubberbanding)

The Auto Path Smoothing – Position (Rubberbanding) function can shift the positions of taught points within user-defined teach pendant programming instructions, but the orientation will remain unchanged unless you enable orientation smoothing at the same time. Unlike orientation smoothing, rubberbanding requires at least one fixed point other than start and end points. You must then define the new location of the fixed point(s) by touching up the program using the Simple Teach Suite interface. The difference between the old and new location will then be used and re-organized for all the points within the selected range.

Guidelines for Using Path Smoothing

When the motion line is selected with circular motion type as the start or end line, only one of the positions can be used. For example, if the via position is selected as a start line, the destination position with the same motion line number cannot be selected as fixed-points or end line. However, both positions with the same motion line number can be selected as fixed-points. The Position Indicator feature ("@" sign will be used when robot reaches taught point) will be temporarily turned off when Simple Teach Suite is used in teach pendant editor. The original setting of the Position Indicator will be resumed after it leaves Simple Teach Suite. Also, if the Position Confirmation feature is turned on and you enter Simple Teach Suite, you might experience one less empty space between motion type and position ID.

Note If for any reason the smoothing results are not ideal for the specific application, the F1 UNDO key is available after the conversion has been successfully applied. However, UNDO will not be available if you enter Simple Teach Suite in another teach pendant program or after a Cold start occurs. Verify the smoothing results as soon as possible.



Caution

Do not use the Path Smoothing function with the 3-angle smoothing method selected for the range of the teach pendant program in which taught points have a zig-zag pattern. If it is needed the small range of the teach pendant program can be selected.

Table 10–93. Simple Teach Suite Main Setup Screen Items

Item	Description
Active Schedule	This item indicates the currently selected schedule.
Original Program	This item indicates the currently used teach pendant program.
Range	This item is the part of the teach pendant program that needs to be converted.
Start line	This item is the start line of this part of the teach pendant program.
End line	This item is the end line of this part of the teach pendant program.
Create/Replace	This item is used after conversion to replace the original teach pendant program or create a new one.
New Program	This item is used to create and name the new program.
Insert line	This item is used to insert a line from an existing program.

Note This screen will be set up (overridden) automatically when Auto Path Smoothing is used from the teach pendant editor.

Table 10–94. Auto Path Smoothing Menu Items

Item	Description
Schedule No [n]	This item displays comments about the current schedule.
Smooth orientation	This item is used to Enable/Disable orientation smoothing.
Smooth method	This item is used to select an orientation smoothing method.
Shift position	This item is used to Enable/Disable position smoothing (rubberbanding).
Reachability detect	This item is used to Enable/Disable taught point reachability detection.

Note This screen needs to be set up prior to using Auto Path Smoothing in the teach pendant editor. Otherwise, the default value will be used.

Use Procedure Procedure 10-62 to set up Auto Path Smoothing.

Use Procedure Procedure 10-63 to use Auto Path Smoothing — Orientation.

Use Procedure Procedure 10-64 to use Auto Path Smoothing – Position (Rubberbanding)

Procedure 10-62 Setting Up Auto Path Smoothing

Conditions

- A correct Tool Center Point has been taught.

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE], and choose Simple Teach. You will see a screen similar to the following.

```
ST1               LINE 0     JOINT 10%
Simple Teach Suite
Main menu:                          1/8
Group :[1]
  1 Active schedule:           1
  2 Original program:         [ST1     ]
  3   Range:                   PART
  4   Start  line:                 0
  5   End    line:                 0
  6 Create/Replace:           REPLACE
  7 New program:   (not used)[        ]
  8   Insert line: (not used) *****

[ TYPE ]  SCHED EXECUTE                >
```

4. Press F2, SCHED. You will see a screen similar to the following.

```
ST1               LINE 0    JOINT 10%
Simple Teach Suite
Schedule list                     1/10
 No  Type               Comment
  1  SMOOTH          [               ]
  2  SMOOTH          [               ]
  3  SMOOTH          [               ]
  4  SMOOTH          [               ]
  5  SMOOTH          [               ]
  6  SMOOTH          [               ]
  7  SMOOTH          [               ]
  8  SMOOTH          [               ]
  9  SMOOTH          [               ]
 10  SMOOTH          [               ]

[ TYPE ]  MAIN  DETAIL  ADJUST  SMOOTH
```

Note Auto Path Smoothing (SMOOTH) is selected for each schedule by default.

5. Move the cursor to select the schedule you want to edit and press F3, DETAIL. You will see a screen similar to the following.

```
DEMO_PIP          LINE 0     JOINT 10%
Simple Teach Suite
Auto smoothing menu:                1/5
  1 Schedule No[ 1]: [***************]
  2 Smooth orientation:       ENABLE
  3 Smooth method:          3-ANGLE
  4 Shift position:           DISABLE
  5 Reachability detect:      ENABLE

[ TYPE ]  MAIN   LIST                >
```

6. Move the cursor to select the items you want to change. When you are done,
   - Press F3, LIST, to return to the schedule list.
   - Press F2, MAIN, to return to the Simple Teach Suite main setup screen.

Procedure 10-63 Using Auto Path Smoothing – Orientation

Steps

1. Press Select.
2. Select the program you want to edit and press ENTER.
3. Press >, NEXT, to display additional menu function keys.
4. Press F5, [EDCMD], and choose Simple Teach. You will see a screen similar to the following.

```
DEMO_PIP          LINE 0      JOINT 10%
DEMO_PIP
                                     3/12
   1:J P[1] 10% FINE
   2:J P[2] 40% FINE
   3sL P[3] 15mm/sec CNT100
   4:L P[4] 15mm/sec CNT100
   5:L P[5] 15mm/sec CNT100
   6:L P[9] 15mm/sec CNT100
   7:L P[6] 15mm/sec CNT100
   8:L P[10] 15mm/sec CNT100
   9:L P[7] 15mm/sec FINE
  10:J P[8] 10% FINE
  11:J P[1] 10% FINE
Move cursor to end line then press F2
         FIX_PT  APPLY   EXIT  TOUCHUP
```

5. Move the cursor or single step to the Start line and press F2, FIX_PT. An “s” will replace the “:” in the motion line. You will see a screen similar to the following.

Note The sequence in which you select the "s", "m", and "e" points is not important because they will be automatically sorted. You only need to press the F2, FIX_PT key. These points can be deselected by pressing F2, FIX_PT while the cursor is located on the line.

Note If you leave Simple Teach Suite without successfully applying smoothing, the original teach pendant program will be restored and touchup points will be lost.

Note The original teach pendant program can still be restored by pressing F1 UNDO after smoothing has been applied successfully. F1 UNDO will not be available if you have already used Simple Teach in the other teach pendant program.

```
DEMO_PIP          LINE 0      JOINT 10%
DEMO_PIP
                                     9/12
   1:J P[1] 10% FINE
   2:J P[2] 40% FINE
   3sL P[3] 15mm/sec CNT100
   4:L P[4] 15mm/sec CNT100
   5:L P[5] 15mm/sec CNT100
   6:L P[9] 15mm/sec CNT100
   7:L P[6] 15mm/sec CNT100
   8:L P[10] 15mm/sec CNT100
   9eL P[7] 15mm/sec FINE
  10:J P[8] 10% FINE
  11:J P[1] 10% FINE
Select more fixed pts or apply
         FIX_PT  APPLY   EXIT  TOUCHUP
```

6. Move the cursor or single step to the End line and press F2, FIX_PT. An “e” will replace the “:” in the motion line. You will see a screen similar to the following.

```
DEMO_PIP          LINE 0     JOINT 10%
DEMO_PIP
                                     6/12
   1:J P[1] 10% FINE
   2:J P[2] 40% FINE
   3sL P[3] 15mm/sec CNT100
   4:L P[4] 15mm/sec CNT100
   5:L P[5] 15mm/sec CNT100
   6mL P[9] 15mm/sec CNT100
   7:L P[6] 15mm/sec CNT100
   8:L P[10] 15mm/sec CNT100
   9eL P[7] 15mm/sec FINE
  10:J P[8] 10% FINE
  11:J P[1] 10% FINE
Select more fixed pts or apply
         FIX_PT  APPLY   EXIT  TOUCHUP
```

7. If necessary, move the cursor or single step to any additional lines that you want to set as fixed points and press F2, FIX_PT. An “m” will replace the“:” in each motion line. You will see a screen similar to the following.

```
DEMO_PIP          LINE 0     JOINT 10%
DEMO_PIP
                                     6/12
   1:J P[1] 10% FINE
   2:J P[2] 40% FINE
   3sL P[3] 15mm/sec CNT100
   4:L P[4] 15mm/sec CNT100
   5:L P[5] 15mm/sec CNT100
   6mL P[9] 15mm/sec CNT100
   7:L P[6] 15mm/sec CNT100
   8:L P[10] 15mm/sec CNT100
   9eL P[7] 15mm/sec FINE
  10:J P[8] 10% FINE
  11:J P[1] 10% FINE
Select more fixed pts or apply
         FIX_PT  APPLY   EXIT  TOUCHUP
```

8. When you are finished, press F3, APPLY, to execute Auto Path Smoothing.
9. Type the proper schedule number and press ENTER. You will be prompted to confirm the settings before the conversion.

Procedure 10-64 Using Auto Path Smoothing – Position (Rubberbanding)

Steps

1. Press Select.
2. Select the program you want to edit and press ENTER.
3. Press Next to display additional menu function keys.
4. Press F5, [EDCMD], and choose Simple Teach. You will see a screen similar to the following.

```
DEMO_PIP          LINE 0      JOINT 10%
DEMO_PIP
                                     3/12
   1:J P[1] 10% FINE
   2:J P[2] 40% FINE
   3:L P[3] 15mm/sec CNT100
   4:L P[4] 15mm/sec CNT100
   5:L P[5] 15mm/sec CNT100
   6:L P[9] 15mm/sec CNT100
   7:L P[6] 15mm/sec CNT100
   8:L P[10] 15mm/sec CNT100
   9:L P[7] 15mm/sec FINE
  10:J P[8] 10% FINE
  11:J P[1] 10% FINE
Move cursor to end line then press F2
         FIX_PT  APPLY   EXIT  TOUCHUP
```

5. Move the cursor or single step to the Start line and press F2, FIX_PT. An “s” will replace the “:” in the motion line. You will see a screen similar to the following.

Note The sequence in which you select the "s", "m", and "e" points is not important because they will be automatically sorted. You only need to press the F2, FIX_PT key. These points can be deselected by pressing F2, FIX_PT while the cursor is located on the line.

Note If you leave Simple Teach Suite without successfully applying smoothing, the original teach pendant program will be restored and touchup points will be lost.

Note The original teach pendant program can still be restored by pressing F1 UNDO after smoothing has been applied successfully. F1 UNDO will not be available if you have already used Simple Teach in the other teach pendant program.

```
DEMO_PIP          LINE 0      JOINT 10%
DEMO_PIP
                                     3/12
   1:J P[1] 10% FINE
   2:J P[2] 40% FINE
   3sL P[3] 15mm/sec CNT100
   4:L P[4] 15mm/sec CNT100
   5:L P[5] 15mm/sec CNT100
   6:L P[9] 15mm/sec CNT100
   7:L P[6] 15mm/sec CNT100
   8:L P[10] 15mm/sec CNT100
   9:L P[7] 15mm/sec FINE
  10:J P[8] 10% FINE
  11:J P[1] 10% FINE
Move cursor to end line then press F2
         FIX_PT  APPLY   EXIT  TOUCHUP
```

6. Move the cursor or single step to the End line and press F2, FIX_PT. An “e” will replace the “:” in the motion line. You will see a screen similar to the following.

```
DEMO_PIP          LINE 0      JOINT 10%
DEMO_PIP
                                     9/12
   1:J P[1] 10% FINE
   2:J P[2] 40% FINE
   3sL P[3] 15mm/sec CNT100
   4:L P[4] 15mm/sec CNT100
   5:L P[5] 15mm/sec CNT100
   6:L P[9] 15mm/sec CNT100
   7:L P[6] 15mm/sec CNT100
   8:L P[10] 15mm/sec CNT100
   9eL P[7] 15mm/sec FINE
  10:J P[8] 10% FINE
  11:J P[1] 10% FINE
Select more fixed pts or apply
         FIX_PT  APPLY   EXIT  TOUCHUP
```

7. Move the cursor or single step to any additional lines that you want to set as fixed points and press F2, FIX_PT. An “m” will replace the “:” in each motion line. You will see a screen similar to the following.

```
DEMO_PIP          LINE 0     JOINT 10 %
DEMO_PIP
                                    6/12
   1:J P[1] 10% FINE
   2:J P[2] 40% FINE
   3sL P[3] 15mm/sec CNT100
   4:L P[4] 15mm/sec CNT100
   5:L P[5] 15mm/sec CNT100
   6mL P[9] 15mm/sec CNT100
   7:L P[6] 15mm/sec CNT100
   8:L P[10] 15mm/sec CNT100
   9eL P[7] 15mm/sec FINE
  10:J P[8] 10% FINE
  11:J P[1] 10% FINE
Select more fixed pts or apply
         FIX_PT  APPLY   EXIT  TOUCHUP
```

8. Define the new location of the fixed points.
   a. Jog the robot TCP to the new location.
   b. Press F5, TOUCHUP.
9. When you are finished, press F3, APPLY, to execute Auto Path Smoothing.
10. Type the proper schedule number and press ENTER. You will be prompted to confirm the settings before the conversion.

10.44 SIMULATED INPUT SKIP

10.44.1 Overview

Simulated Input Skip provides a way for TP programs that are WAITing on simulated inputs to continue after a specified time if the input condition is not satisfied. You can define a Simulated Input Wait Delay Time, that will apply to all programs, and to designate specific inputs as Skip When Simulated inputs.

With Simulated Input Skip enabled, if a running program executes a WAIT for an input that is simulated and designated as Skip When Simulated, the program will either continue when the input condition is satisfied or it will continue after the specified delay time.

10.44.2 Setup

Two setup procedures are required to configure the system to use this feature.

You must:

- enable the feature
- designate any input points that are to be used with the feature as Skip When Simulated.

Enabling Simulated Input Skip

1. Press MENUS.
2. Select SYSTEM.
3. Press TYPE.
4. Select CONFIG. You should see a screen similar to the following:

```
System/Config
31 Signal to set in AUTO mode DO[    0]
32 Signal to set in T1 mode   DO[    0]
33 Signal to set in T2 mode   DO[    0]
34 Signal to set if E-STOP    DO[    0]
35 Set if INPUT SIMULATED     DO[    0]
36 Sim. Input Wait Delay:      0.00 sec
37 Set if Sim. Skip Enabled   DO[    0]
38 Hand broken :             <*GROUPS*>
39 Remote/Local setup:    Remote
40   External I/O (ON:Remote) :DI[   0]
41 UOP auto assignment:   Full
```

5. Move the cursor to Sim. Input Wait Delay and enter a value from 0 to 99.99. This setting is the time in seconds that a WAIT instruction will delay before continuing.
6. Move the cursor to Set if Sim. Skip Enabled and enter the index of a valid digital output that will be turned on to indicate that Simulated Input Skip is enabled. Valid numbers for this item are 0 through 32766.
7. Turn the controller off, then on again to recognize the change.

Designating the Inputs to Time Out when Simulated

1. Press MENU.
2. Select I/O.
3. Press TYPE.
4. Select Digital or Robot and use the IN/OUT function key to select inputs. You will see a screen similar to the following:

```
I/O Digital in
                                      1/512
  DI[   1] U  OFF  [                    ]
DI[   1] U  OFF  [                    ]
DI[   2] U  OFF  [                    ]
DI[   3] U  OFF  [                    ]
DI[   4] U  OFF  [                    ]
DI[   5] U  OFF  [                    ]
DI[   6] U  OFF  [                    ]
DI[   7] U  OFF  [                    ]
DI[   8] U  OFF  [                    ]
DI[   9] U  OFF  [                    ]
DI[  10] U  OFF  [                    ]
DI[  11] U  OFF  [                    ]
```

5. Move the cursor to the input point you want to designate as Skip When Simulated.

6. Press NEXT until the DETAIL function key is displayed, and press DETAIL. You will see a screen similar to the following:

```
I/O Digital in
Port Detail                        3/3
    Digital Input           [   1]

1 Comment:  [                         ]
2 Polarity:              NORMAL
3 Skip when simulated:   FALSE
```

7. Move the cursor to Skip when simulated and press

- TRUE to set the timeout designation, or
- FALSE to clear the timeout designation.

8. Press PRV-PT or NXT-PT to do the same setting for other input points.

9. Press PREV to return to the I/O monitor menu. You will see a screen similar to the following: :

```
I/O Digital in
                                        1/512
DI[   1](S) OFF  [                      ]
DI[   1](S) OFF  [                      ]
DI[   2](S) OFF  [                      ]
DI[   3](U) OFF  [                      ]
DI[   4](U) OFF  [                      ]
DI[   5] U  OFF  [                      ]
DI[   6] U  OFF  [                      ]
DI[   7] U  OFF  [                      ]
DI[   8] S  OFF  [                      ]
DI[   9] S  OFF  [                      ]
DI[  10] U  OFF  [                      ]
DI[  11] U  OFF  [                      ]
```

On this screen, any points that have been designated as Skip When Simulated will have parenthesis '( )' in the Sim column. If you have an *i*Pendant, if the point is simulated, the Sim entry will be highlighted in green.

10.44.3 Operation

After you set up Simulated Input Skip and designate input points as simulated and Skip when simulated, you can use these inputs in a WAIT instruction in any teach pendant program. The WAIT statement will time out if the wait condition is not satisfied. The WAIT statement does not need any additional clauses.

The following are examples of this type of WAIT instruction:

```
WAIT DI[1] = ON
WAIT DI[2] = OFF-
WAIT RI[1] <> DO[5]
```

Limitations

There are two restrictions on the use of Simulated Input Skip in a WAIT instruction.

- If the WAIT instruction contains a TIMEOUT clause, the instruction will time out based on the clause rather than the Simulated Input Skip, even if Simulated Input Skip is enabled and a Skip when simulated input is used in the instruction. After time out, program execution will continue at the label specified in the TIMEOUT clause.
- The WAIT instruction must contain only one condition, and not several ANDed or ORed together. For example, the following instructions will not time out even if Simulated Input Skip is enabled and if they contain 'Skip when simulated' inputs.

```
    WAIT DI[1] = ON AND DI[2] = ON
WAIT DI[1] = ON OR RI[2] = ON OR RI[3] = ON
```

Program Example

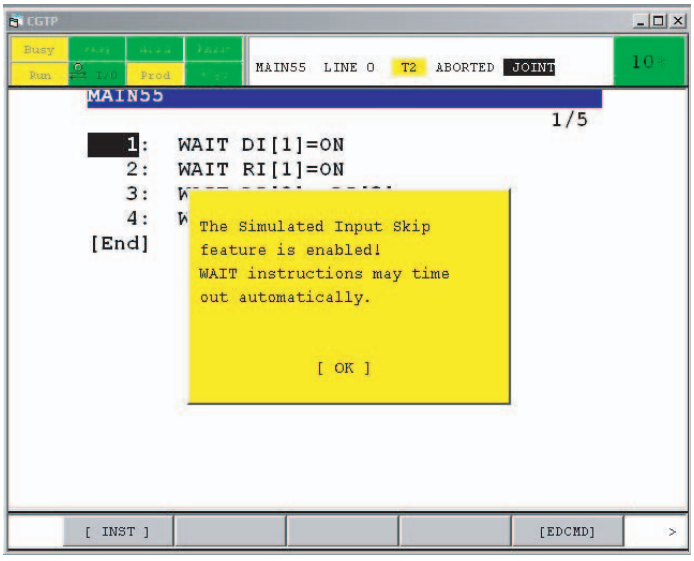
The following is an example of a program that can use Simulated Input Skip. If all of the inputs to the left of the operators are designated as Skip When Simulated and if they are all simulated, when each instruction is executed, the WAIT will time out in the time specified in Sim. Input Wait Delay if the condition is not satisfied.

Note DI[3] on the right side of the operator in line 3 does not need to be designated as Skip When Simulated.

```
 1:  WAIT DI[1]=ON
 2:  WAIT RI[1]=ON
 3:  WAIT DI[2]<>DI[3]
 4:  WAIT RI[5]=OFF-
[End]
```

When a program is started from the teach pendant using SHIFT FWD or SHIFT BWD, if the feature is enabled (that is,‘Sim. Input Wait Delay’ is greater than zero and at least one input point is defined to timeout) a pop-up window will be displayed indicating that the program is being run in Simulated Input Skip mode.

Figure 10–164. Simulated Input Skip Enabled Screen



This is to alert an operator running a program in test mode that WAIT instructions may time out.

When this message is displayed, you must press ENTER on the Teach Pendant to begin program execution. This warning will not be displayed if a program is started from a source other than the pendant, such as the SOP, UOP or from KCL.

Error Messages

Two error messages are used by Simulated Input Skip. They are:

- PRIO-188 PAUSE Sim. Skip output not defined

  PRIO-188 will be posted to indicate that the‘Set if Sim. Skip Enabled’ configuration item is not set properly. IE it is set to zero or to an invalid digital output index. This error is posted at run time when a WAIT instruction is executed if Simulated Input Skip is enabled and the WAIT contains a ‘Skip when simulated’ input. The program will pause on the WAIT instruction. A user can continue program execution by first unsimulating the input used in the instruction or by setting the Skip When Simulated mode of the input to FALSE and then restarting the program.

- PRIO-189 WARN (%s^4, %d^5) WAIT will time out

  PRIO-189 will be posted whenever a WAIT instruction that will time out due to Simulated Input Skip is executed. This error will not stop execution. It will contain the program name and line number being executed and the text “WAIT will time out”. The purpose of this error is to provide additional warning that WAIT will time out.

10.45 SINGULARITY CHECK FUNCTION

When positional data is used in Cartesian coordinates to perform motion instruction teaching or position modification, the robot position might be close to a singularity point. In such a case, when the taught motion instruction is executed, the robot can operate at an attitude other than that with which the instruction was taught.

To prevent this, the singularity check function is provided. When a position is taught, this function checks whether the taught position is at a singularity position. The function can record the position in joint coordinate format, if so specified by the user.

Function

To enable the singularity check function, set the system variable $MNSING_CHK to TRUE.

If a motion instruction is taught using the SHIFT and POINT keys, or if a position is modified using the SHIFT and TOUCHUP keys, while the robot is at a singularity point, this function checks whether the taught position is at a singularity point.

This check is performed when all of the following conditions are satisfied:

- The position is recorded in Cartesian coordinates.
- An additional instruction with INC or offset is not specified.
- The UF (user frame number) of the position data is 0.

If the check finds that the taught position is at a singularity point, the following warning message appears in the upper two lines on the teach pendant:

```
[Can't record on Cartesian (G:i)]
[In singularity]
```

where, "i" is the number of the motion group at the singularity point.

At the same time, the following prompt appears at the bottom on the teach pendant:[Record current position on joint?]

The [YES] and [NO] function keys are displayed. Select the appropriate key.

- [YES]: The position data is recorded in joint coordinate format.
- [NO]: The position is not taught or modified.

For position data in a program that has multiple motion groups, the singularity check is performed in ascending order of group number.When more than one group is at a singular point, the warning message and prompt message appear repeatedly for each group.

You cannot use this function to teach BOTTOM POINTS for the palletizing function or to teach the ROUTE POINTS.

10.46 SOFT FLOAT (OPTION)

10.46.1 Overview

Soft Float is a feature that is used to compensate for variances in workpiece precision in applications where the robot is used to mount workpieces on a machine tool.

In most applications, the robot moves accurately to the taught points. When the robot is used to mount workpieces on a machine tool, variances in workpiece precision can cause the workpiece position to shift relative to the tool. This shifting could result in interference between the workpiece and the tool. The Soft Float function can compensate for these variances.

There are several robots that do not support all Soft Float types. The following Soft Float types are available:

- Joint Soft Float — Softness around the robot joint can be specified.
- Cartesian Soft Float — Softness along and around Cartesian coordinates can be specified. It is good at contouring and face matching with robot motion.
- Pushout Soft Float — (High Performance Soft Float only) Force and moment necessary become smaller than the current Cartesian Soft Float. Pushout Soft Float does not work where J5 = 3.5 ~ +3.5. It is good at following the pushout.

The Soft Float function is enabled and disabled using an instruction in a teach pendant program. Its conditions are also specified using the instruction.

To use the Soft Float function, you must do the following:

- Set **payload information**. Accurate payload information is crucial in getting the desired result with the Soft Float function. Refer to Section 3.12 .
- Set up **Soft Float schedules**. Setting up schedules involves choosing between Cartesian and Joint mode for each schedule, as well as specifying the flexibility. Refer to Section 10.46.3 for details on setting up Soft Float schedules.
- Add **Soft Float instructions** to your teach pendant program. For a description of Soft Float teach pendant instructions, refer to Section 10.46.6 .

10.46.2 Restrictions

When you use the Soft Float function, keep in mind the following restrictions:

Warning

When the Soft Float function is enabled,

•The robot might not always follow the taught path

•The taught path changes according to the override

•Execution time might be increased

Be aware of these effects before you enable the Soft Float function. Otherwise, you could injure personnel or damage equipment.

Warning

When the Soft Float function is enabled, robot brakes are affected as follows:

•If an attempt is made to enable Soft Float with a brake enabled, the brake is released automatically before the function is enabled.

•When the Soft Float function is enabled, brake control is ineffective.

Be aware of these effects before you use the Soft Float function. Otherwise, you could injure personnel or damage equipment.

- The Soft Float function is **disabled automatically** when
  - — Program execution starts.
  - — Program execution ends.
  - — The program stops due to an alarm that turns off the servomotors.
  - — Jogging is performed with the program paused.
  - — The program is restarted after the cursor is moved manually with the program paused.
  - — Backward execution is performed.
  - — Power is turned on.

- If the **program is paused and then restarted**, the states of the Soft Float function (such as enabled/disabled and the exec start ratio) are set to the conditions that existed before the program was paused, except for the cases listed in the previous item that cause the Soft Float function to be disabled automatically.
- The Soft Float function **cannot be enabled** by any method other than by using the Soft Float instruction.
- When the **Soft Float function is enabled**, the robot moves using the termination type CNT0 (no position check is made), even if FINE has been specified as motion instruction termination type.
- When the **Soft Float function is enabled**, if an external force causes the robot to move beyond a certain distance, the following servo error messages are displayed:
  - — If the robot is at rest: [SRVO-023 Stop error excess(G:i A:j)]
  - — If the robot is operating: [SRVO-024 Move error excess(G:i A:j)]

- If an attempt is made to **enable the Soft Float function with a brake applied,** the brake is released automatically before the function is enabled.
- When the **Soft Float function is enabled**, brake control is ineffective.

- If the **motion group mask in a program is [*,*,*,*,*,*,*,*]** (there is no motion group), when the program issues instructions with the Soft Float function, the following error message is displayed:[INTP-216 (program name, line number) Invalid value for group number]
- Minimize the **range of motion** used with the Soft Float function enabled.

  Load distribution on the robot's axes varies with the position and speed of the robot. When the robot is moved away from the nominal position (at which the Soft Float function was enabled), these variations can cause the robot to move in a direction other than the direction of the applied external force. Since gravity is the most dominant load, this motion is often along the z direction. As a result of this behavior, you should try to keep the range and speed of robot motion to a minimum when the Soft Float function is enabled.

10.46.3 Setting Up Joint Soft Float Schedules

You specify Soft Float schedules on the SETUP Soft Float screen. You can specify up to ten schedules for the Soft Float function.

Table 10–95 lists and describes each item Soft Float schedule item on the SETUP Soft Float screen. Refer to the motion option section of Section 10.46.6 for more information on the exec start ratio.

Use Procedure 10-67 to set up Joint Soft Float schedules.

Table 10–95. Joint Soft Float Schedule Setup Items

ITEM	DESCRIPTION
Comment	This item allows you to type a description of the schedule. To enter a comment move the cursor to line 1, press ENTER, and then press the appropriate function keys to type the comment. When you have finished, press ENTER.
Exec Start Ratio	This item specifies the point where the Soft Float function is enabled if the Soft Float[n] instruction is used as a motion option.
Axis***n*** Soft Ratio (Servo Flexibility)	This item, also known as servo flexibility or softness ratio, indicates how strongly the axis resists external forces. It is specified between 0% and 100%. A flexibility of 100% corresponds to being the most flexible. Servo flexibility for each axis (*n*) can be specified on the third line of the DETAIL screen and on subsequent lines.
Enabled/disabled	This item specifies whether the Soft Float function is enabled or disabled for each axis. Specify this on line 3 of the DETAIL screen and on subsequent lines. Move the cursor to the rightmost end (enabled/disabled setting position) of each line to cause the F4 (ENABLE) and F5 (DISABLE) keys to appear. Use these keys to specify whether to enable or disable the Soft Float function.

Procedure 10-67 Setting Up Joint Soft Float Schedules

Conditions

- You have set up payload information properly. Refer to Section 3.12 .

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Soft Float. See the following screen for an example of the listing screen.

```
SETUP Soft Float
Group 1
 No    TYPE     START %       Comment
  1     CART       ***    [                ]
  2     CART       ***    [                ]
  3     CART       ***    [                ]
  4     CART       ***    [                ]
  5     CART       ***    [                ]
  6     CART       ***    [                ]
  7     CART       ***    [                ]
  8     CART       ***    [                ]
  9     CART       ***    [                ]
 10     CART       ***    [                ]
```

5. **To change the display from detail to listing,** press F3, LIST.

   To change the display from listing to detail, press F3, DETAIL.

6. **To set up Joint Soft Float schedules,**

   a. Move the cursor to the schedule you want to set up and press F3, DETAIL, and then press F4, JOINT. See the following screen for an example.

```
SETUP/Soft Float
    Group 1
  1 Schedule No[   1]:***************]
  2 Exec Start Ratio :   0 %
  3 Axis1 Soft Ratio :   0 %   DISABLE
  4 Axis2 Soft Ratio :   0 %   DISABLE
  5 Axis3 Soft Ratio :   0 %   DISABLE
  6 Axis4 Soft Ratio :   0 %   DISABLE
  7 Axis5 Soft Ratio :   0 %   DISABLE
  8 Axis6 Soft Ratio :   0 %   DISABLE
  9 Axis7 Soft Ratio :   0 %   DISABLE
 10 Axis8 Soft Ratio :   0 %   DISABLE
 11 Axis9 Soft Ratio :   0 %   DISABLE
```

b. **To enable or disable the Soft Float function** for an axis, move the cursor to the word "ENABLE" or "DISABLE" on the line that corresponds to the axis you want and press F4, ENABLE, or F5, DISABLE.

c. Set the exec start ratio and the appropriate softness ratios to the desired values.

d. **To select another page of the detail screen** for other schedules, press F2, NUMBER.

10.46.4 Setting Up Cartesian Soft Float Schedules

You specify Soft Float schedules on the SETUP Soft Float screen. You can specify up to ten schedules for the Soft Float function.

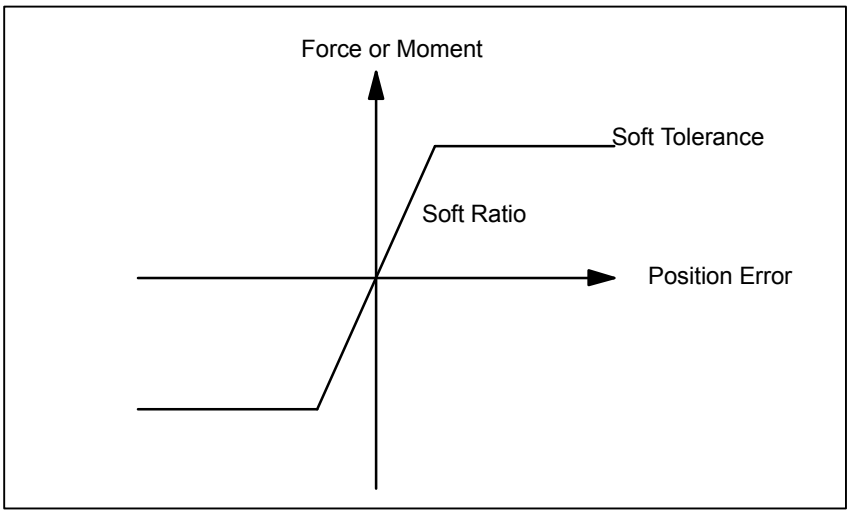
Table 10–96 lists and describes each item Soft Float schedule item on the SETUP Soft Float screen. Refer to the motion option section of Section 10.46.6 for more information on the exec start ratio.

Use Procedure 10-68 to set up Cartesian Soft Float schedules.

Table 10–96. Cartesian Soft Float Setup Items

ITEM	DESCRIPTION
Schedule No.	This item specifies the schedule number. By default, you can set up ten schedules.
Enable/disable	This item specifies whether to enable or disable the Cartesian Soft Float function. When set to DISABLE, Soft Float will not be executed.

Table 10–96. Cartesian Soft Float Setup Items (Cont'd)

ITEM	DESCRIPTION
Coordinate	This item specifies the frame used: World, User, or Tool
X direction Y direction Z direction X rotation Y rotation Z rotation	This item specifies the softness (flexibility) in the x, y, and z axis directions and rotations. When Soft Ratio is enlarged, the spring constant in the direction will become smaller. When Soft Tolerance is enlarged, the maximum of the force and the moment put out in the direction will be reduced. 

Procedure 10-68 Setting Up Cartesian Soft Float Schedules

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Soft Float. See the following screen for an example of the listing screen.

```
SETUP Soft Float
Group 1
 No    TYPE     START %       Comment
  1     CART      ***    [               ]
  2     CART      ***    [               ]
  3     CART      ***    [               ]
  4     CART      ***    [               ]
  5     CART      ***    [               ]
  6     CART      ***    [               ]
  7     CART      ***    [               ]
  8     CART      ***    [               ]
  9     CART      ***    [               ]
 10     CART      ***    [               ]
```

5. To set up schedules for Cartesian Soft Float,

a. Press F3, DETAIL, and then press F5, CART. See the following screen for an example.

```
Soft Float (CARTESIAN)
    Group 1
  1 Schedule No[   1]:[                    ]
  2 Enable Disable   : [DISABLE]
  3 Coordinate       : [WORLD]
                   Soft Rat   Soft Tol
  4 X direction    [  0]%     [  0]%
  5 Y direction    [  0]%     [  0]%
  6 Z direction    [  0]%     [  0]%
  7 X rotation     [  0]%     [  0]%
  8 y rotation     [  0]%     [  0]%
  9 Z rotation     [  0]%     [  0]%
```

b. **To enable or disable the Soft Float function** for a direction, move the cursor to the word "ENABLE" or "DISABLE" and press F4, ENABLE, or F5, DISABLE.

c. Set the coordinate system to the desired value.

d. Set the softness ratio and softness tolerance to the desired values.

e. **To select another page of the detail screen** for other schedules, press F2, NUMBER.

10.46.5 Setting Up Pushout Soft Float Schedules

You specify Soft Float schedules on the SETUP Soft Float screen. You can specify up to ten schedules for the Soft Float function.

Table 10–97 lists and describes each item Soft Float schedule item on the SETUP Soft Float screen. Refer to the motion option section of Section 10.46.6 for more information on the exec start ratio.

Use Procedure 10-69 to set up Pushout Soft Float schedules.

Table 10–97. Pushout Setup Items

ITEM	DESCRIPTION
Schedule No.	This item specifies the schedule number. By default, you can set up ten schedules.
Enable/Disable	This item specifies whether to enable or disable the Cartesian Soft Float function. When set to DISABLE, Soft Float will not be executed.

Table 10–97. Pushout Setup Items (Cont'd)

ITEM	DESCRIPTION
Coordinate	This item specifies the frame used: World, User, or Tool
Motion Type	Only one axis can be FREE in one group.
Soft direction	

Procedure 10-69 Setting Up Pushout Soft Float Schedules

Conditions

- You have set up payload information properly. Refer to Section 3.12 .

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Soft Float. See the following screen for an example of the listing screen.

```
SETUP Soft Float
Group 1
 No    TYPE     START %       Comment
  1     CART       ***    [               ]
  2     CART       ***    [               ]
  3     CART       ***    [               ]
  4     CART       ***    [               ]
  5     CART       ***    [               ]
  6     CART       ***    [               ]
  7     CART       ***    [               ]
  8     CART       ***    [               ]
  9     CART       ***    [               ]
 10     CART       ***    [               ]
```

5. Press F3, DETAIL.
6. Press >, NEXT until F4, PUSH is displayed.
7. Press F4, PUSH. You will see a screen similar to the following.

```
Soft Float(PUSHOUT)             JOINT  10%
                                       1/6
  Group1
  1 Schedule No.[10]:[****************]
  2 Enable/Disable:[DISABLE]
  3 Coordinate:[WORLD]
   Motion Type
  4 J1:[PUSH  ]  J2:[PUSH  ] J3:[PUSH  ]
  5 J4:[PUSH  ]  J5:[PUSH  ] J6:[PUSH  ]
  6 Soft direction:[None   ]
```

8. Set each item as desired.

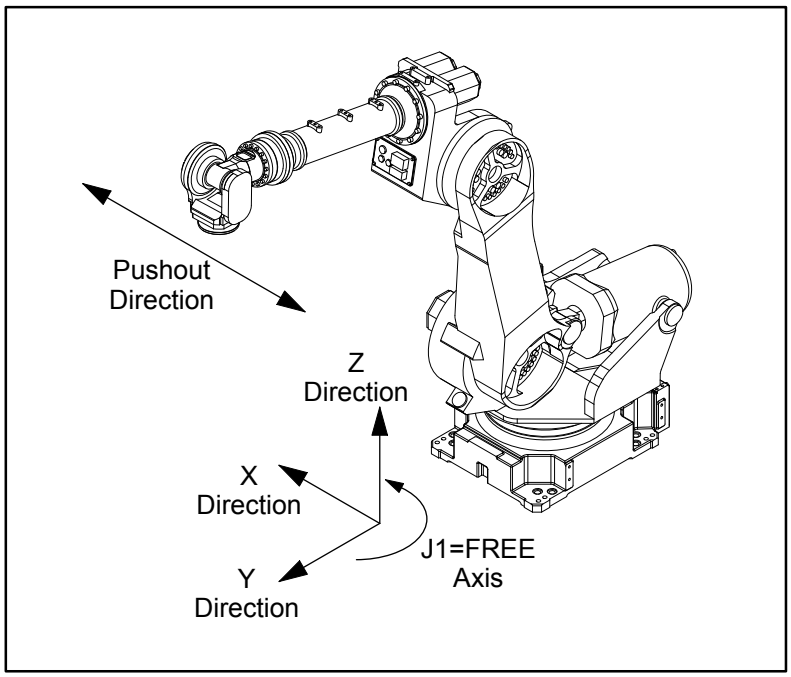
9. Set the motion type of the axis that should move passively following the force from outside by setting that axis to FREE. See the following screen for an example.

```
Soft Float(PUSHOUT)             JOINT  10%
                                       1/6
  Group1
  1 Schedule No.[10]:[****************]
  2 Enable/Disable:[DISABLE]
  3 Coordinate:[WORLD]
   Motion Type
  4 J1:[FREE  ]  J2:[PUSH  ] J3:[PUSH  ]
  5 J4:[PUSH  ]  J5:[PUSH  ] J6:[PUSH  ]
  6 Soft direction:[None   ]
```

10. Set the motion type of the axis which should move passively following the force from outside to FREE. Only one axis can be FREE in one group.

For example, to make the X direction of world coordinate soft when J1 is 90 degree, the motion type of J1 should be “FREE”. See Figure 10–165 .

Figure 10–165. Soft Float in the X Direction



10.46.6 Program Instructions

The following Soft Float instructions can be used:

- Soft Float[n]
- Soft Float END
- FOLLOW UP

Soft Float[n]

Soft Float[n] enables the Soft Float function from the next instruction in your teach pendant program until a Soft Float END instruction is executed. The Soft Float function enabled will use Soft Float schedule "n."

Refer to Section 10.46.3 to define Soft Float schedules.

Note You must set up payload information properly and specify the correct payload schedule in the program to use the Soft Float function. Accurate payload information is crucial in getting the desired result with the Soft Float function. Refer to Section 3.12 .

You can use Soft Float[n] in two forms within your teach pendant program. The form you use determines when during program execution the Soft Float function will be active. The Soft Float[n] instruction can be specified:

- As an **independent instruction**, on its own line within the program.
- As a **motion option** within a motion instruction.

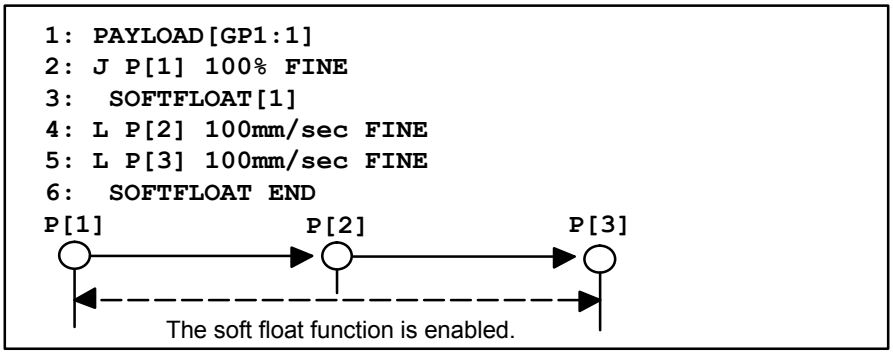
Independent Instruction

Soft Float[n]

When specified as an **independent instruction** on its own line within the program, the Soft Float function is enabled after the end of the motion specified on the line preceding the independent Soft Float[n] instruction.

In Figure 10–166 , the Soft Float function is enabled after the motion specified on line 1 completes, and disabled using Soft Float END on line 5. In this example, the FINE termination type in lines 4 and 5 has the same effect as CNT0.

Figure 10–166. Soft Float[n] Independent Instruction Example



Motion Option

L P[i] 100mm/sec FINE Soft Float[n]

When specified as a **motion option** within a motion instruction, the Soft Float function becomes enabled during the execution of the motion statement that contains the Soft Float[n] motion option.

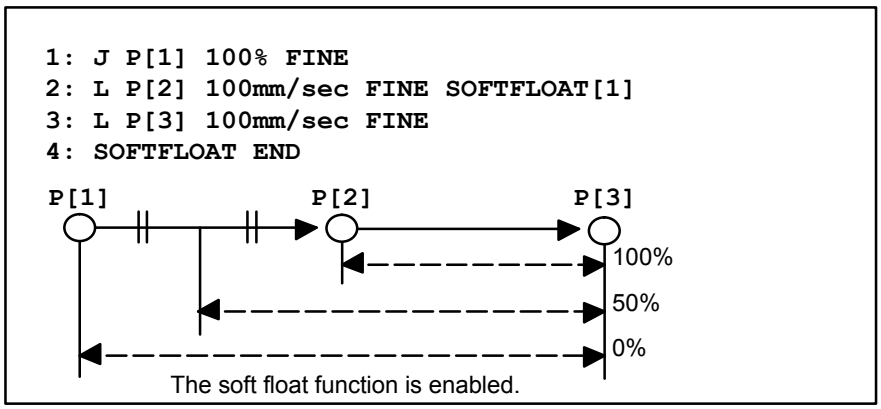
The point at which the Soft Float function becomes enabled is determined by a Soft Float schedule item, Exec Start Ratio. This is available only for JOINT Soft Float schedules.

The exec start ratio is specified as the percentage (from 0% to 100% in 1% steps) of a distance to be traveled before the robot reaches the taught point that corresponds to the motion instruction that contains the Soft Float[n] motion option.

In Figure 10–167 , the Soft Float function is effective between P[1] taught using the motion instruction on line 1, and P[2] taught using the motion instruction on line 2 that contains the Soft Float[n] motion option. In this example, the FINE termination type in line 3 has the same effect as CNT0.

- **Exec Start Ratio 100%** - The robot has completed 100% of the move to P[2] when Soft Float is enabled. This means that the robot is at P[2].
- **Exec Start Ratio 50%** - The robot has completed 50% of the move to P[2] when Soft Float is enabled. This means that the robot is halfway between P[1] and P[2].

Figure 10–167. Soft Float[n] Motion Option Example



Soft Float END

Soft Float END disables the Soft Float function. Add this instruction within your teach pendant program at the location where you want to stop using the Soft Float function.

FOLLOW UP

When an external force is removed from a robot, the robot usually tries to go back to the taught point. However, the FOLLOW UP instruction causes the robot to assume that the current position is the taught point, and prevents it from going back to the taught point. Use this instruction to override the robot's usual behavior when an external force is removed.

10.46.7 General Soft Float Guidelines

10.46.7.1 Overview

This function is NOT intended to be used when the **tooling is flexible, the fixture is not fixed, or the part is fragile (breakable)**. Also, CSF can be used when the space between part and fixture is around couple of millimeters. CSF is NOT intended to use for mating parts in tight tolerance.

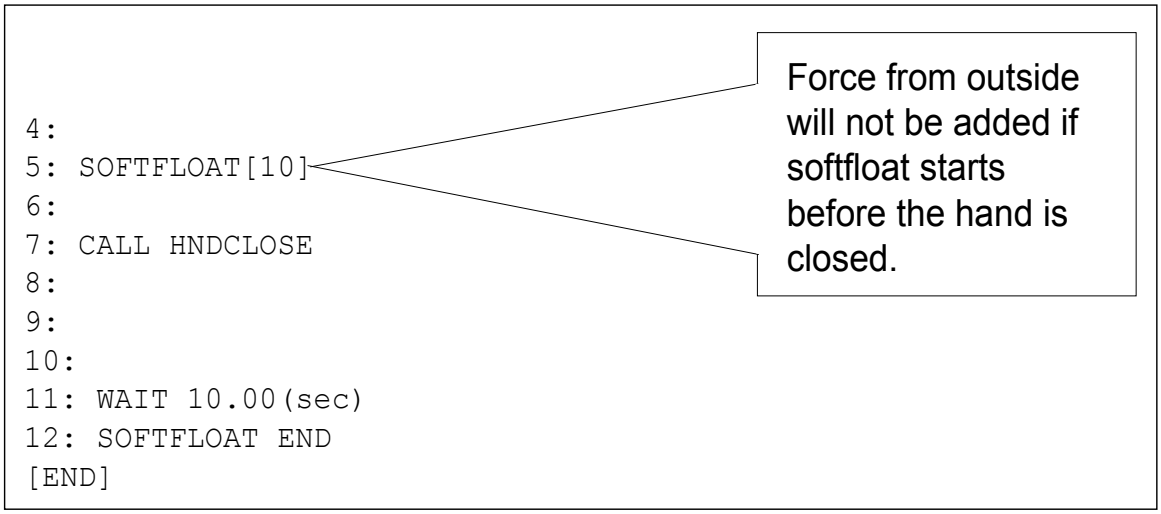
When Soft float is enabled, no external force can be applied to the robot. When Soft float is enabled, the servos will observe gravity by recording the motor torque at the instance and then change the rigidity of robot. If the external force is present at the moment when Soft Float is enabled, the servos will not be able to distinguish this external force from the gravity effect. This means that SRVO-023 Stop/Move Error Excess or SRVO-112 or SRVO-111 Soft Float Time Out might occur. For example in a pushout handling application, you should make a program in which the Soft Float starts before the hand is closed. See the following for an example.

- **Bad example**

```
4:
5:
6:
7: CALL HNDCLOSE
8:
9: SOFTFLOAT[10]
10:
11: WAIT 10.00(sec)
12: SOFTFLOAT END
[END]
```

Force from outside will be added if softfloat starts after the hand is closed.

- **Good example**

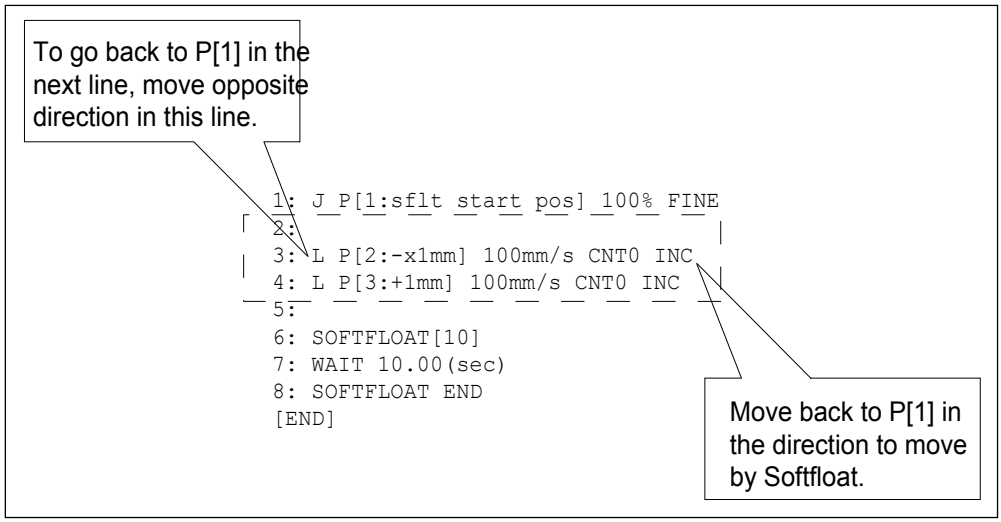


10.46.7.2 Improving Softness

Even with the same Soft Float schedule settings, the softness of Soft Float differs depending on the direction of the motion just before Soft Float starts. This is because Soft Float is more soft to move in the same direction to the motion just before and less soft to move in the opposite direction to the motion just before.

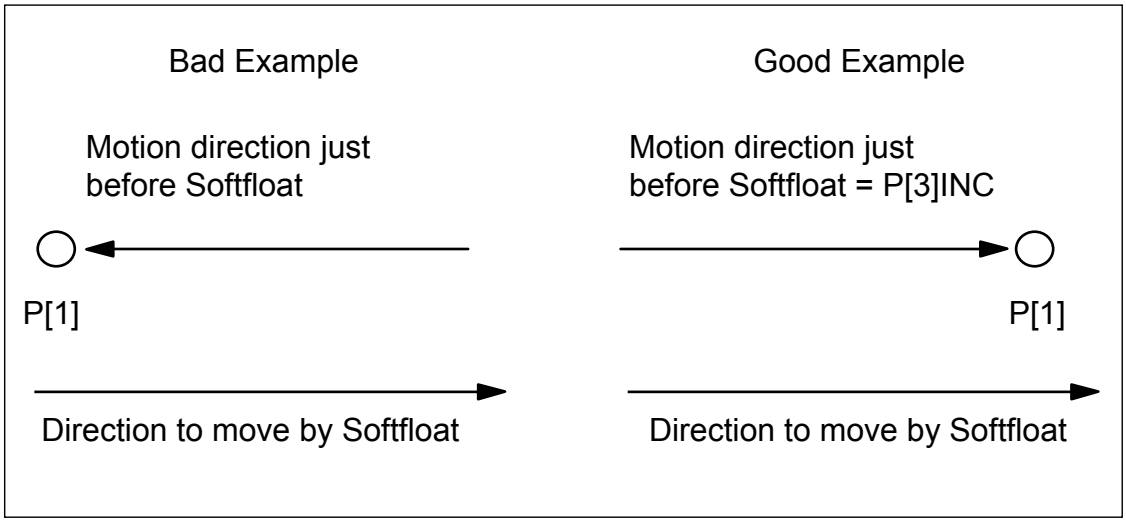
Therefore, make the program as that just before Soft Float starts and move the robot a little in the direction opposite to the direction the robot should move using Soft Float, then move the robot back to the position to start Soft Float. The distance necesary is 1–2mm or longer. For example, when you want to start Soft Float to move in +X direction in P[1], you should add the following motion instruction between the motion instruction to P[1] and the Soft Float instruction. See Figure 10–168 .

Figure 10–168. Improving Softness Example



By this method, the robot will become more soft to move in the +X direction and less soft to move in the -X direction. This is effective for use in Joint, Cartesian, and Pushout Soft Float.

Figure 10–169. Improving Softness



10.46.8 Cartesian Soft Float Guidelines

10.46.8.1 Overview

Cartesian Soft Float is supported both in the current Soft Float and in High Performance Soft Float. Softness along and around Cartesian coordinates can be specified.

10.46.8.2 General Cartesian Soft Float Guidelines

Layout and Tool Selection

The area of 10% – 20% inside from the outside border of mechanical unit operation area is recommended. Select the shortest tool possible.

TCP Settings

Use the actual tool frame setting.

Installation of robot and peripheral equipment/tooling selection

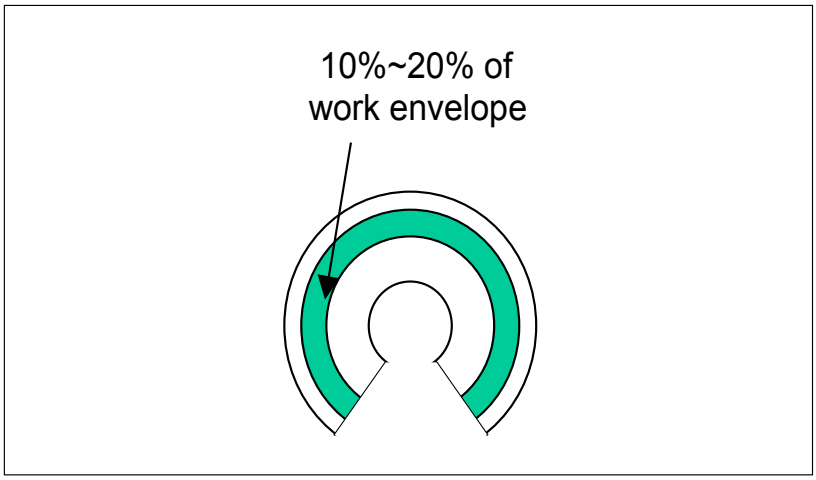
CSF has better performance if CSF is used within 10%~20% of the robot work envelope. Take this into account when you decide the robot installation and tooling selection.

Also, try to use short tooling if possible. The longer the tool, the more difficult it is to make it soft in XYZ directions.

The moment is needed for this re-orientation motion. select the correct tooling and set the proper softness in CSF orientation directions (WPR).

See Figure 10–170 .

Figure 10–170. Layout and Tool Selection



Schedule Setup

Set 100% in soft ratio of the direction or rotation that you want to make soft. Adjust Soft Tolerance in the pushing direction to limit the force and to avoid pushing with too large a force.

Velocity Teaching

In Soft Float with robot motion, the robot might not move if a small velocity is taught. Teach the velocity more than about 100 mm/sec.

Softness Settings

- **Insertion Motion with No Reorientation** See Figure 10–171 .

Figure 10–171. Insertion Motion

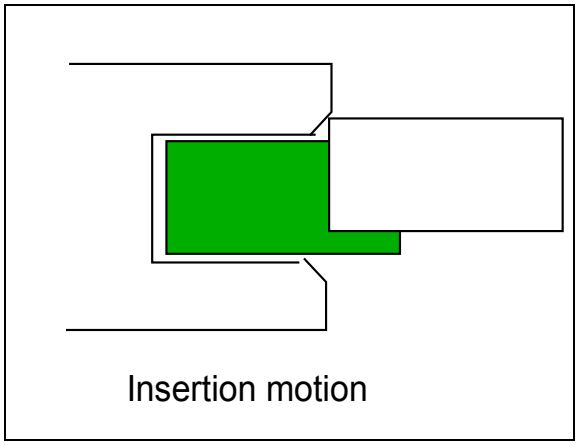


Table 10–98. Softness Settings

	Soft Ratio	Soft Tolerance	
X direction	[100 %	[0] %	XY directions are soft
Y direction	[100] %	[0] %	
Z direction	[0] %	[70] %	Pushing part in
X rotation	[0] %	[0] %	Orientation is fixed
Y rotation	[0] %	[0] %	
Z rotation	[0] %	[0] %	

Programming

If the robot is used to move or push the part against the fixture or objects when CSF is enabled, the programmed speed might not be reached easily due to the friction. Use 100 mm/sec programmed speed.

For example:

```
SOFTFLOAT[1]
L P[4]100 mm/sec FINE  insertion motion
```

TCP Settings

If Utool XYZ is set to 0 when CSF is enabled, the flexibility in WPR will be increased. For example:

```
UTOOL_NUM = 9   Utool schedule for XYZ = 0 (WPR no change)
SOFTFLOAT[1]
:
:
SOFTFLOAT END
UTOOL_NUM = 2   original utool setting
```

- **Insertion Motion with Reorientation** See Figure 10–172 .

Figure 10–172. Face-to-Face Motion

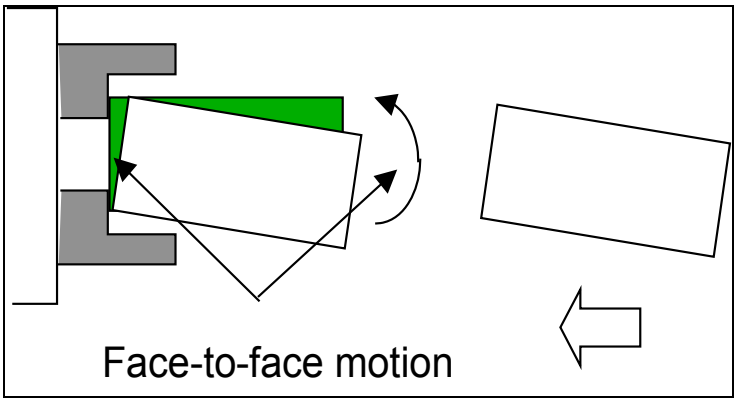


Table 10–99. Softness Settings

	Soft Ratio	Soft Tolerance	
X direction	[0] %	[0] %	Edge of part in XY directions is not soft
Y direction	[0] %	[0] %	
Z direction	[0] %	[70] %	Pushing part in
X rotation	[100] %	[0] %	Orientation of part should be very soft
Y rotation	[100] %	[0] %	
Z rotation	[100] %	[0] %	

Programming

If the robot is used to move/push part against the fixture or objects when CSF is enabled, the programmed speed may not be reached easily due to the friction. Use 100 mm/sec programmed speed. For example,

```
SOFTFLOAT[1]
L P[4]100 mm/sec FINE  insertion motion
```

10.46.8.3 Contouring Application Guidelines

Schedule Setup

For example, in the case of a push in the Z direction, refer to Table 10–100 .

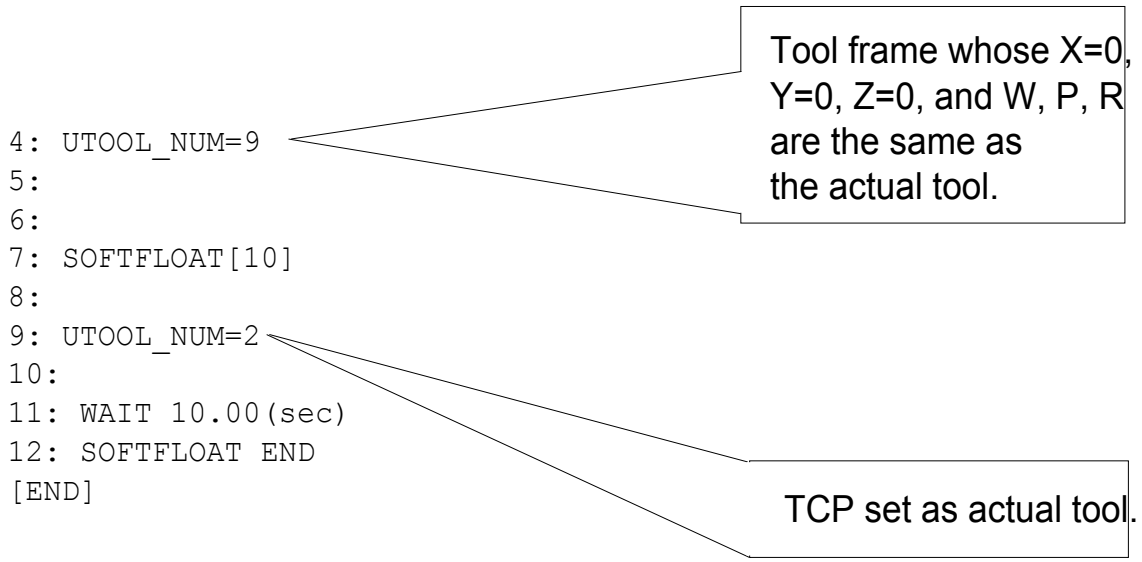
Table 10–100.

	Soft Ratio		Soft Tolerance		
X direction	100	%	0	%	Make X, Y direction soft.
Y direction	100	%	0	%	
Z direction	0	%	70	%	Limit force to avoid pushing with too large a force.
X rotation	0	%	0	%	Make rotation hard.
Y rotation	0	%	0	%	
Z rotation	0	%	0	%	

TCP Setup

In applications in which the tool direction must be kept, the TCP should be X, Y, and Z=0 when Soft Float starts. By this, difference between soft direction and hard direction appears well. See Figure 10–173 .

Figure 10–173. TCP Setup Example



10.46.8.4 Face Matching Application Guidelines

Schedule Setup

For example, in the case of a push in the Z direction, refer to Table 10–100 .

Table 10–101.

	Soft Ratio		Soft Tolerance		
X direction	0	%	0	%	Make X, Y direction soft.
Y direction	0	%	0	%	
Z direction	0	%	70	%	Limit force to avoid pushing with too large a force.
X rotation	100	%	0	%	Make rotation soft.
Y rotation	100	%	0	%	
Z rotation	100	%	0	%	

TCP Setup

Use the tool frame set as the actual tool. You do not have to change the TCP when Soft Float starts. Select the tool as short as possible. X, Y, and Z direction cannot become hard enough if the tool is too long. The moment of the direction to match the face is necessary. Select the tool that can generate moment of the direction to match the face by pushing even with the largest orientation mismatch.

10.46.9 Other Guidelines

10.46.9.1 Soft Float Setup Variables

When the Soft Float function is enabled, if follow-up processing requires more time than specified in the system variable $SFLT_FUPTIM, the servo alarm or program pause alarm occurs. The system variable $SFLT_ERRTYP specifies which alarm will occur.

$SFLT_FUPTIM

Default value: 1000 (ms)This value varies from one system to another. The large value that does not cause an alarm during normal operation should be used.

$SFLT_ERRTYP

Default value: 0-If 0, servo alarm "SRVO-111 Soft Float time out" occurs.-If 1, Program pause alarm "SRVO-112 Soft Float time out" occurs. (The alarm number is different between the alarms.)The default value should be used unless turning the servo off invites any inconvenience in the system.

Refer to the FOLLOW UP teach pendant instruction in Section 10.46.6 for more information on follow-up processing.

When the Soft Float function is enabled, follow-up processing is normally performed for individual motion instructions.This processing is enabled or disabled according to the system variable $SFLT_DISFUP.

$SFLT_DISFUP

Default value: FALSE

- If FALSE, follow-up is performed at the start of each motion instruction in the program.
- If TRUE, follow-up is not performed for individual motion instructions in the program.

10.46.9.2 True Position Variables

When Soft Float is enabled, the positional data that is displayed is the commanded robot position. The true robot position might be very different due to forces being applied to the robot. The following set of variables allows you to see the actual, machine pulse position of the robot. The machine pulse position is the current robot position (not the commanded position) based on the mechanical unit location, which is determined using encoder (machine) pulses. These variables will display a variety of useful values and are described below. See $SCR_GRP[n].$DPOS_DST Example for an example of how you can examine one of the variables to determine whether the robot TCP is within a given tolerance.

$SCR_GRP[ ].$m_pos_enb

This switch enables the current position function from machine pulse.

True = Enable

False = Disable

When it is enabled, Joint angle and x, y, z data of the TCP position are copied to $mch_ang [ ] and $mch_pos.

Note This variable must be set to true to make use of any of the other variables.

Note When $SCR_GRP[].$m_pos_enb = TRUE, Axis Limits will be enforced while in Soft Float mode. Refer to Section 10.46.9.3 .

$SCR_GRP[ ].$mch_ang[ ]

This variable displays the current joint position from machine pulse. When $M_POS_ENB = T, $mch_ang[ ] will be update continuously.

$SCR_GRP[ ]$mch_pos

This variable displays the current TCP x, y, z, w, p, and r position from machine pulse. When $m_pos_enb = T and program motion executes, it sets the current TCP position.

$SCR_GRP[ ].$dpos_dst

This variable displays the *distance to destination* position. When $m_pos_enb = T, the distance between the machine pulse position and the destination position of the program is set during programmed motion. Under normal conditions, this number will approach zero as it approaches the taught point. With Soft Float, this value might never reach zero and can be used to determine how close the TCP actually is to the taught position.

$SCR_GRP[n].$DPOS_DST Example

```
1:  LBL[1]
  2:  L PR[1] 400 mm/sec FINE Soft Float[1]
  3:  WAIT 1(sec)
  4:  ! Allow for any settling
  5:  IF $SCR_GRP[1].$DPOS_DST >= 100 , JMP LBL[2]
  6:  ! Compare distance with tolerance
  7:  L P[8] 400mm/sec FINE
  8:  Soft Float END
  9:  J P[5] 100% FINE
 10:  JMP LB[1]
 11:  LBL[2]
 12:  ! TCP is out of tolerance
```

```
13:  UALM[1]
```

10.46.9.3 Axis Limit Checking in Soft Float

While in Soft Float mode, the robot can move or be moved past the axis software limits by default. Program motion beyond axis limits is never allowed.

By setting $SCR_GRP[].$m_pos_enb = TRUE, the robot limits will be checked while in Soft Float mode. When the robot exceeds the software limits, the following alarm will be posted:

```
SRVO-116 Axis Limit Exceeded (G:x, A:y)
```

When this alarm occurs, the robot will have to be jogged back within axis limits prior to resuming.

10.47 SPACE CHECK FUNCTION (OPTION)

The space check function, incorporated into a robot, monitors a predetermined interference area (space). When another robot or a peripheral unit is located within that space, the function stops robot operation if it is being commanded to move into that space. See Figure 10–174 . The space check function releases the stop state and allows robot operation to continue only after checking that the other robot or peripheral unit has moved out of the area. See Figure 10–175 .

Figure 10–174. Space Check - Single Robot Controller

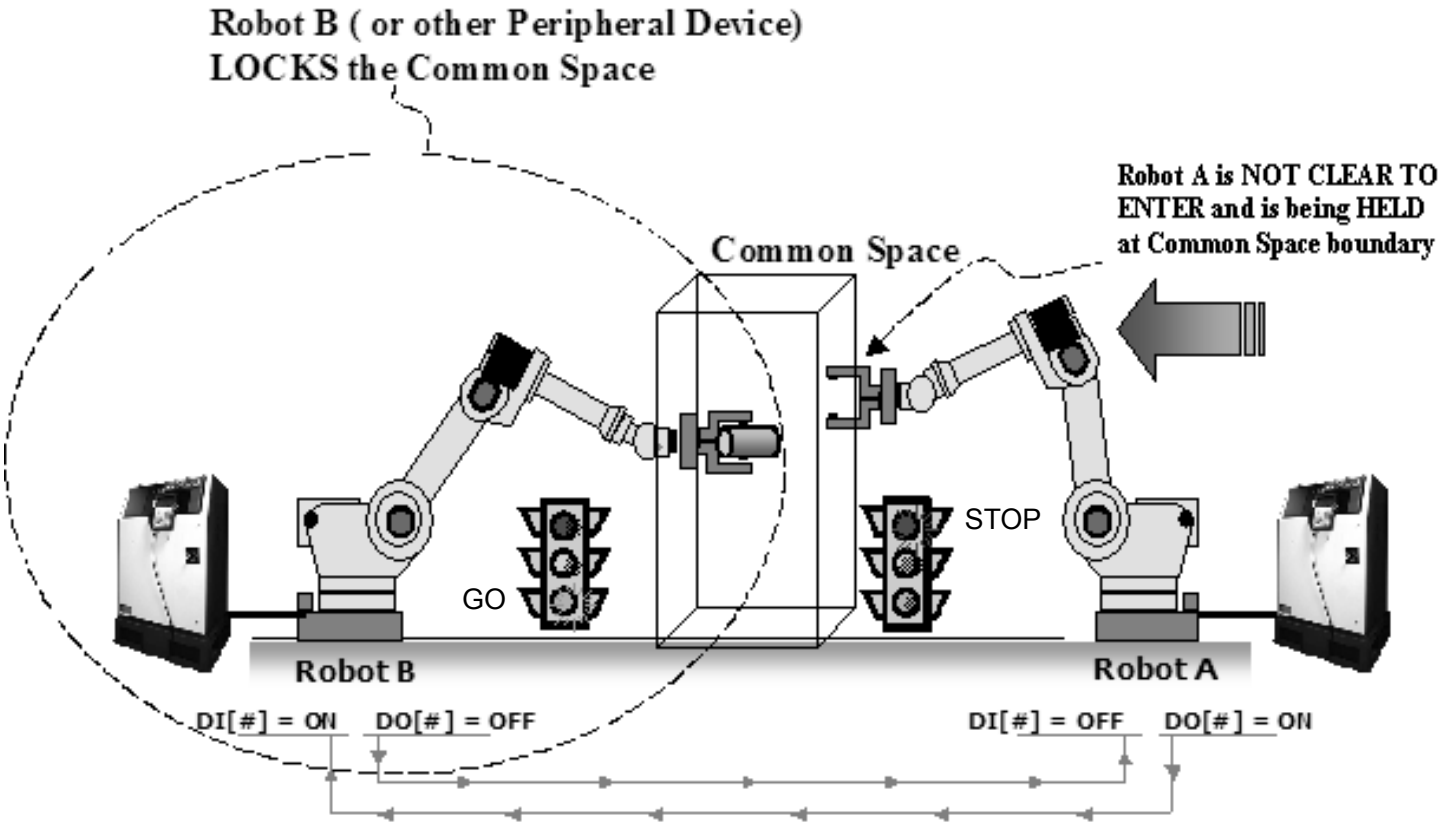
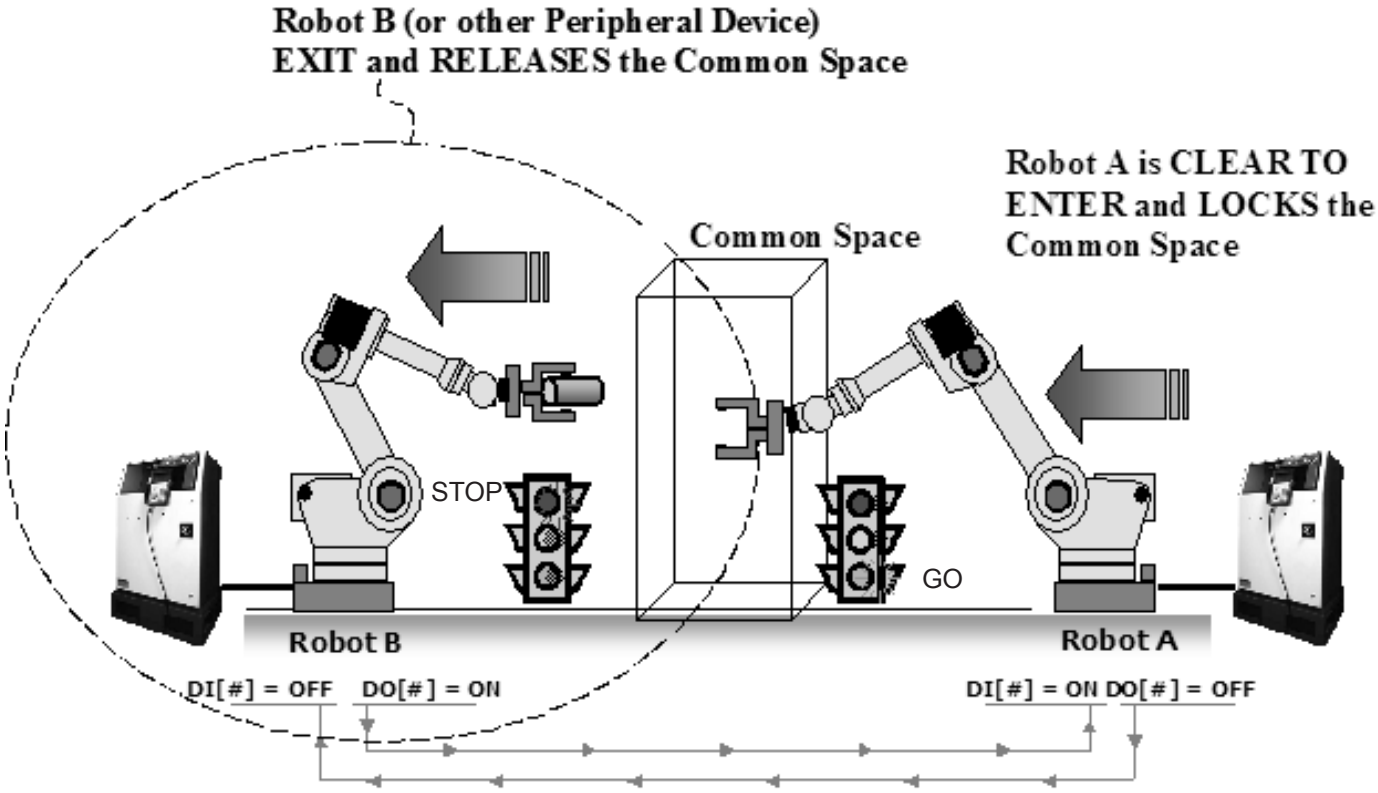


Figure 10–175. Space Check - Single Robot Controller



You can define up to three interference areas. If you need to define more than three interference areas, change the value of the system variable $NUM_RSPACE[1], turn off the controller, then turn it on again. Note that increasing the number of interference areas will increase processing time significantly.

Two interlock signals are assigned to a single interference area: one input and one output. These interlock signals are used for communication between the robot and another robot or a peripheral device.

Interlock Input Signal

When an attempt is made to move the robot into the interference area while the interlock input signal is OFF, the robot enters a hold state. If the robot is under program control, the program is still running, and the Busy Running lamp on the teach pendant remains ON. When the input signal goes ON, the hold state is released and the robot resumes automatic operation.

Warning

Since the robot starts decelerating as soon as the tool center point enters the interference area, the robot can stop at a point within the area. The faster the robot speed, the further within the area is the point where the robot stops. Taking this fact and the size of the tool into consideration, specify a larger interference area than what actually exists. Otherwise, you could injure personnel or damage equipment.

Override Interlock Input Signal Function

This function allows you to override the interlock input signal for a specified common space. After this function is enabled, the robot will never enter into a hold state even it moves into a common space while the interlock input signal is OFF or the interlock input is not specified. This function does not change the behavior of the interlock output signal.

You can enable or disable this function by setting a system variable $NO_USE_DI = TRUE or FALSE for each common space. You can also enable or disable this function On The Fly.

Warning

If you change $NO_USE_DI from TRUE to FALSE you must specify the interlock input. Otherwise, you could injure personnel or damage equipment.

Interlock Output Signal

When the tool center point enters the interference area, the interlock output signal goes OFF. When the tool center point is not located within the area, the signal is ON. Refer to Table 10–102 .

Table 10–103 lists and describes the items on the Space Fnct screens.

Table 10–102. Common Space I/O

Input (DI/RI)	Output (DO/RO)	Robot (which defines the Common Space)		Other Robot/Device (which signals the input)
OFF	OFF	Deadlock	Enters/Locks the Space	Locks the Space
OFF	ON	NOT Clear to Enter	Outside Space or HELD at boundary	Locks the Space
ON	OFF	Clear to Enter	Enters / Locks the Space	Releases the Space
ON	ON	Clear to Enter	Outside the Space	Releases the Space

10.47.1 Setting Up Space Check

You set up the space check function using the SETUP Space Fnct screens. Table 10–103 lists and describes the items on the Space Fnct screens.

Use Procedure 10-70 to set the conditions for the space check function.

Table 10–103. Space Check Function Screen Items

ITEM	DESCRIPTION
Rectangular Space DETAILED Screen Items	
Enable/Disable	This item enables and disables the space check function for that particular Common Space. To set or modify other conditions for an area, you must set the condition to Disable for that area.
Comment	This item allows you to specify up to ten characters as a comment.
Output Signal	This item allows you to specify the number of the interlock output signal.
Input Signal	This item allows you to specify the number of the interlock input signal. **Warning** **If the robot is running a program and is being held at a Common Space boundary, do NOT disable the Space unless after the robot EMERGENCY STOP button has been pressed or the robot is HELD by the HOLD signal. This will pause the program. Otherwise, robot motion is possible immediately after the Space is disabled.**

Table 10–103. Space Check Function Screen Items (Cont'd)

ITEM	DESCRIPTION
Priority High/Low	This item allows you to specify which of two robots has priority when both robots, which both use the space check function, attempt to enter the same area simultaneously. The robot for which Priority High has been specified can enter the area first. After the robot leaves the area, the other robot, for which Priority Low has been specified, is allowed to enter the area. A different setting must be made for each robot. **Warning** **If the same setting (Priority High or Priority Low) is specified for both robots and they both attempt to enter the interference area simultaneously, both enter a dead lock state. In this case, check that the correct priority has been specified for both robots and perform the following recovery:** **1. Apply EMERGENCY STOP to both robots. Note that if EMERGENCY STOP is not applied to both robots, as soon as one robot leaves the interference area, the other robot will start operating automatically. This is extremely dangerous. Never attempt this operation without applying EMERGENCY STOP to both robots.** **2. Check that the immediate vicinity of the robot is clear of personnel and equipment.** **3. Disable the space check function.** **4. Move one robot out of the interference area by jogging it.** **Otherwise, you could injure personnel or damage equipment.**
Inside/outside	This item allows you to specify whether the inside or outside of the rectangular box you define in space is used as the interference area.
Common Space Num	This item is currently not used.
SPACE SETUP Screen Items	
BASIS VERTEX	This item allows you to specify one of the corners of the rectangular space you define as a reference.
SIDE LENGTH/SECOND VERTEX	For SIDE LENGTH, use this item to specify the length of each side of the rectangular space, relative to the reference vertex along the x-, y-, and z-axes of the user coordinate system. Each side of the box must be parallel to an axis of the user coordinate system. For SECOND VERTEX, use this item to specify a vertex other than the reference vertex.

See Figure 10–176 . It is most typical to define a space as inside. Figure 10–177 is an example of an outside space.

The rectangular space that consists of the reference vertex and the specified vertex as its diagonal vertexes is the interference area. See Figure 10–178 .

Figure 10–176. Common Space Inside/Outside

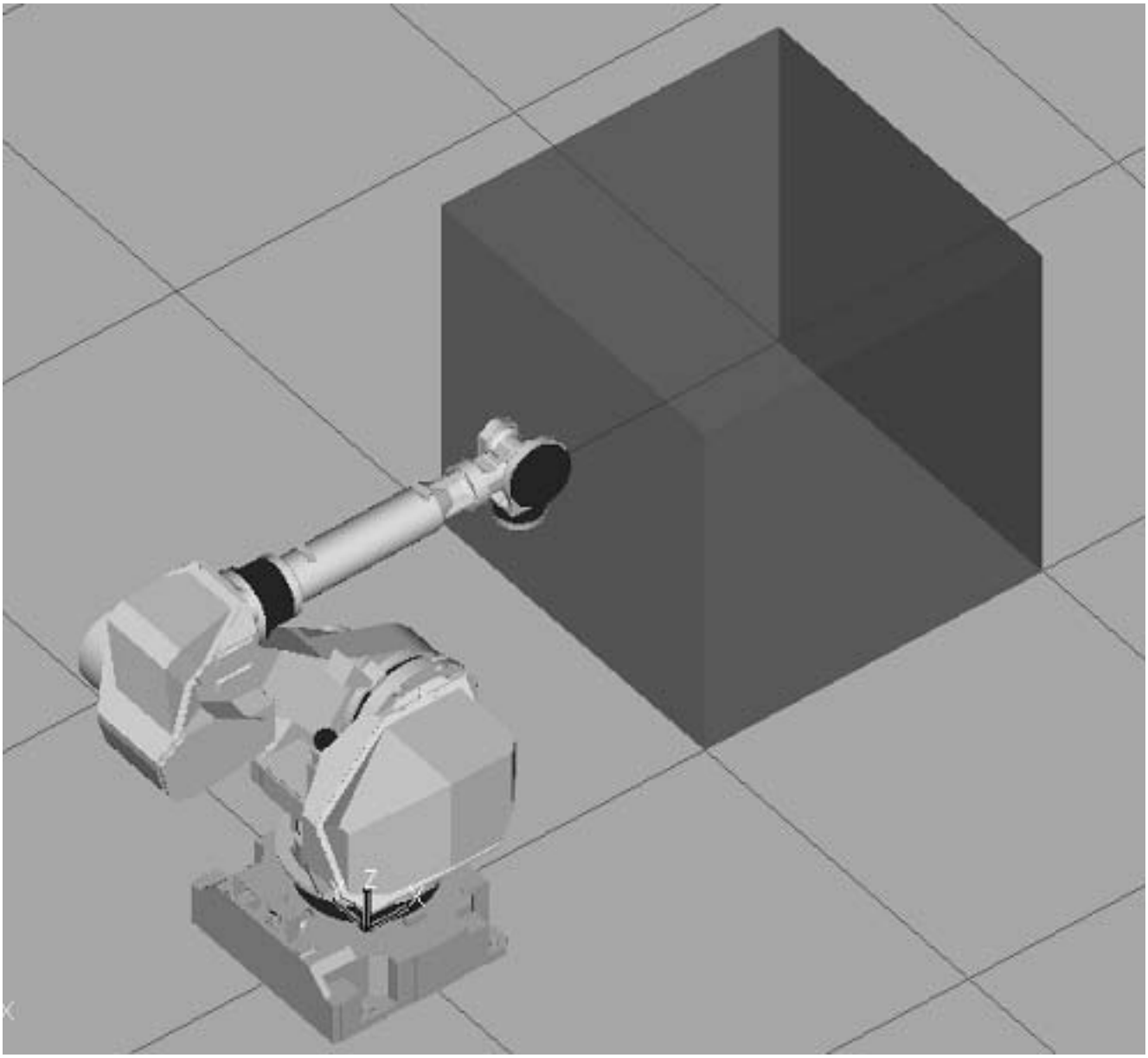
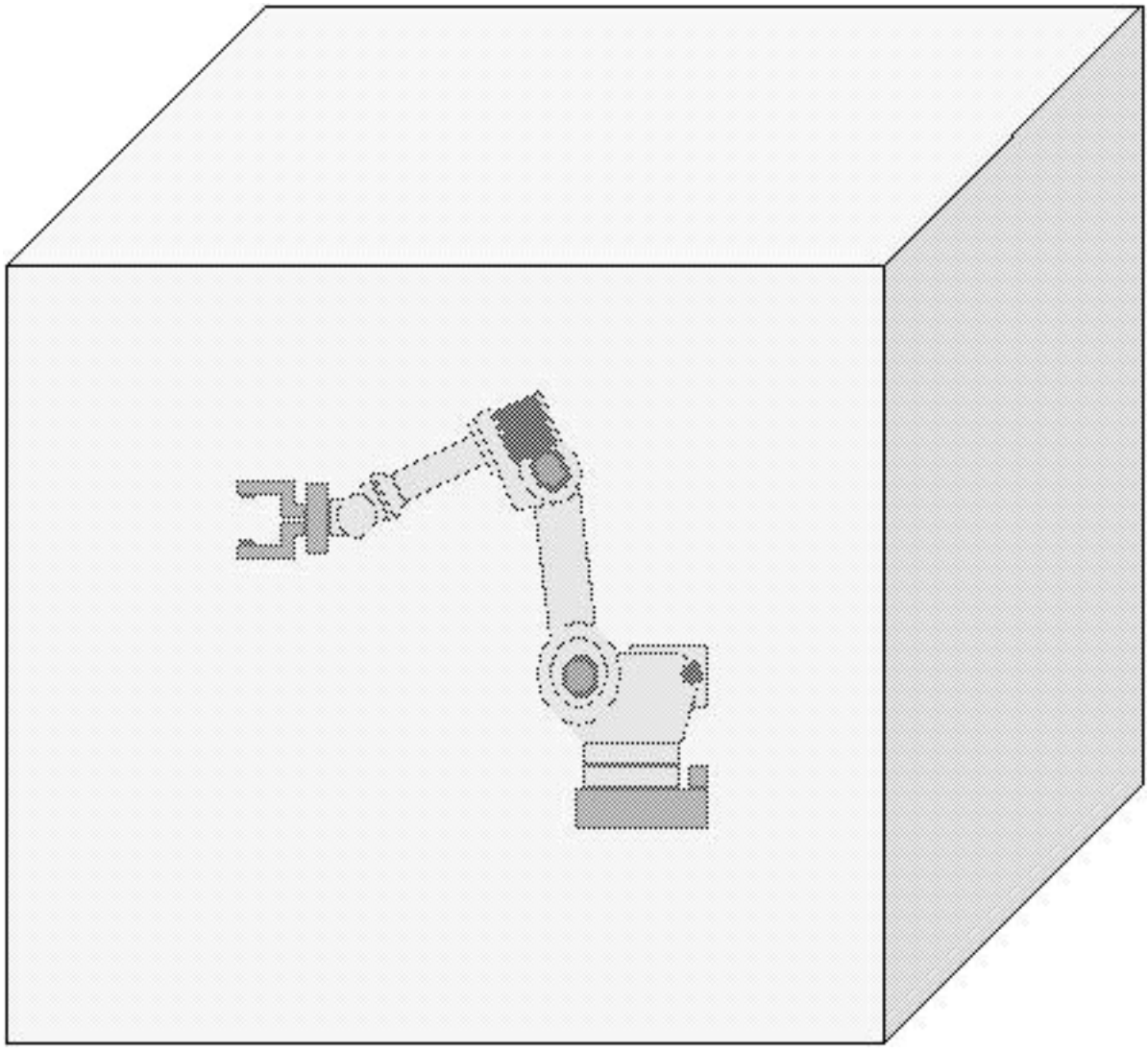


Table 10–104.

Common Space Setting	Definition	State of robot in Figure 10–178
Inside	The area inside the box is the Common Space	Robot is currently NOT in the Common Space
Outside	The area outside the box is the Common Space	Robot is currently IN the Common Space

Figure 10–177. Common Space Outside Boundary

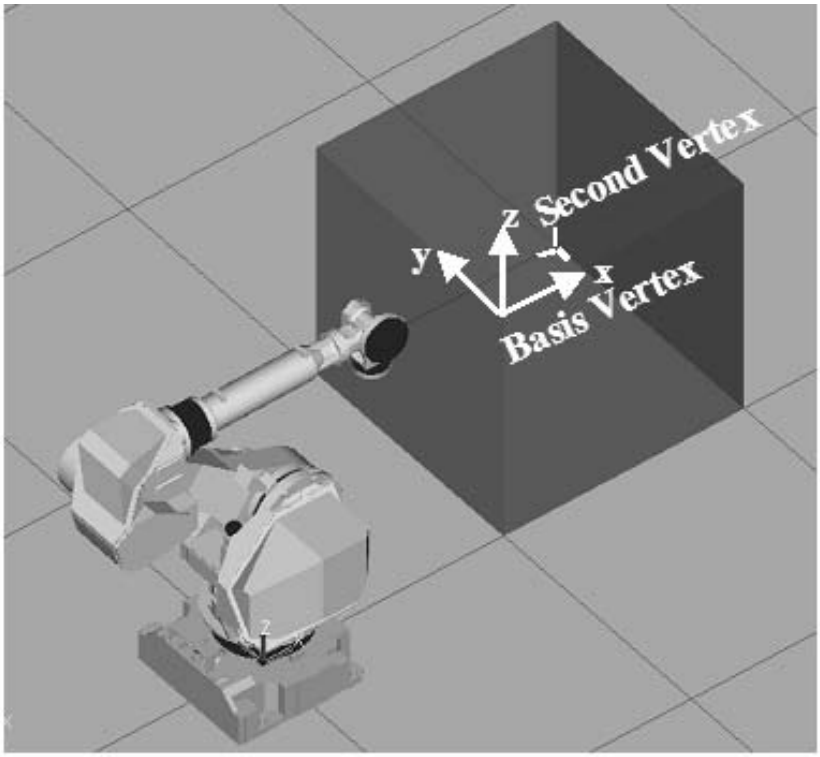


Note In Figure 10–177 ,

- If common space is defined as *outside*

- When enabled and if DI = OFF, the robot is not allowed to move outside the box (into common space for example)
- It will be held at the box boundary

Figure 10–178. Common Space Vertexes




```
SPACE SETUP                              1/4

     SPACE :1              GROUP :1

     UFRAME :0             UTOOL :1
  1 :  BASIS VERTEX        [SECOND\VERTEX]
  2 :X 1000.0    mm        2000.0    mm
  3 :Y 0.0       mm        1000.0    mm
  4 :Z 436.3     mm        -563.8    mm


[ TYPE ]  OTHER            [CHOICE] RECORD
```

```
SPACE SETUP                              1/4

     SPACE :1              GROUP :1

     UFRAME :0             UTOOL :1
  1 :  BASIS\VERTEX        [SIDE LENGTH  ]
  2 :X 1000.0    mm        1000.0    mm
  3 :Y 0.0       mm        1000.0    mm
  4 :Z 436.3     mm        -1000.0   mm


[ TYPE ]  OTHER                    RECORD
```

Procedure 10-70 Setting the Conditions for the Space Check Function

Steps

1. Press MENUS.
2. Select SETUP.
3. Press the F1, [TYPE].
4. Select Space fnct. The space list screen appears. See the following screen for an example.

```
Rectangular Space
LIST SCREEN
 No.Enb/Dsbl   Comment          Usage
  1  DISABLE[AG        ]Common Space
  2  DISABLE[          ]Common Space
  3  DISABLE[          ]Common Space
```

In a single robot controller, define the Common Space only for Group 1, which is the robot group. In a MultiARM controller, which has more than one robot group, press F2, GROUP, and type the group number to define the Common Space for that group. This allows each robot group to have its own Common Space defined. Each Space could then be used by each individual robot, each running its own independent program. See Figure 10–179 for an example of a MultiARM system using Space Check. In this example, memory map I/O is used. Refer also to Section 10.47.2 .

5. Move the cursor to the condition you want to set.
6. **To enable or disable a condition** ,
   - Press F4, ENABLE to enable it.
   - Press F5, DISABLE to disable it.
7. **To specify a comment** ,
   a. Move the cursor to the comment space and press ENTER.
   b. Select a method of naming the comment.
   c. Press the appropriate function keys to enter the comment.
   d. When you are finished, press ENTER.
8. **To define the Common Space**, press F3, DETAIL. See the following screen for an example.

```
Rectangular Space
DETAILED SCREEN
    SPACE :1              GROUP :1
    USAGE : Common Space
  1 Enable/Disable:     DISABLE
  2 Comment:         [        AG]
  3 Output Signal:       DO [  0]
  4 Input Signal:        DI [  0]
  5 Priority:            High
  6 Inside/outside:      Inside
  7 Common Space Num:       0
```

9. Move the cursor to the item you want to change. Use the function keys or numeric keys to enter the appropriate information.

10. To define the location and size of a space, press F2, SPACE. The space setting screen appears. See the following screen for an example.

```
Rectangular Space
SPACE SETUP
    SPACE :1               GROUP :1
    UFRAME :0              UTOOL :1
  1 :   BASIS\VERTEX       [SIDE LENGTH  ]
  2 :X 0.0      mm         0.0     mm
  3 :Y 0.0      mm         0.0     mm
  4 :Z 0.0      mm         0.0     mm
```

11. Specify the reference vertex (corner) and the length of each side or the diagonal vertex (corner) using one of the following methods. See Figure 10–178 .

Method 1: Move the cursor to X, Y, and Z, in turn, on the screen, then enter the appropriate coordinates using the numeric keys.

Method 2: Move the robot to the corner of the current rectangular box in space. Record the robot's current position: press and hold SHIFT and then press F5, RECORD.

12. **After you have specified the interference area** , press the PREV key to return to the DETAILED SCREEN. Press the PREV key again to return to the LIST SCREEN.

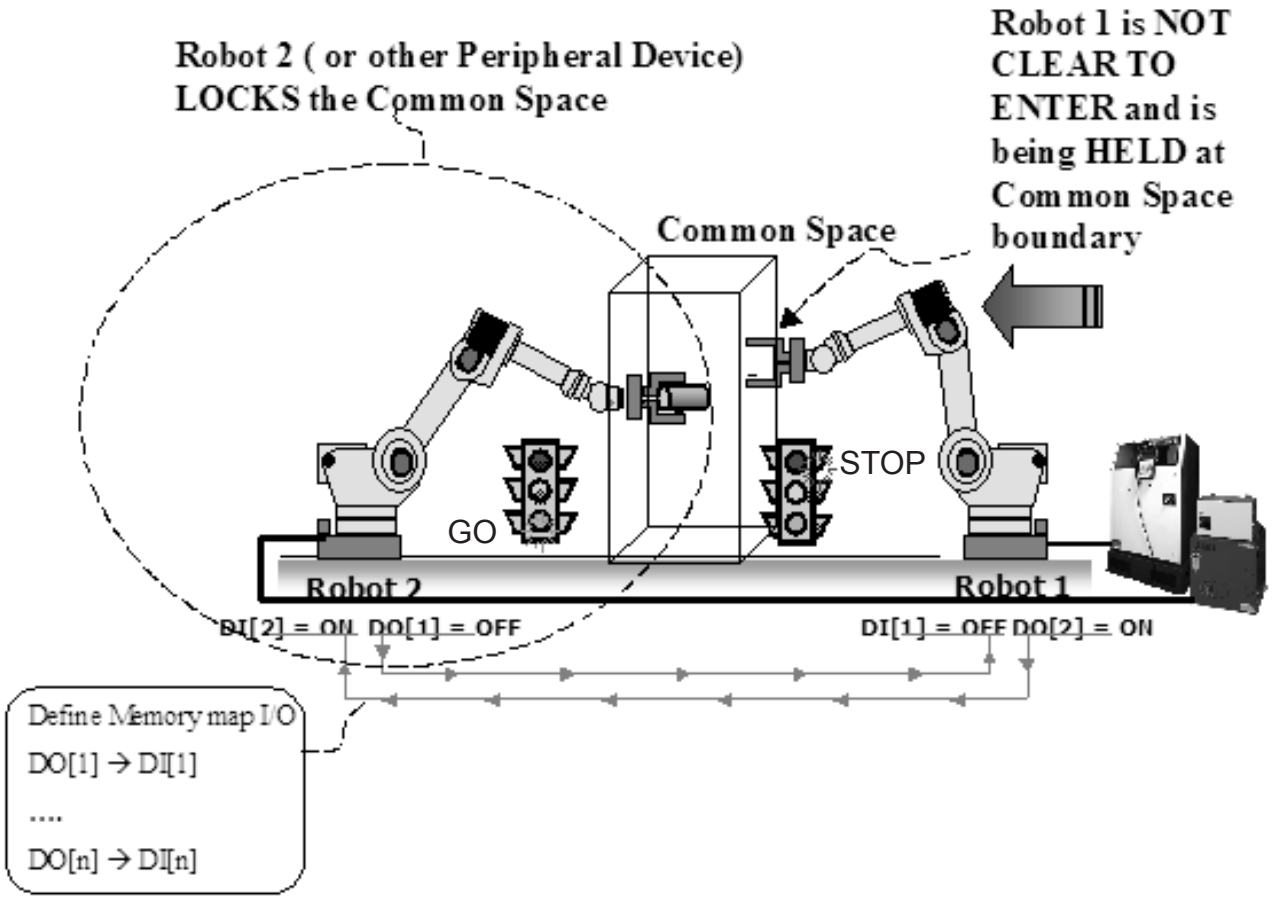
13. After you have defined the Common Space, cycle controller power as follows for the settings to take effect:

a. Press FCTN.

b. Select 0, NEXT.

c. Select Cycle Power.

Figure 10–179. MultiARM Controller



10.47.2 Limitations

Space Check currently does **not** support any tracking functions, any functions involving moving frames, Continuous Turn, and programs with multiple robot groups. Therefore, it is not compatible with:

- Line/Rail/Circular Tracking
- Robot Link
- Any other sensor tracking functions
- Coordinated Motion
- Dynamic remote Tool Center Point
- Simultaneous motion programs with more than one robot group

Therefore, do not load these functions with Space Check. If loaded, the common space defined will need to be disabled when the above functions are used. Otherwise, the SSPC-004 alarm CTV option not allowed, or other Space Check alarms could occur due to mis-detection. Common space could be disabled from either the user interface or by setting $RSPACE*g*[*n*].$ENABLED to FALSE, where *g* is the motion group number and *n* is the common space number for that group.

10.48 TCP SPEED OUTPUT (OPTION)

10.48.1 Overview

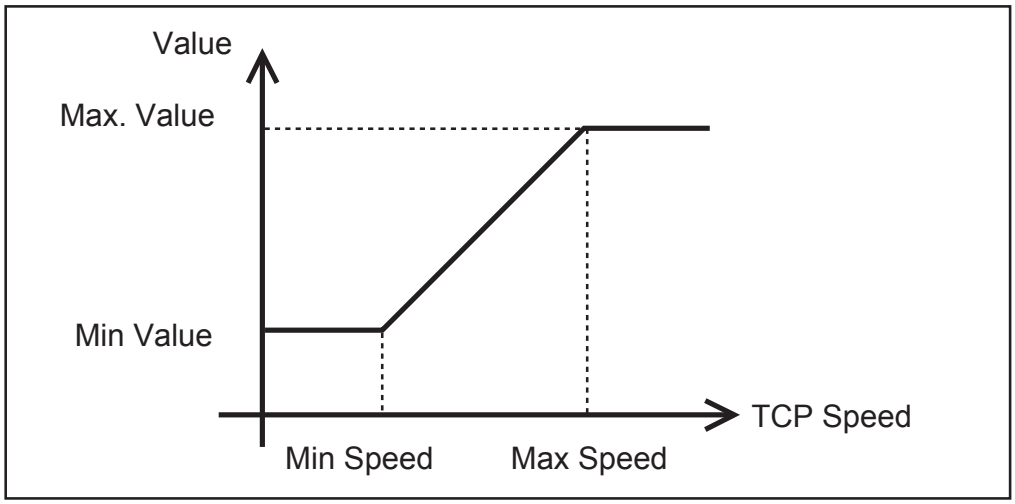
TCP speed output is a function that outputs a value proportional to the TCP speed to a *target* that can be an analog output, a group output, or a register.

The proportional rate is decided by four parameters: Max value, Max speed, Min value, and Min speed.

- When the TCP speed is less than Min speed, the value has Min value.
- When the TCP speed is greater than Max speed, the value has the Max value.

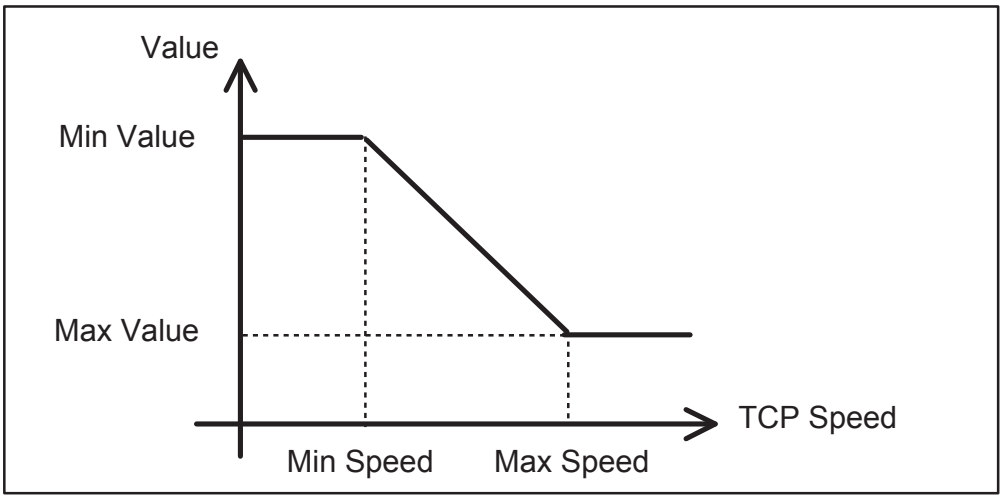
See Figure 10–180 .

Figure 10–180. Max Value



- The Max value can be less than the Min value. In this case the proportional rate is negative. See Figure 10–181 .

Figure 10–181. Max Speed



- When Max speed is less than Min speed, the value is always 0.

You can define up to 10 targets (outputs) to which the system will output the TCP speed proportional value.

- Every target can have individual proportional parameters.
- Every target can be disabled individually.
- When a target is disabled, the value is not set to the target.

The specified targets are set by the function **only** while a motion instruction is being executed. Otherwise the specified target values keep their respective Min values.

10.48.2 Setting Up TCP Speed Output

Use Procedure 10-71 to set up TCP Speed Output.

Procedure 10-71 Setting up TCP Speed Output

Steps

1. Press MENUS.
2. Select 5, I/O.
3. Press F1, TYPE
4. Select TCP Speed. You will see a screen similar to the following:

```
TCP speed output
List                                1/10
   Enable Target   Min (Val/Spd)      Max
  1 ON AO[       1] 100/ 10 4095/ 2000
  2 OFF AO[       2] 0/ 0 0/ 0
  3 OFF AO[       3] 0/ 0 0/ 0
  4 OFF AO[       4] 0/ 0 0/ 0
```

5. The TCP speed output List menu allows you to enable a target that has already been assigned.

The list menu shows the setting of all 10 targets. Min value, min speed, max value and max speed are all displayed as integer values. You can enable targets in this menu, but you cannot change other items.

To enable a target, move the cursor to the target you want and press F4, ON.

6. To set up details of a target, press F3, DETAIL. You will see a screen similar to the following:

```
TCP speed output
TCP_SPD[1]
  1 Enable:                    ON
  2 Target:                 AO[    1]
  3 Min. value:              10.000
  4 Max. value:            4096.000
  5 Min. speed (mm/sec):    100.000
  6 Max. speed (mm/sec):    2000.000
```

The Detail menu allows you to set all the parameters of a target. The min value, min speed, max value, and max speed are all displayed as real values.

You can change the type of the target to AO, GO, or R in this menu.

To return to the List menu, press F3, LIST, or press the PREV key.

7. To set prediction time, at the List menu press F2, SETUP. You will see a screen similar to the following:

```
TCP speed output
Setup
  1 Prediction time (msec): 0
```

The Setup menu allows you to set a TCP speed prediction time for all targets.

When you set a TCP speed prediction time, all targets will report what the TCP proportional speed will be at a specified future time, for example, 10ms. There is only one prediction time setting for the entire system. You cannot set individual prediction times for each target.

If you do **not** set a TCP speed prediction time, each target will report the actual TCP proportional speed as it occurs, according to the proportional parameters that were set for that target.

To return to the last menu, press F3, LIST, or press the PREV key.

10.48.3 TCP Speed Output Program Instruction

The TCP speed output function can be set up by TP program instructions. The TCP_SPD item is available under the Register and I/O instructions when Mixed logic is selected. The format of the instruction is the following.

```
target_type [ target_index ] =  ( TCP_SPD [ index ] )
```

- The ‘target_type’ is the type of the target: AO, GO, or R.
- The ‘target_index’ is the index of the target.
- The ‘TCP_SPD’ is the word used to enable the TCP speed output function.
- The ‘index’ is the line number of the target on the List menu.

When the following instruction is executed,‘Enable’ of TCP_SPD[1] will be turned on and ‘Target’ of TCP_SPD[1] will be set to AO[1].

```
AO[1]=(TCP_SPD[1])
```

Register and I/O instructions are also used to disable TCP speed output function. When the following instruction is executed,‘Enable’ of TCP_SPD[1] will be turned off and the output value of AO[1] will be forced to be 100.

```
AO[1]=100
```

TCP Speed Output Instructions

“TCP_SPD” is treated as mixed logic instructions. You can put TCP_SPD only in the area specified by parentheses.

Use Procedure 10-72 to add TCP Speed Output instructions to a teach pendant program.

Procedure 10-72 Adding TCP Speed Output Instructions to a Program

1. In the EDIT menu, press F1, INST.
2. Select Registers.
3. Select . . . = (. . .)
4. To add a Register Statement, select R[ ].

   If you want to add a GO Statement, select GO [ ], or to select an AO Statement, select AO [ ].

5. Enter the index number for the Register, GO, or AO.

```
Mixed logic                    JOINT  10 %
 1 UI[]              5 WI[]
 2 UO[]              6 WO[]
 3 RI[]              7 TCP_SPD[]
 4 RO[]              8 --- next page ---

   1   R[1]=(...)
 [END]
```

   Select TCP_SPD[ ].

```
Mixed logic                    JOINT  10 %
 1 UI[]              5 WI[]
 2 UO[]              6 WO[]
 3 RI[]              7 TCP_SPD[]
 4 RO[]              8 --- next page ---

   1   R[1]=(TCP_SPD[...])
 [END]
```

6. Enter 1 to assign the first target.

```
Mixed logic                    JOINT  10 %
 1 UI[]              5 WI[]
 2 UO[]              6 WO[]
 3 RI[]              7 TCP_SPD[]
 4 RO[]              8 --- next page ---

   1   R[1]=(TCP_SPD[1])
 [END]
```

10.49 TCP SPEED PREDICTION (OPTION)

10.49.1 Overview

The TCP Speed Prediction option provides the predicted value of the TCP speed continuously as the robot is moving. Some robot applications require that you know the Tool Center Point (TCP) speed of the robot ahead of time. This is especially useful for applications that dispense or spray material from a gun attached to the robot arm, where dispensing or spraying equipment cannot be turned on instantaneously. A delay, called the *equipment delay* , occurs before the flow of fluid becomes effective or before its rate, called the *flow rate* , changes. If the TCP speed is known ahead of time by the equipment delay time, the flow rate can be adjusted accordingly to increase or decrease depending upon the TCP speed. This provides better quality of dispensed material on the workpiece.

You must enable the TCP Speed Prediction option and set the equipment delay using the system variables described in this section. The TCP Speed Prediction software then computes the predicted value and sets it in another system variable. Your application can access these predicted values through teach pendant or KAREL programs and use them to control equipment precisely.



Caution

When you use TCP Speed Prediction, do not run a KAREL program that performs motion. For TCP Speed Prediction to be accurate, all motion must be initiated from a teach pendant program. Inaccurate speed prediction can cause pressure surges that damage the dispensing equipment.

Although there are a few system variables related to this option, the following are the most important:

- $TCPPIR.$ena ble_tcpp - enables or disables the TCP Speed Prediction option
- $TCPPIR.$tcd elay - sets the prediction or equipment delay time
- $TCPPSPEED.$ speed - this is where the output or the TCP predicted value is output continuously

You use the TCP Speed Prediction option by setting system variables that control how the option works. This section contains the following information:

- An **overview** of the TCP Speed Prediction option to help you decide when you can use the option - Section 10.49.2
- TCP Speed Prediction **system variables** that you set to the appropriate values to use TCP Speed Prediction - Section 10.49.3

10.49.2 Setting Up and Running TCP Speed Prediction

This section contains an overview of the TCP Speed Prediction option in the following areas:

- Software setup
- Execution
- Limitations

Software Setup

You must set several TCP Speed Prediction system variables properly in order to use TCP Speed Prediction. In particular, you must set the $enable_tcpp and $tcdelay fields in the $TCPPIR system variable structure properly. Refer to Section 10.49.3 for details.

Execution

After you have enabled TCP Speed Prediction using the system variables, TCP Speed Prediction runs automatically as a background task and does not require any additional user input or action. However, to take advantage of TCP Speed Prediction, an application task should be written that monitors the $TCPPSPEED system variable structure (typically the $speed field) for the predicted speed values, and performs whatever processing is required to generate application process control signals.

If the Remote TCP option is used, TCP Speed Prediction shifts its reference frame automatically as an application program switches to the Remote TCP mode and back again. Therefore, no user action is required. However, you must understand that the speed reference - the $TCPPSPEED.$vspeed and $TCPPSPEED.$speed variables in particular - are with respect to the Remote TCP frame instead of the WORLD frame during Remote TCP motions.

Limitations

The TCP Speed Prediction option has the following limitations:

- Prediction is supported only when all motion is initiated from a teach pendant program. TCP Speed Prediction will not be accurate if a KAREL program that performs motion is run.

- Prediction is supported only for group 1 motion. Multiple motion groups can exist and be programmed, but are not supported for TCP Speed Prediction.
- Prediction for non-integrated extended axes is not supported.
- Prediction for JOINT motions is approximated as if the motions were Cartesian motions. Such estimation is generally good enough if the JOINT motion is not too large. If predicted speed accuracy is important, linear motion is recommended.
- Prediction becomes invalid during instances of speed and motor limit conditions and other motion warnings and errors. In these cases, you could correct the programmed path to remove the error conditions. Prediction also becomes temporarily invalid during instances of dynamic speed override changes.
- You can change the prediction time delay ($TCPPIR.$tcdelay) on-the-fly. However, this might cause a momentary discontinuity in the value of the predicted speed output.
- TCP Speed Prediction has no knowledge of WAIT or other system delay instructions that might interrupt robot motion. Therefore, TCP Speed Prediction will continue to predict the speed based upon subsequent motion commands in these cases. However, TCP Speed Prediction will resynchronize itself properly after the delay condition has been met and motion resumes.
- TCP Speed Prediction is not compatible with any of the general tracking options, such as seam tracking and line tracking.
- Transitions between normal TCP Speed Prediction and Remote TCP Speed Prediction include a change of reference frame from WORLD frame to USER (REMOTE TCP) frame. When this transition takes place, the TCP Speed Prediction task will "blend" this change of reference frame. This might cause the predicted speed signal to show an apparent speed decrease during the blending period.

10.49.3 System Variables

Three system variable structures are generated when you load the TCP Speed Prediction option. These structures control the state and configuration of this option, and provide the speed and other output values for user program access. These variables are described in this section:

- $TCPPIR structure
- $TCPPSPEED structure
- $TCPP_CFG structure

$TCPPIR STRUCTURE

Name: TCP Speed Prediction Instruction Record System Variable Structure

Description: This set of variables allows the user to enable the TCP Speed Prediction softpart and set the equipment delay (prediction) time. The individual fields within this structure are defined below.

User Interface Location: SYSTEM Variables screen

$TCPPIR.$enable_tcpp

Minimum: FALSE

Default: FALSE

Maximum: TRUE

KCL/Data: RW

Program: RW

GET/SET_VAR: RW

Data Type: BOOLEAN

Name: TCP Speed Prediction Enable Switch

Description: This allows you to enable or disable the TCP Speed Prediction softpart. A predicted speed output is generated when the softpart is enabled. This variable is initially set FALSE at the time of loading, and must be set TRUE by the user prior to using the speed prediction values.

Power Up: This value initializes to FALSE and maintains its previous value over subsequent power cycles.

User Interface Location: SYSTEM Variables screen

$TCPPIR.$tcdelay

Minimum: -5000.0

Default: 200.0

Maximum: 5000.0

KCL/Data: RW

Program: RW

GET/SET_VAR: RW

Data Type: REAL

Name: TCP Speed Prediction (Equipment) Delay Time (milliseconds)

Description: This value allows the user to specify the equipment delay time to be used as the prediction time by the TCP Speed Prediction softpart. This variable is initially set to 200 at the time of loading, but should be set to an appropriate positive value by the user prior to using the speed prediction values.

Power Up: This value initializes to 200 (msec) and maintains its previous value over subsequent power cycles.

User Interface Location: SYSTEM Variables screen

NOTE: Modified values of this variable will take effect immediately if $TCPP_CFG.$otf_tim_enb is TRUE, but will not take effect until all robot motion has stopped if $TCPP_CFG.$otf_tim_enb is FALSE.

$TCPPSPEED STRUCTURE

Name: TCP Speed Prediction Speed Output System Variable Structure

Description: These are the output system variables that involve the predicted =speed of the robot Tool Center Point (TCP). These variables are updated by the TCPP task softpart at the interval specified by the value of $TCPP_CFG.$tcpp_time. The individual fields within this structure are defined below.

User Interface Location: SYSTEM Variables screen

See Also: $TCPP_CFG.$tcpp_time

$TCPPSPEED.$accel

Minimum: 0 Default: 0

Maximum: 1000.0

KCL/Data: RO

Program: RO

GET/SET_VAR: RO

Data Type: REAL

Name: TCP Speed Prediction Acceleration (millimeters/second)

Description: This is the predicted acceleration of the robot TCP which corresponds to the value of the predicted speed ($TCPPSPEED.$speed). It is a directionless value, but its sign is valid (positive for increasing TCP speed, negative for decreasing TCP speed).

Power Up: This value initializes to 0 at each power cycle and is updated automatically by the TCPP task softpart at every TCPP cycle.

User Interface Location: SYSTEM Variables screen

See Also: $TCPPSPEED.$speed

$TCPPSPEED.$motype

Minimum: 6

Default: 6

Maximum: 8

KCL/Data: RO

Program: RO

GET/SET_VAR: RO

Data Type: INTEGER

Name: TCP Speed Prediction Programmed Motion Type

Description: This is the programmed motion type of the last active TCPP motion segment at the time of the most recent TCP Speed prediction update. The values are as follows:

- JOINT = 6
- LINEAR = 7
- CIRCULAR = 8

Power Up: This value initializes to 0 at each power cycle and is updated automatically by the TCPP task softpart at every TCPP cycle.

User Interface Location: SYSTEM Variables screen

$TCPPSPEED.$prog_sp eed

Minimum: 0.0

Default: 0.0

Maximum: 2000.0

KCL/Data: RO

Program: RO

GET/SET_VAR: RO

Data Type: REAL

Name: TCP Speed Prediction Programmed Speed

Description: This is the programmed speed of the last active TCPP motion segment at the time of the most recent TCP Speed prediction update.

Power Up: This value initializes to 0 at each power cycle and is updated automatically by the TCPP task softpart at every TCPP cycle.

User Interface Location: SYSTEM Variables screen

$TCPPSPEED.$speed

Minimum: 0.0

Default: 0.0

Maximum: 2000.0

KCL/Data: RO

Program: RO

GET/SET_VAR: RO

Data Type: REAL

Name: TCP Speed Prediction Speed (Absolute Velocity Magnitude) (millimeters / second)

Description: This is the predicted speed of the robot TCP computed as the absolute value of the magnitude of the predicted Cartesian velocity ($TCPPSPEED.$vspeed). It is referenced (without sign or direction) to the WORLD frame for normal prediction, and to the currently selected Remote TCP frame (also without sign or direction) for Remote TCP operation.

Power Up: This value initializes to 0 at each power cycle and is updated automatically by the TCPP task softpart at every TCPP cycle.

User Interface Location: SYSTEM Variables screen

See Also: $TCPPSPEED.$vspeed

$TCPPSPEED.$tcdelay _mon

Minimum: -5000.0

Default: 200.0

Maximum: 5000.0

KCL/Data: RO

Program: RO

GET/SET_VAR: RO

Data Type: REAL

Name: TCP Speed Prediction Delay Time Monitor (milliseconds)

Description: This is a copy of the TCPP equipment delay time ($TCPPIR.$tcdelay) which is currently being used as the prediction time within the TCPP softpart. This value is presented within this structure to provide you with a convenient monitor point. This variable is intended to be used as a reference to indicate to the application processing task which value of TCDELAY is being used. Refer to the $tcdelay field of the $TCPPIR system variable to set the prediction equipment delay time.

Power Up: This value initializes to 0 at each power cycle and is updated automatically by the TCPP task softpart at every TCPP cycle.

User Interface Location: SYSTEM Variables screen

See Also: $TCPPIR.$tcdelay

$TCPPSPEED.$timestamp

Minimum: 0.0

Default: 0.0

Maximum: Maximum signed INTEGER

KCL/Data: RO

Program: RO

GET/SET_VAR: RO

Data Type: INTEGER

Name: TCP Speed Prediction Timestamp (ROS Ticks)

Description: This is the timestamp (in ROS ticks) corresponding to the most recent time when the current values of the $TCPPSPEED data structure were computed and recorded.

Power Up: This value initializes to 0 at each power cycle and is updated automatically by the TCPP task softpart at every TCPP cycle.

User Interface Location: SYSTEM Variables screen

See Also: $TCPPSPEED

$TCPPSPEED.$vspeed

Minimum: -2000.0

Default: 0.0

Maximum: 2000.0

KCL/Data: RO

Program: RO

GET/SET_VAR: RO

Data Type: VECTOR of REAL

Name: TCP Speed Prediction Velocity Vector (millimeters / second)

Description: This system variable is the predicted speed of the TCP recorded as a vector with components x, y, and z. It is referenced to the WORLD frame for normal TCP operation, and to the currently selected Remote TCP frame for Remote TCP operation.

Power Up: These values initialize to 0.0 at each power cycle and are updated automatically by the TCPP task softpart at every TCPP cycle.

User Interface Location: SYSTEM Variables screen

$TCPP_CFG STRUCTURE

Name: TCP Speed Prediction Configuration System Variable Structure

Description: This set of variables controls the mode of operation of TCP Speed Prediction. The individual fields within this structure are defined below.

User Interface Location: SYSTEM Variables screen

$TCPP_CFG.$group_num

Minimum: 1

Default: 1

Maximum: Maximum number of motion groups

KCL/Data: RW

Program: RW

GET/SET_VAR: RW

Data Type: INTEGER

Name: TCP Speed Prediction Motion Group Number

Description: This system variable sets the number of the motion group for which the TCP Speed Prediction softpart will predict TCP speed. Currently this value is restricted to group 1 because TCP Speed Prediction is available only for group 1 motion.

Power Up: This value initializes to group 1 and will maintain its value over subsequent power cycles.

User Interface Location: SYSTEM Variables screen

$TCPP_CFG.$otf_tim_en b

Minimum: FALSE

Default: TRUE

Maximum: TRUE

KCL/Data: RW

Program: RW

GET/SET_VAR: RW

Data Type: BOOLEAN

Name: TCP Speed Prediction On-The-Fly delay time update enable.

Description: This system variable when true allows for on-the-fly updating of the TCPP delay time value ($TCPPIR.$tcdelay). Since on-the-fly changes to the TCPP delay time might result in predicted speed value discontinuities, this variable might be set to false which will prevent delay time changes until the robot has stopped moving at the end of a program.

Power Up: This value initializes to TRUE and maintains its previous value over subsequent power cycles.

User Interface Location: SYSTEM Variables screen

See Also: $TCPPIR.$tcdelay

$TCPP_CFG.$warning_en b

Minimum: FALSE

Default: TRUE

Maximum: TRUE

KCL/Data: RW

Program: RW

GET/SET_VAR: RW

Data Type: BOOLEAN

Name: TCP Speed Prediction Warning Message Enable

Description: This system variable when true allows the posting of warning level error messages. These are warnings only, not faults, which you might want not to be displayed. Currently these messages include the following: - TCPP-011 "Pred time skips first motion" - TCPP-018 "Begin Error Mode at line:nn" - TCPP-019 "Speed Ovrd Mode at line:nn"

Power Up: This value initializes to TRUE and maintains its previous value over subsequent power cycles.

User Interface Location: SYSTEM Variables screen

10.50 TEACH PENDANT SHIM

10.50.1 Overview

The teach pendant shim (TP Shim) utility allows you to adjust individual positions of a teach pendant by a small amount in either the x, y, or z direction. It also allows you to track those adjustments and recall shims already applied to that position using the History screen. You can shim positions in one group or positions in multiple groups. The teach pendant shim varies from the Program Adjust Utility in that Program Adjust moves multiple positions in a program while the shim moves only one. Also, Program Adjust does not allow you to track previous adjustments.

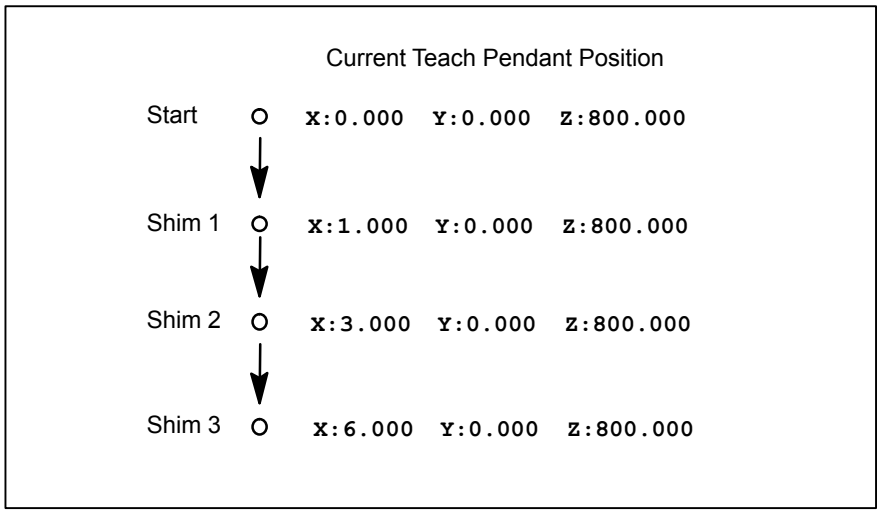
A TP Shim Example

Typically, you should shim positions in order, one after the other. This allows the shims to be additive and sequential. If you start off with the following position: X:0.000 Y:0.000 Z:800.000 Shims: 0/0 and you apply a shim of **1 mm in the x direction** , the position changes and the shim number and shim maximum increment by one. The new shimmed position is: **X:1.000** Y:0.000 Z:800.000 Shims: 1/1.

Apply another shim of**2 mm in the x direction**, the position changes, and the shim number and shim maximum increment by one. The new current position is now: **X:3.000** Y:0.000 Z:800.000 Shims: 2/2.

Apply another shim of**3 mm in the x direction**, the position changes, and shim number and shim maximum increment by one. The new current position is: **X:6.000** Y:0.000 Z:800.000 Shims: 3/3 See Figure 10–182 .

Figure 10–182. Typical TP Shim Example



An Advanced TP Shim Example Using USEMAST

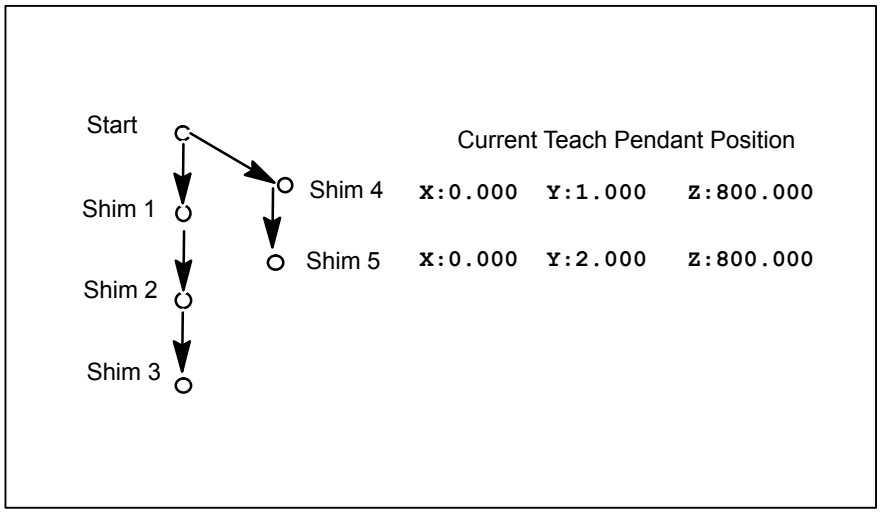
TP Shim allows you to shim from the original position using F3, USEMAST.

When you press F3, USEMAST, the original position is restored in the teach pendant program. At this point you can start shimming from that point without disturbing the previously applied shims.

For example,

1. Press F3, USEMAST, and F4, YES, to restore the original position. The new current position is: X:0 Y:0 Z:800 Shims: 0/3.
2. Apply a shim of**1 mm in the y direction**, the position changes, and the shim number and shim maximum increment by one. The new current position is: X:0.000 **Y:1.000** Z:800.000 Shims: 4/4.
3. Apply a shim of**1 mm in the y direction**, the position changes, and the shim number and shim maximum increment by one. The new current position is: X:0.000 **Y:2.000** Z:800.000 Shims: 5/5.

Figure 10–183. Advanced TP Shim Example Using USEMAST



An Advanced TP Shim Example Using HISTORY

You can also use the HISTORY screen to select a previously applied shim from which to start shimming positions. For example,

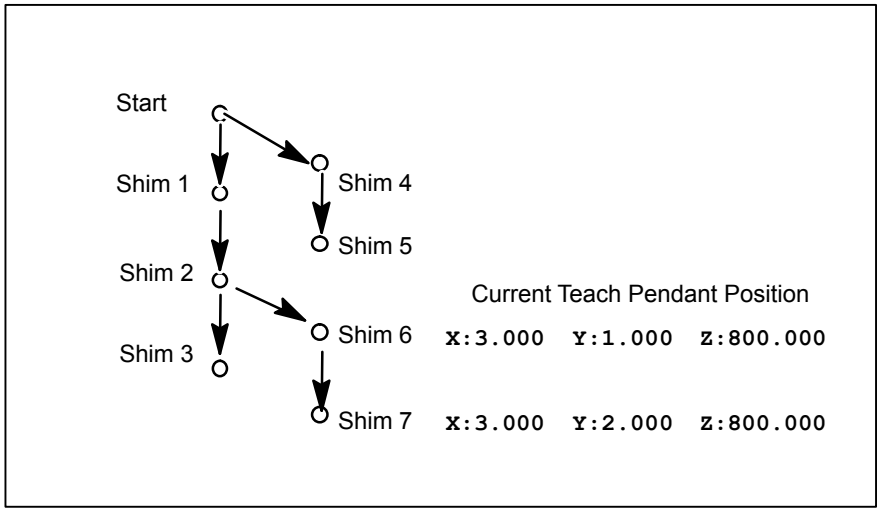
1. Press NEXT,>, then press F5, HIST, and select shim 2 and APPLY it. The new current position is: X:3.000 Y:0.000 Z:800 Shims: 2/5. See the following screen for an example.

```
Shim Utility
  Group 1: F-100      Program:FANUCTST
Position: 2:J P[2]    Shims: 2/5
  Units: MM
Shim: X: 2.000  Y:  0.000  Z:  0.000
  Total:
  X:   3.000    Y:  0.000  Z:  800.000
  Current:
  X:   3.000    Y:  0.000  Z:  800.000
  Master:
  X:   0.000    Y:  0.000  Z:  800.000
```

2. Display the SHIM screen.
3. Apply a shim of**1 mm in the y direction**. The position will change, and the shim number and shim maximum increment by one. The new current position is: X:3.000 **Y:1.000** Z:800.000 Shims: 6/6.

4. Apply another shim of**1 mm in they direction**, the position changes, and the shim number and shim max increment by one. The new current position is: X:3.000 **Y:2.000** Z:800.000 Shims: 7/7. See Figure 10–184 .

Figure 10–184. Advanced TP Shim Example Using HISTORY



10.50.2 Setting Up and Using TP Shim

You must select the program and instruction you are going to shim before you can apply a shim. Refer to Table 10–105 for more information on the items you must set up to use TP Shim. Use Procedure 10-73 to shim a position.

Note You cannot shim an incremental teach pendant position, or a position register.

Table 10–105. Teach Pendant Shim Screen Items and Function Keys

ITEM	DESCRIPTION
Group	This item indicates the number of the current robot group, or ALL if the shim will be applied to multiple groups.
Program	This item indicates the name of the program to which the shim will be applied.
Position #	This item indicates the current position that will be shimmed.
Shim# default: 10 min: 1 max: 100	This item indicates the number of shims that apply to the current position.

Table 10–105. Teach Pendant Shim Screen Items and Function Keys (Cont'd)

ITEM	DESCRIPTION
Units units: mm or inches	This item indicates the unit of measurement that is displayed for each shim.
Shim default: 9.9 mm min: 00 range: +/-9.9	This item indicates the value, in x, y, and z, that the shim will be adjusted. This item cannot be changed from this screen. You must use the HISTORY screen to change this item.
CURRENT	This item indicates the current x, y, and z value of the actual position. After you press F2, APPLY, this value will change to reflect the shim amount.
MASTER	This item indicates the original x, y, and z value of the position before any shims were applied.
APPLY [F2]	This key applies the shim to the current position.
USEMAST [F3]	The key restores the original taught point. The $FX_DELSHM_EN system variable default is TRUE. When you press F3, USEMAST all the shim data is cleared. If you set $FX_DELSHM_EN to FALSE, the shim data will stay at "Where"?
CHOICE [F4]	This key allows you to select groups and programs.
HELP [F5]	This key displays the HELP text.
NEXT	This key displays the next page of the function keys.

Procedure 10-73 Shimming a Teach Pendant Position

Conditions

- The system variable $FX_TPSH_ENB is TRUE (default).
- The program to be shimmed has been created on the controller.

Steps

1. Select the program that contains the positions you want to modify.

Note You can start shimming positions from within a teach pendant program or you can select the program after you have started the TP Shim utility.

a. Press MENUS.

b. Select UTILITIES and press ENTER.

c. Press F1, [TYPE].

d. Select TP SHIM UTIL.

e. **If the program you want to shim is not currently selected or displayed,** move the cursor to Program and press F4, [CHOICE]. A list of all the available teach pendant programs will be displayed. Select the program you want to shim.

f. If you have a multi-group system, move the cursor to GROUP and press F4, [CHOICE], or press ENTER.

g. Select the group or groups you want to modify and go to Step 2 .

2. Move the cursor to Position and type the position number you want to shim. You will see a screen similar to the following.

```
Shim Utility
  Group 1: F-100     Program:FANUCTST
Position: 1:J P[1]     Shims: 0/0
  Units: MM
Shim: X:  0.000  Y:  0.000  Z:  0.000
  Current:
  X:      0.000  Y:  0.000  Z:  800.000
  Master:
  X:*********  Y:*********  Z:*********
```

3. Move the cursor to the x, y, and z values of Shim individually and type the shim value for x, y, or z that you want to apply to the current position. For example, if you want to shim the z component of the current position by .700 mm, you will see a screen similar to the following.

```
Shim Utility
  Group 1: F-100     Program:FANUCTST
Position: 1:J P[1]     Shims: 0/0
  Units: MM
Shim: X:  0.000  Y:  0.000  Z:  0.700
  Current:
  X:      0.000  Y:  0.000  Z:  800.000
  Master:
  X:*********  Y:*********  Z:*********
```

4. To apply the requested shim to the displayed position, press F2, APPLY. You will see a screen similar to the following.

```
Shim Utility
  Group 1: F-100     Program:FANUCTST
Position: 1:J P[1]     Shims: 1/1
  Units: MM
Shim: X:  0.000  Y:  0.000  Z:  0.000
  Current:
  X:      0.000  Y:  0.000  Z:  800.700
  Master:
  X:      0.000  Y:  0.000  Z:  800.000
```

Note After you apply a shim, the displayed shim values for x, y, and z will return to 0.

Note The shim move will be instituted the next time the program is run.

5. **To accept the master position as the current position and back out of the current shim(s),** press F3, USEMAST.
6. **To select another program** , move the cursor to Program, and press F4, [CHOICE].
7. **To select another position within the current program**,
   a. Move the cursor to Position and enter the position you want to shim.
8. **To change the shim display between millimeters and inches**, move the cursor to Units and press F4, MM, to display millimeters and F5, INCHES, to display inches.

10.50.3 Using TP Shim History to Apply Previous Shims

You can apply previous shims to the current position by using the TP Shim History screen. Use Procedure 10-74 to use the HISTORY screen to recall a shim and apply it to a teach pendant position.

Refer to Table 10–105 for a description of each of the TP Shim setup items displayed on the HISTORY screen. Refer to Table 10–106 for a description of the items displayed on the HISTORY screen that are different from those on the TP Shim Screen.

Table 10–106. HISTORY Screen Items

ITEM	DESCRIPTION
Shim default: 9.9 mm min: 00 range: +/-9.9	This item indicates the value, in x, y, and z, that the shim will be adjusted. This item can be adjusted to display a previously recorded shim.
Total	This item indicates the total number of shim values applied to the original position. After you press F2, APPLY, the Current value becomes the same as Total.

Procedure 10-74 Shimming a Teach Pendant Position Using HISTORY

Conditions

- The system variable $FX_TPSH_ENB is TRUE (default).
- The program to be shimmed has been created on the controller.

Steps

1. Select the program that contains the positions you want to modify.
   a. Press MENUS.
   b. Select UTILITIES and press ENTER.
   c. Press F1, [TYPE].
   d. Select TP SHIM UTIL.
   e. **If the program you want to shim is not currently selected or displayed,** move the cursor to Program and press F4, [CHOICE]. A list of all the available teach pendant programs will be displayed. Select the program you want to shim.
   f. If you have a multi-group system, move the cursor to GROUP and press F4, [CHOICE], or press ENTER.
   g. Select the group or groups you want to modify and go to Step 2 .
2. Move the cursor to Position and type the number of the position you want to shim using the HISTORY Screen. You will see a screen similar to the following.

```
Shim Utility
  Group 1: F-100     Program:FANUCTST
Position: 2:J P[2]     Shims: 2/5
  Units: MM
Shim: X:  2.000  Y:  0.000  Z:  0.000
  Total:
  X:      3.000  Y:  0.000  Z:  800.000
  Current:
  X:      0.000  Y:  2.000  Z:  800.000
  Master:
  X:      0.000  Y:  0.000  Z:  800.000
```

3. **To display the next or previous shim for the current position**,
   a. Display the history screen by pressing NEXT, F5, HIST.
   b. **To display the next shim**, move the cursor to Shim and press F5, NXSHM.
   c. **To display the previous shim**, move the cursor to Shim and press F4, PRSHM.
4. **To apply the requested shim to the displayed position,** press F2, APPLY. You will see a screen similar to the following.

```
Shim Utility
  Group 1: F-100     Program:FANUCTST
Position: 2:J P[2]     Shims: 2/5
  Units: MM
Shim: X:  2.000  Y:  0.000  Z:  0.000
  Total:
  X:      3.000  Y:  0.000  Z:  800.000
  Current:
  X:      3.000  Y:  0.000  Z:  800.000
  Master:
  X:      0.000  Y:  0.000  Z:  800.000
```

5. Press NEXT, > and then press F5, SHIM, to display the Shim Utility Screen. This allows you to add new shims.
6. **To accept the master position as the current position and back out of the current shim(s),** press F3, USEMAST.
7. **To select another program**, move the cursor to Program, and press F4, [CHOICE]. This will allow you to choose programs that you have already shimmed.

10.51 TIME BEFORE/AFTER MOTION OPTION INSTRUCTION

10.51.1 Overview

Normally, when a teach pendant program is executed, the instruction that follows a motion instruction is not executed until the motion has been completed. The TIME BEFORE/AFTER motion option instruction allows you to specify a teach pendant program that is to be called or an output that is to be set at a specified time before or after the completion of a motion instruction.

For example, you might specify that a teach pendant program CLS_GRIP is to be called 600 ms before the completion of the move. CLS_GRIP might consist of the instruction DOUT[GRIP]=ON.

This function can reduce external device communication time and improve cycle time. This section contains information on the following:

- Program execution
- Execution timing
- Recording a TIME BEFORE or TIME AFTER instruction
- TIME BEFORE instruction program example
- Programming

Note The Time Before/After Motion Option only exists if the Condition Monitor Option is loaded.

10.51.2 Program Execution

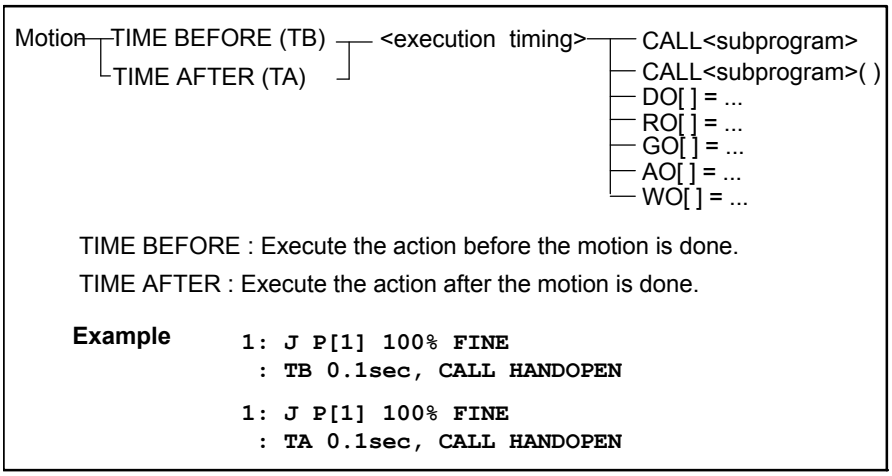
The motion instruction and the sub program (called by the main program) are executed in parallel. Because of this, the execution of the sub program does not affect the robot motion in the main program.

The instruction that follows a TIME BEFORE or TIME AFTER instruction will not be executed until the subprogram specified in the TIME BEFORE or TIME AFTER instruction has been executed.

You must specify the called program in the TIME BEFORE/AFTER instruction and specify the time when the CALL instruction is to be executed (*execution timing*). If the execution timing is 0 sec, this indicates that the robot has stopped moving. The exact time that the robot stops is determined by the termination type (FINE, CNT 100 and so forth).

The action and the execution timing are taught in the motion option instruction. See Figure 10–185 .

Figure 10–185. TIME BEFORE / TIME AFTER Motion Option Instructions



Robot TCP Path

With TB instruction, robot TCP path varies with the value of $timebf_ver.

$timebf_ver = 1: Execution of the next line will start without waiting for the completion of the TB action;
$timebf_ver = 2: Execution of the next line will not start until the completion of the TB action;

For example,

```
1 L P[1] 2000 mm/sec FINE
2 L P[2] 2000 mm/sec CNT100 TB 0.5 sec. DO[1]=ON
3 L P[3] 2000 mm/sec FINE
```

with $timebf_ver = 2, the execution of line 3 does not start until DO[1] is turned ON. With $timebf_ver=1, line 3 is executed as if there were no TB motion option.

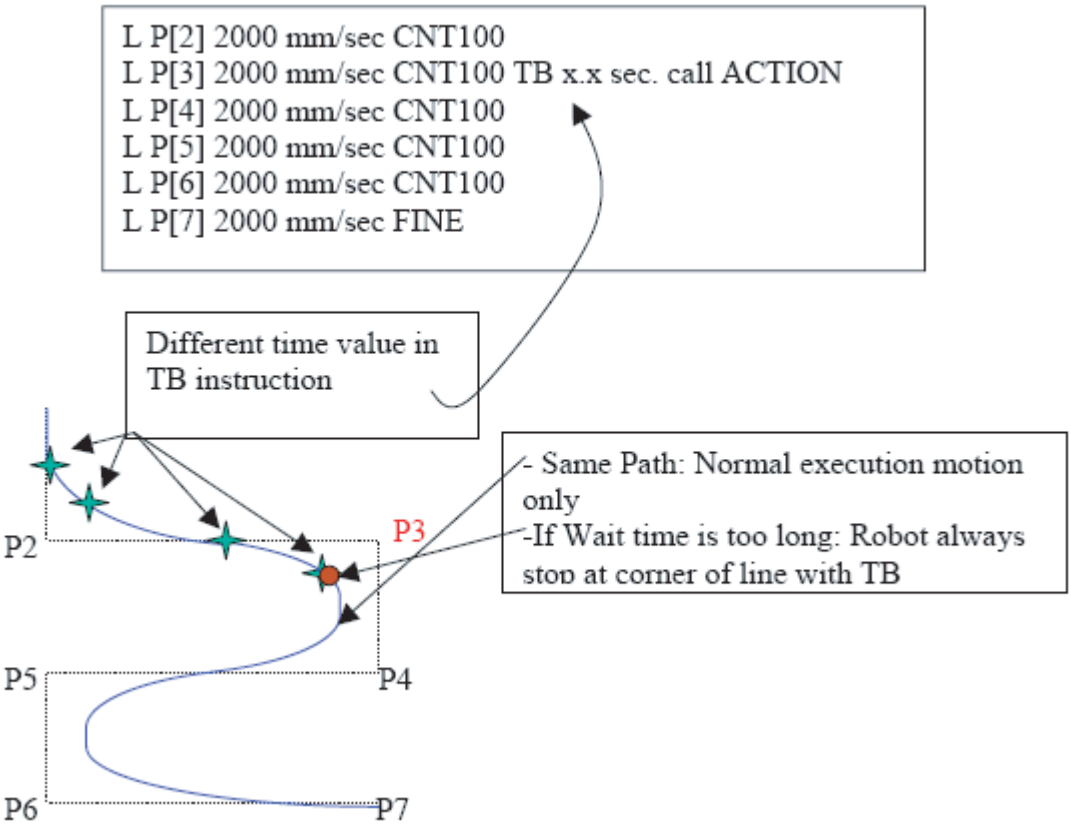
$timebf_ver = 3: The system will automatically determine the behavior based on TB action routine itself:

- If TB action routine is not too long and does not contain WAIT statement, then execution of the next line will start without waiting for the completion of the TB action (the same as when $timetbf_ver = 1);
- If TB action routine is lengthy or has WAIT statement, then execution of the next line will not start until the completion of the TB action (the same as when $timebf_ver=2);

Note With $timebf_ver = 1, 2, or 3, constant path feature is not guaranteed. That is, the TCP path of a robot may vary with the time value in the TB instruction and the time in a WAIT statement of the action (if the action has a WAIT statement);

$timebf_ver = 4: the system tries to maintain constant path. That is, robot TCP path can be maintained the same regardless of the time value in TB instruction, regardless of the time in WAIT statement in TB action and regardless of speed override, as shown in the following figure:

Figure 10–186. Normal Execution when $timebf_ver=4



Warning

During Hold, E-stop, and Resume, robot TCP path will not be maintained the same path, no matter what the value of $timebf_ver is!

Single Step Execution

If you are single stepping through a TIME BEFORE/AFTER instruction, the motion is paused when the sub program is called. The robot moves to the destination position as you single step through the sub program.

Power Fail Recovery

If Hot start is enabled and the controller is turned off while the sub program is executing, the sub program will resume execution from the same line the next time the controller is turned on. Because of this, the execution timing of the sub program is different from normal execution.

10.51.3 Execution Timing

Execution timing is the specified time when the CALL instruction is to be executed. Execution timing can be specified as:

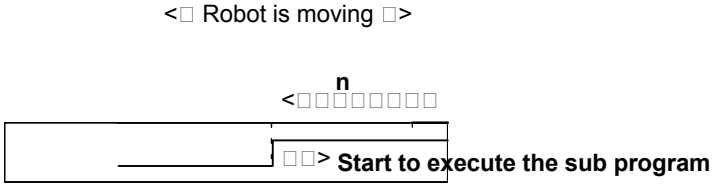
- TIME BEFORE : 0 to 30.0 sec
- TIME AFTER : 0 to 0.5 sec

The execution timing begins counting from the time robot motion is completed. Execution timing is not related to override.

If execution timing is set to 0 sec, the sub program is executed at almost the same time as the statement following the MOVE instruction. When 0 sec is set, the next line of the main program can be executed before the sub program starts to execute. The execution timing acts as follows:

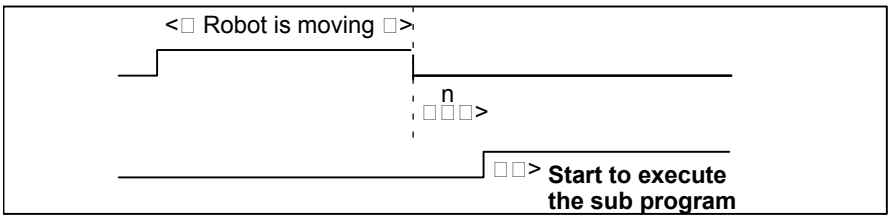
Specify [n sec] in the TIME BEFORE instruction. See Figure 10–187 .

Figure 10–187. Timing Sequence (TIME BEFORE instruction)



Specify [n sec] in the AFTER instruction. See Figure 10–188 .

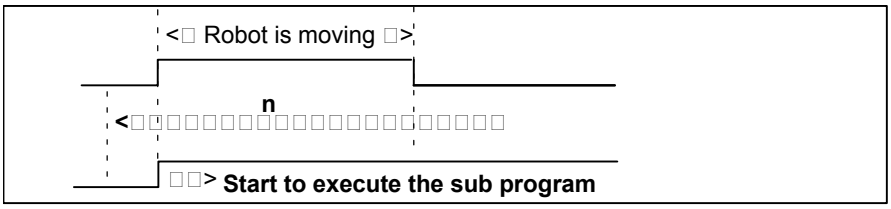
Figure 10–188. Timing Sequence (AFTER instruction)



The execution timing exceeds the period of the motion.

The sub program is executed at the same time motion is started. See Figure 10–189 .

Figure 10–189. Timing Sequence (TIME BEFORE instruction)



10.51.4 Recording a TIME BEFORE/AFTER Instruction

Use Procedure 10-75 to record a TIME BEFORE or TIME AFTER Instruction.

Procedure 10-75 Recording a TIME BEFORE or TIME AFTER Instruction

Steps

1. Move the cursor to the position where you want to add the motion option instruction.

```
PNS0001
  1: J P[1] 100% FINE
[END]
```

2. Press F4, [CHOICE]. You will see a screen similar to the following.

Note To search for the CALL item of a TIME BEFORE or TIME AFTER instruction press F5, [EDCMD], and then select FIND. **To replace the TIME BEFORE <-> TIME AFTER** , press F5, [EDCMD], and select REPLACE. Then select TIME BEFORE/AFTER. **To replace the**

CALL <program name> , press F5, [EDCMD], and select REPLACE. Then the <program name> can be replaced.

```
Motion Modify
 1                    5
 2                    6
 3 TIME BEFORE        7
 4 TIME AFTER         8
PNS0001
  1: J P[1] 100% FINE
[END]
```

3. Select TIME BEFORE. You will see a screen similar to the following.

```
PNS0001
  1: J P[1] 100% FINE
   :   TB sec...
[END]
 Enter Value
```

4. Type the execution time and press ENTER. For example, type 2.0 sec and press ENTER.

```
TIME statement
 1 CALL program                 5 GO[ ] = ...
 2 CALL program ()              6 AO[ ] = ...
 3 DO[ ] = ...                  7 WO[ ] = ...
 4 RO[ ] = ...                  8
PNS0001
  1: J P[1] 100% FINE
   :   TB 2.Osec
[END]
 Select item
```

5. Select CALL program. A list of available programs will be displayed.

```
PROGRAM list
 1 HANDOPEN          5
 2 HANDCLOS          6
 3                   7
 4                   8
PNS0001
  1: J P[1] 100% FINE
   :   TB 2.0sec
[END]
 Select item
```

6. Select the program you want to call with this instruction. In the following screen, HANDOPEN was selected. You will see a screen similar to the following.

```
PNS0001
  1: J P[1] 100% FINE
   :   TB 2.0sec HANDOPEN
[END]
```

10.51.5 TIME BEFORE Instruction Program Example

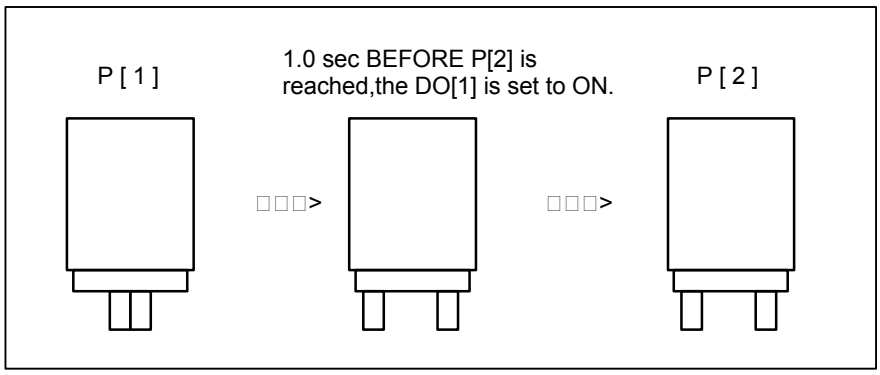
Figure 10–190 shows an example main and sub program which illustrate the use of the TIME BEFORE Instruction.

Figure 10–190. Main and Sub Program Examples

```
MAIN PROGRAM : PNS0001
     1: J P[1] 100% FINE
     2: J P[1] 100% CNT 100
      : TB 1.0 sec CALL
HANDOPEN
     3: CALL HANDCLOS
SUB PROGRAM : HANDOPEN
     1: DO[1] = ON
```

Figure 10–191 shows the execution of the main program, PNS0001.

Figure 10–191. Program Example for TIME BEFORE Instruction



10.51.6 Programming Hints

The following programming hints apply to the TIME BEFORE or TIME AFTER instruction.

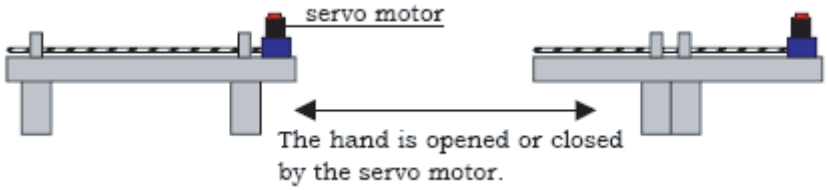
- The sub program called from the TIME BEFORE or TIME AFTER instruction cannot contain motion instructions. The motion group mask of the program must be [*,*,*,*,*,*,*,*].
- Until the called program has finished executing, the next line cannot be executed.
- There is no limit to the number of lines in a sub program.
- You can use the TIME BEFORE or TIME AFTER instruction in combination with any other motion option instructions, except application instructions such as the Arc Start[ ] instruction or the SKIP instruction.
- Only one TIME BEFORE or TIME AFTER instruction can be used with a single motion instruction.
- If you add CNT to a motion instruction, the timing when the motion instruction is completed is changed by the value of CNT.Even if 0 sec is specified in the TIME BEFORE instruction, the sub program might be executed too soon. You might need to use the TIME AFTER instruction to adjust the execution timing.
- When the TIME BEFORE or TIME AFTER instruction is used in the last line of the program, the sub program might not be called. This is because the execution of the main program is completed before the sub program is called. Therefore, do not teach the TIME BEFORE or TIME AFTER instruction on the last line.

10.52 TORQUE LIMIT

10.52.1 Introduction

The torque limit instruction allows you to limit the maximum torque value of the specified motor.

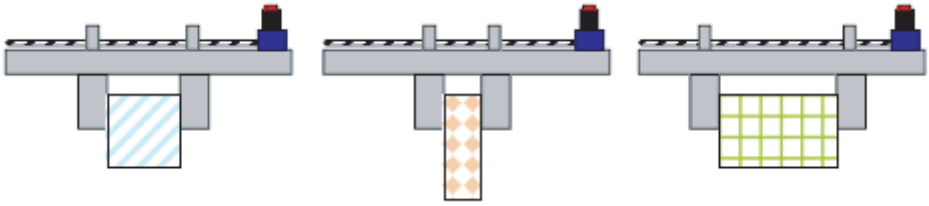
This function has two advantages, as shown by the example of the servo controlled gripper.



- **Advantage 1. Simple teaching**
  The taught point is only one (the point at which the fingers close completely) regardless of the size of part.
- **Advantage 2. Holding force is adjustable.**
  The output of motor can be limited to an arbitrary value and therefore holding force is adjustable and constant.

(1) Operation without torque function

Each part must have its own position for the part closed position. If sufficient force is applied to hold the part, errors could occur during operation. If the part were to crush slightly during handling, the force on the part would not be maintained and it might be dropped. Also, it is inconvenient in terms of maintenance (part size might change slightly from one part to the next).

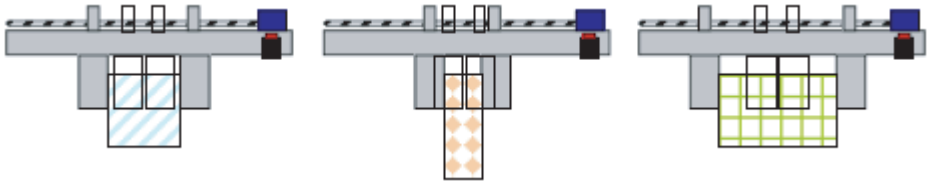


The taught point must be changed because the size of the workpiece varies.

(2) Operation with torque function

With the torque function feature enabled, only the open and closed positions must be programmed.

As the gripper starts to close, the fingers of the gripper will stop when they encounter the part to be picked up. The force of the grip is programmable and is maintained until the gripper is opened.



(3) Holding force is adjustable.

The maximum value of motor torque can be programmed in the torque limit instruction. Therefore the holding force can be adjusted from 1 to 100% of the maximum motor torque.

10.52.2 Specification

- This function is designed to limit the maximum value of torque generated for a servo-controlled gripper.
- The limit on torque is a percentage of a maximum value of motor torque.

  Syntax:

```
TORQ_LIMIT t% (t:0.0～100.0)
```

- The following system variables are used to determine which motor is under torque limitation. The default value is 7th axis of group 1.

  $TORQUE_LMT.$GROUP[i] … This value represents the group number. Please pay attention when setting this variable, since this could affect robot axes.

  i: group number

  $TORQUE_LMT.$GAi[j] … This value represents the group and the axis number for the motor to which torque limitation is applied.

  i: group number,
  j: axis number

  For example, if the motor is the first axis of group 2, $GA2[1] is set to TRUE and others are set to FALSE.

```
$TORQUE_LMT.$GROUP[2] = TRUE
$TORQUE_LMT.$GA2[1] = TRUE
```

```
$TORQUE_LMT.$GA2[2] = FALSE
 :       :
$TORQUE_LMT.$GA2[9] = FALSE
```

— **GA1[1] to GA1[6] are normally robot axes and are not set TRUE.**

— Multiple axes can be set TRUE.

- You must determine the optimal value of force for each part. The part weight and gripped surface may affect the torque required to hold the part firmly.
- During torque limit, the maximum of motor torque is not changed until the next torque limit instruction.

  Example.

```
1: J P[1] 100% FINE
2:   TORQ_LIMIT 20.0%        <- Torque limitation is set to 20%.
3: J P[2] 100% FINE
4: L P[2] 500mm/sec CNT100   <- Maximum Torque is limited to 20%.
 : :
6:   TORQ_LIMIT 100.0%       <- Torque limitation is set to 100%.
```

- If the torque limit instruction lowers the motor torque too much , the axis might not be able to reach the taught point. The value of the following system variable should be large enough that errors do not occur during normal operation.

  — **PARAM_GROUP[group].$STOPTOL[axis]**

  — **PARAM_GROUP[group].$STOPERLIM[axis]**

  — $ **PARAM_GROUP[group].$MOVER_OFFST[axis]**

- You can specify the maximum value of torque limit by using the following system variable.

  $TORQUE_LMT.$MAX_TRQ_LMT = Upper limit Lower limit is 0.1 fixed value.

10.52.3 Restriction

Warning

Do not enable this function for an axis that could fall due to the robot's own mass, because this function is also applied to the robot axis.

- This function only limits the *maximum* value of motor torque and, consequently, has no assurance of output torque of motor.
- This function is designed for the FANUC servo motor only. Experimentation is needed before using this feature if the motor is manufactured by another company.

Chapter 11

SERVO TOOL CHANGE FUNCTION

Contents

Chapter 11	SERVO TOOL CHANGE FUNCTION	11–1
11.1	OVERVIEW	11–3
11.1.1	1-1 Feature of function	11–3
11.1.2	1-2 Basic specification	11–3
11.1.3	Restrictions	11–3
11.1.4	1-4 System Configuration	11–4
11.1.5	1-5 Outline of installation	11–6
11.2	INITIAL SETUP	11–7
11.3	PRELIMINARY TOOL ATTACH OPERATION	11–7
11.4	TOOL CHANGE SETUP	11–9
11.5	SETTING THE REFERENCE POSITION	11–17
11.5.1	Battery-less type tools	11–17
11.5.2	Battery-mounted type tools	11–17
11.6	TOOL CHANGE INSTRUCTION	11–18
11.6.1	TOOL DETACH instruction	11–18
11.6.2	TOOL ATTACH instruction	11–18
11.6.3	Sample program	11–19
11.6.4	Foward execution	11–19
11.6.5	Backward execution	11–20
11.7	TOOL CHANGE SEQUENCE	11–21
11.8	TOOL CHANGE STATUS	11–21
11.9	TEACHING	11–22
11.9.1	Notice for teaching	11–22
11.9.2	Sample program	11–23
11.10	CONSTRUCTION OF SERVO TOOL CHANGE SCREEN	11–24
11.10.1	INITIAL SETUP SCREEN This screen can be displayed at CONTROL START Menu.	11–24
11.10.2	SETUP SCREEN	11–24

11.10.3 - STATUS SCREEN 11–25
11.11 TOOL CHANGE INITIAL SETUP 11–25
11.11.1 Setting Motion Parameters For Servo Tool Axes 11–25
11.12 TOOL CHANGE REFERENCE POSITION SETUP METHOD (BATTERY-LESS TYPE) 11–27
11.12.1 Reference Position Setup for Calibration Types 3 and 4 11–27
11.12.2 Reference Position Setup for Calibration Types 5 and 6 11–30
11.12.3 Quick Mastering Reference Position Setup 11–31
11.13 TROUBLESHOOTING 11–32
11.13.1 The Attach Instruction Is Executed When the Tool Is Not Attached. 11–32
11.13.2 The Robot Stopped During Calibration. 11–33
11.13.3 Calibration Motion Failed. 11–34
11.13.4 A Different Tool from That Specified by the Attach Instruction Is Attached. 11–34
11.13.5 The Attached Tool Has Been Detached by Mistake (Without Using the Detach Instruction). 11–35
11.13.6 TheTool Axis of a Detached Tool Has Moved. 11–35
11.13.7 The Battery Ran Low While the Tool Was Detached. 11–36

11.1 OVERVIEW

11.1.1 1-1 Feature of function

What is a servo tool?

A servo tool is a tool having an axis that is driven by a servo motor controlled by a robot controller.

- It is possible to switch more than two servo tools without power off.
- Switching the air drive tool and the servo tool is also available.

11.1.2 1-2 Basic specification

Table 11–1.

Item	Description
The total number of changeable tool	6 tools. Only one tool can be controlled at the same time.
The period of tool detach (standard)	0.4 sec (Battery type) / 0.4 sec (Battery less type) *1)
The period of tool attach (standard)	4 sec (Battery type)/6 sec (Batteryless type) *1)
Backup battery	Both battery type and batteryless type can be available.

(*1) The period includes the internal processing period, not include the period of robot motion for changing the tools.

11.1.3 Restrictions

Restrictions matters

- Switching is possible only between servo tools that are set within the same motion group.
- The tool change function cannot be used with a servo tool having more than one axis.
- Because servo tools that do not have a battery for tool axis pulse value backup require calibration after they are attached, these tools require longer processing time for tool change than battery-equipped tools.
- The motor of the tool change must be controlled by the same amp.

- The tool change instruction can not used by multi task. (The tool change instruction can not be execute by plural tasks at the same time)

Prohibition matters

- Do not force to detach the tool while the controller is off.
- Line tracking can not used with tool change function.

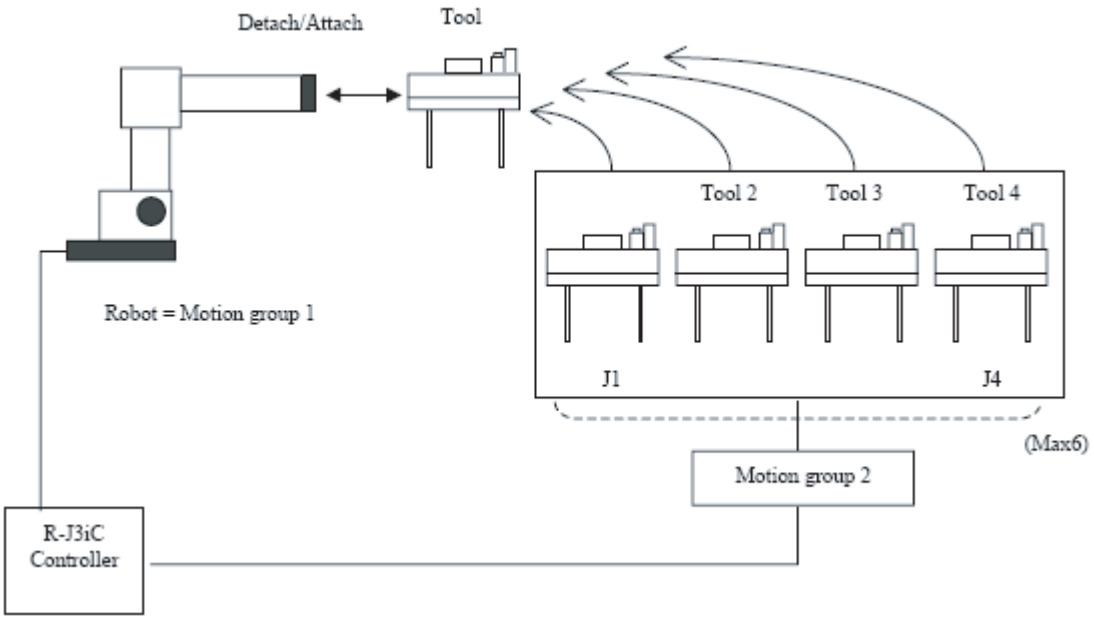
Notes on using the batteryless type

- When calibration motion type 1 or 2 is selected, calibration cannot be interrupted by a hold or emergency stop. (Program execution must not be stopped during tool attach processing.)
- When calibration motion type 3 or 4 is selected, the stroke limit must be expanded so that the tool axis touches the open limit.
- When calibration motion type 5 or 6 is selected, a detection switch such as a limit switch must be installed in the tool.

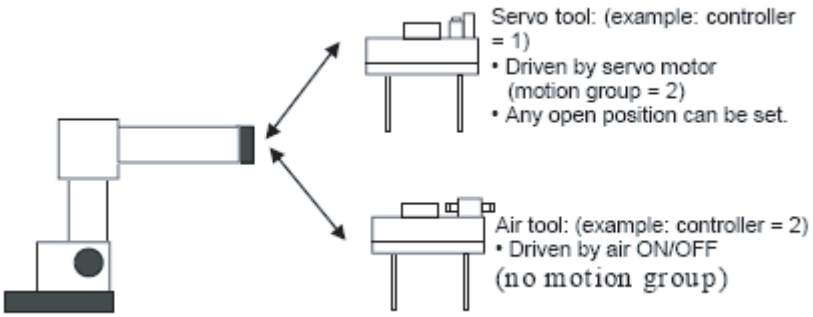
11.1.4 1-4 System Configuration

Servo tool change system is constructed as follows.

Figure 11–1. Example: Servo tool is the 2nd group and the number of tools is four.

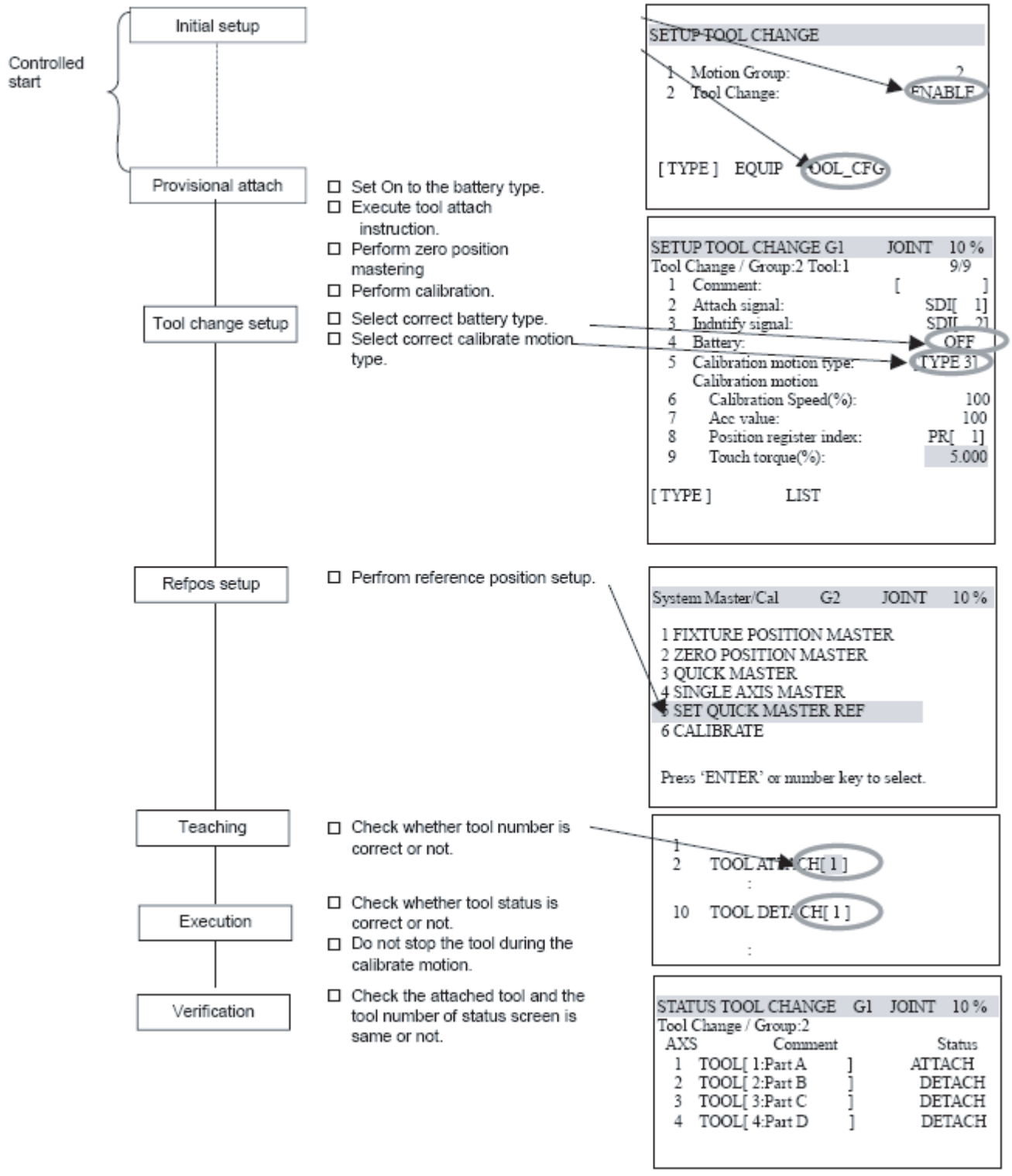


You can replace the servo tool with the air tool by switching on the controller.


Servo tool: (example: controller = 1)
• Driven by servo motor (motion group = 2)
• Any open position can be set.
Air tool: (example: controller = 2)
• Driven by air ON/OFF
(no motion group)

11.1.5 1-5 Outline of installation

Figure 11–2.



11.2 INITIAL SETUP

The following initial setup is needed to install the servo tool change system.

- Motion parameter setup (Independent Axes setup) of servo tool axes
- The configuration setup of servo tool number
- Enable the tool change function.

Note These setup must be done at the beginning of system installation.
These setup must be done only one time.

These setup is done at the controlled start. See Section 11.11.1 for detail information.

11.3 PRELIMINARY TOOL ATTACH OPERATION

When “Tool Change” is enabled as described in Section 11.2 , tools assigned to a target operation group are all detached.

Note Only when the tool change software is installed in a system in which servo tools have already been operated and a tool used previously is left attached, the tool is attached. In this case, preliminary tool attach steps xxx 1 to 9 explained below are unnecessary.

When using a tool for the first time, be sure to follow the steps below to perform the preliminary attach operation, then proceed to Section 11.4 .

1. 1. Press the MENUS key to display the screen menu.
2. Select 6,SETUP.
3. Press F1,TYPE to display the screen switch menu.
4. Select “Tool Change.” The tool change list screen appears.

```
SETUP TOOL CHANGE  G2    JOINT  10 %
Tool Change / Group:2
 AXS         Comment          BatteryType
  1  TOOL[ 1: Part A     ]   ON[------]
  2  TOOL[ 1: Part B     ]   ON[------]
  3  TOOL[ 1: Part C     ]   ON[------]
[ TYPE ]  GROUP  DETAIL
```

5. From the list, select one tool, and press F3,DETAIL. The tool change setup detail screen then appears.

6. On the tool change setup detail screen, temporarily set the battery to ON. (Even when the actual tool has no battery, select ON in this step. This item is to be set again after the preliminary attach operation is completed.)

```
SETUP TOOL CHANGE G1     JOINT  10 %
Tool Change / Group:2 Tool:1            1/4
  1  Comment:                 [Part A     ]
  2  Attach signal:                SDI[  1]
  3  Identify signal:              SDI[  2]
  4  Battery:                          ON

[ TYPE ]         LIST
```

7. Attach a target tool to the tool changer.
8. Create a program to teach an attach instruction.

```
1 TOOL ATTACH[ 1 ] ;
```

9. Execute the TOOL ATTACH instruction to attach the tool temporarily.
10. After the TOOL ATTACH instruction is executed, the following are issued:
    - SRVO-075 Pulse not established
    - SRVO-062 BZAL alarm
11. On the calibration screen, press F3,RES_PCA, and select YES.

```
System Master/Cal     G2     JOINT   10 %

 1 FIXTURE POSITION MASTER
 2 ZERO POSITION MASTER
 3 QUICK MASTER
 4 SINGLE AXIS MASTER
 5 SET QUICK MASTER REF
 6 CALIBRATE


 [TYPE]  LOAD  RES_PCA        DONE
```

12. Turn the power off then on.
13. Jog the servo tool axis. Continue jogging the servo tool axis until the following message disappears when the reset button is pressed: SRVO-075 Pulse not established.

```
System Master/Cal     G2     JOINT   10 %

 1 FIXTURE POSITION MASTER
 2 ZERO POSITION MASTER
 3 QUICK MASTER
 4 SINGLE AXIS MASTER
 5 SET QUICK MASTER REF
 6 CALIBRATE
```

14. In the same manner as for ordinary servo tools, perform zero-position mastering for the servo tool axis at the position where the stationary-side tip and the movable-side tip touch.
15. After completing tool change setup and reference position setting for the first tool, detach the first tool by using the TOOL DETACH instruction, then perform the preliminary attach operation for the second and subsequent tools.

11.4 TOOL CHANGE SETUP

The tool change setup must be done for each tool.

1. Press MENUS key.
2. Select “6 SETUP”.
3. Press F1 [TYPE].
4. Select “Tool change”

```
SETUP TOOL CHANGE  G2    JOINT  10 %
Tool Change / Group:2
 AXS         Comment          BatteryType
  1  TOOL[ 1: Part A     ]   ON[------]
  2  TOOL[ 1: Part B     ]   ON[------]
  3  TOOL[ 1: Part C     ]   ON[------]
[ TYPE ]  GROUP  DETAIL
```

5. Select tool from the list of tools, then press F3:DETAIL. The detail screen is displayed.

```
SETUP TOOL CHANGE G1     JOINT  10 %
Tool Change / Group:2 Tool:1            1/4
  1  Comment:                [Part A     ]
  2  Attach signal:               SDI[  1]
  3  Identify signal:             SDI[  2]
  4  Battery:                          ON

[ TYPE ]         LIST
```

Table 11–2. Common setup

Comment	Comment for each tool
Attach signal	This signal used for tool set verification. Available I/O type = SDI / RDI • This signal is check before tool attach/detach instruction. If this signal is OFF, the error occurs. • If the index is zero, the above checking is canceled.
Identify signal	This signal used for tool identification. Avaialble I/O type = SDI / RDI / GI • This signal is check before tool attach/detach instruction. If this signal is OFF, the error occurs. • If the index is zero, the above checking is canceled.
Battery	Select battery type. • ON = Battery backup type • OFF = Batteryless type
(Select the following when Battery is set to OFF in the common settings.)	
Calibration Motion Type	On battery-less tools, the position (pulse value) is lost when the tool is detached. So, calibration must be carried out when tool is attached. Calibration involves the following two operations: • Turning the motor two turns for positioning the pulse coder (this operation is simply referred to as "positioning" from here on) • Calibrating at a preset reference position The user selects from the following types, 1 to 6, to suit the user's particular system requirements. (See Table 11–3 and Figure 11–3 .)

Calibration Motion Type Selection Criteria
Types 1, 2

Select these types on systems capable of holding the tool detachment position as a result of braking the servo tool axis motor up to the next servo tool attachment.

Note With type 1 and 2, if calibration is interrupted by a hold or emergency stop, reference position data is lost. In such a case, see Section 11.13.2 .

When these types are set, the position where the tool is detached is automatically memorized as the reference position, and calibration is carried out at that position after the tool is attached. For this reason, the position when the tool is attached must be the same as position when it is detached.

The allowable errors of the detach and attach positions must fall within the respective following ranges for the direct-acting axis and rotary axis:

- Direct-acting axis: ±(gear ratio/2)mm or less around the detach position
- Rotary axis: ±((360/gear ratio)/2)deg or less around the detach position

Note that positional misalignment occurs during tool attachment if the above ranges are exceeded.

The difference between types 1 and 2 is in the operating direction during calibration. (See Table 11–3 and Figure 11–3 .)

Type 1: Specify this type when the tool is detached at a position where it is impossible to move the tool in the positive direction of manual feed by two or more turns of the motor.

Type 2: Specify this type when the tool is detached at a position where it is impossible to move the tool in the negative direction of manual feed by two or more turns of the motor.

Types 3, 4

Select these types on systems not capable of holding the tool detachment position as a result of the servo tool axis motor not having a brake up to the next servo tool attachment. On such a system, the reference position must be set to another position as the tool detachment position cannot be used as the reference position.

With these types, contact of the tool axis with the open limit is detected by using the motor torque, and the contact position is set as the reference position. (This is because, in the standard setting, the calibration motion causes the tool axis to move in the negative direction of manual feed.) For details on how to set the reference position, see Section 11.5 .

The difference between types 3 and 4 is in the operating direction during calibration. (See Table 11–3 and Figure 11–3 .)

Type 3: Specify this type when the tool is detached at a position where the distance to the open limit is two or more turns of the motor.

Type 4: Specify this type when the tool is detached at a position where the distance to the open limit is two or less turns of the motor.

Note In the calibration, to move the tool axis in the positive direction of manual feed (to use the limit on the positive side of manual feed), change the following system variable to 0: $SVTCTOOLxx.$CLOSDIR = 1 —> 0 (xx: Tool number)

Note With types 3 and 4, it is necessary to expand the stroke limit so that the tool axis touches the open limit.

Types 5, 6

Like types 3 and 4, specify these types on systems not capable of holding the tool detachment position as a result of the servo tool axis motor not having a brake up to the next servo tool attachment. With types 5 and 6, however, a detection signal must be provided on the tool by means of a limit switch, and the contact position must be set as the reference position. Install the detection switch in such a place that the detection switch is turned on when the tool axis is moved in the negative direction of manual feed (this is because, in the standard setting, the calibration motion causes the tool axis to move in the negative direction of manual feed). For details on how to set the reference position, xxx see Section 11.5 .

The difference between types 5 and 6 is in the operating direction during calibration. (See Table 11–3 and Figure 11–3 .)

Type 5: Specify this type when the tool is detached at a position where the distance to the position where the detect signal turns ON is two or more turns of the motor.

Type 6: Specify this type when the tool is detached at a position where the distance to the position where the detect signal turns ON is two or less turns of the motor.

Note In the calibration, to move the tool axis in the positive direction of manual feed, change the following system variable to 0: $SVTCTOOLxx.$CLOSDIR = 1 –> 0 (xx: Tool number)

Note With types 5 and 6, a switch such as a limit switch must be installed in the tool.

Table 11–3. Calibration Motion Type

Type	Calibration Motion ($SVTCTOOLxx.$CLOSDIR is 1.)	Set Position of Reference position
Type 1	The tool is moved in the negative direction of manual feed by two turns of the motor and is subsequently moved in the positive direction by two turns, then the tool is calibrated.NOTE 1 Set setting item (1) below.	Position when tool is detached Reference position is not required. NOTE 2

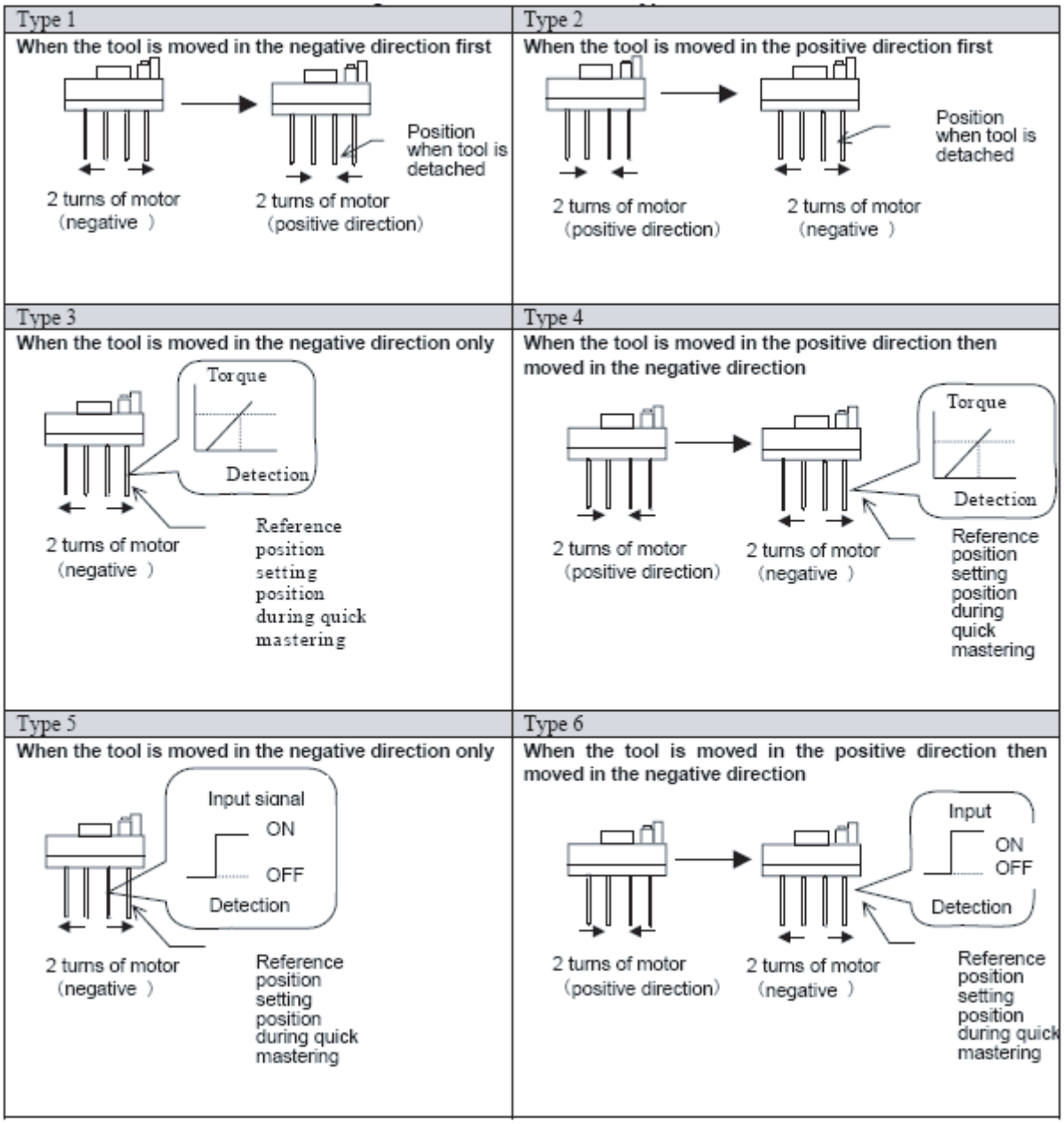
Table 11–3. Calibration Motion Type (Cont'd)

Type 2	The tool is moved in the positive direction of manual feed by two turns of the motor and is subsequently moved in the negative direction by two turns, then the tool is calibrated. Set setting item (1) below. NOTE 1	Position when tool is detached Reference position is not required. NOTE 2
Type 3	The tool is moved in the negative direction of manual feed until touch torque is detected, then the tool is calibrated. Arrival at the limit is detected by motor torque. Set setting items (1), (2) and (3) below.	Open limit position
Type 4	The tool is moved in the positive direction of manual feed by two turns of the motor and is subsequently moved to the limit on the negative side until touch torque is detected, then the tool is calibrated. Arrival at the limit is detected by motor torque. Set setting items (1), (2) and (3) below.	Open limit position
Type 5	The tool is moved in the negative direction of manual feed until the detection signal is turned on, then the tool is calibrated. Set setting items (1), (2) and (4) below.	Position where detection signal turns ON
Type 6	The tool is moved in the positive direction of manual feed by two turns of the motor and is subsequently moved in the negative direction until the detection signal is turned on, then the tool is calibrated. Set setting items (1), (2) and (4) below.	Position where detection signal turns ON

Note 1: Tool master data is lost if the operation is canceled due to a hold or an emergency stop during calibration that is carried out during the tool change sequence when types 1 or 2 are selected. If this happens, manually carry out zero position mastering.

2: With types 1 and 2, the system automatically memorizes the position where the servo tool is detached as the reference position. This memorized reference position is used in calibration when the same tool is next attached. For this reason, when types 1 or 2 are selected, the detachment position and the attachment position must be the same. Do not select types 1 or 2 when a motor with a brake is used for the servo tool axis.

Figure 11–3. Calibration Motion Type



The positive and negative directions in the above figure are determined on the assumption that $SVTCTOOLxx.$CLOSDIR = 1 is set. If this system variable is set to 0, the directions are reversed.

Screen for Calibration Types 1, 2

```
SETUP TOOL CHANGE G1     JOINT  10 %
Tool Change / Group:2 Tool:1            1/7
  1  Comment:                [          ]
  2  Attach signal:                 SDI[  1]
  3  Identify signal :              SDI[  2]
  4  Battery:                       OFF
  5  Calibration motion type:       [TYPE 2]
     Calibration motion
  6    Calibration Speed(%):            100
  7    Acc value:                       100
```

Screen for Calibration Types 3, 4

```
SETUP TOOL CHANGE G1     JOINT  10 %
Tool Change / Group:2 Tool:1            1/9
  1  Comment:                [          ]
  2  Attach signal:                  SDI[  1]
  3  Identify signal:                SDI[  2]
  4  Battery:                             OFF
  5  Calibration motion type:        [TYPE 4]
     Calibration motion
  6    Calibration Speed(%):             100
  7    Acc value:                        100
  8    Position register index:       PR[  1]
  9    Touch torque(%):                 5.000
```

Screen for Calibration Types 5, 6

```
SETUP TOOL CHANGE G1     JOINT  10 %
Tool Change / Group:2 Tool:1            1/9
  1  Comment:                [          ]
  2  Attach signal:              SDI[  1]
  3  Identify signal:            SDI[  2]
  4  Battery:                         OFF
  5  Calibration motion type:    [TYPE 6]
     Calibration motion
  6    Calibration Speed(%):          100
  7    Acc value:                     100
  8    Position register index     PR[  1]
  9    Detect signal:             SDI[  0]
```

1. Open limit operation setting (Set on all calibration types.)

Table 11–4. Calibration Speed and Acceleration Rate

Calibration speed	Specify the calibration motion operation speed
Accelerate rate	Specify the calibration motion accelerate rate Default is 100(%).

2. Position register No. setting

(Set as follows when types 3 to 6 are set in the Calibration Setup.)

Table 11–5. Position Register No.

Position register No	Position register for holding the momentary position where arrival at the reference position (open limit or signal input position) was detected Set the position register specified here exclusively for calibration. (Do not use the same position register in the program, for example.)

3. Open limit detection torque setting

(Set as follows when types 3 or 4 are set in the Calibration Setup.)

Table 11–6. Touch Torque

Touch torque (%)	Threshold value of the motor torque for detecting arrival at the open limit Set the value within range 0.0 to 100.0%.

4. Detection signal setting (Set as follows when types 5 or 6 are set in the Calibration Setup.)

Table 11–7. Detection Signal

Detection signal	Type and No. of input signal for detecting arrival at the reference position • SDI • RDI Specify a No. available for each of the signals as the signal No.

11.5 SETTING THE REFERENCE POSITION

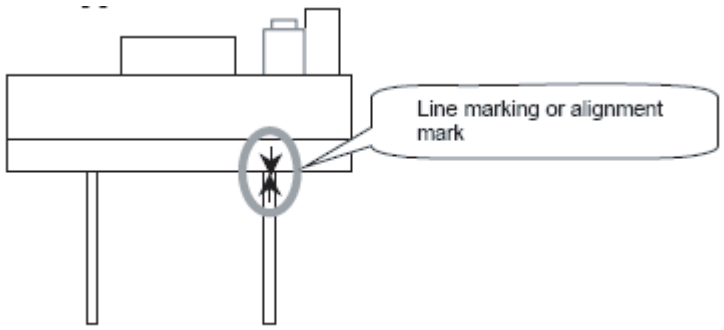
11.5.1 Battery-less type tools

On battery-less tools, the position (pulse value) is lost when the tool is detached. So, calibration must be carried out when tool is attached.
The position taken as the reference for calibration is called the "reference position."
On battery-less tools, the reference point must be set before the tool is detached. How the reference point is set differs according to the calibration motion type.

- Types 1, 2
  The system automatically sets the reference position when the servo tool is detached.
- Types 3, 4
  Create a program for detecting the reference position using the touch torque, and execute this program to calculate the reference position. For details, refer to Section 11.12.1 .
- Types 5, 6 Create a program for detecting the reference position using the detection signal, and execute this program to calculate the reference position. For details, refer to Section 11.12.2 .

11.5.2 Battery-mounted type tools

On battery-mounted type tools, the position (pulse value) is not lost even if the tool is detached. Therefore, the reference position used for calibration when the tool is attached need not be set as it needs to be on a battery-less type tool. However, that we recommend setting a reference position for quick mastering by marking a line or alignment mark as shown in the figure below for extra safety in the event that the battery comes loose. For details on how to set the reference position for quick mastering, refer to Section 11.12.3 .



11.6 TOOL CHANGE INSTRUCTION

For changing servo tools, the following instructions are provided.

11.6.1 TOOL DETACH instruction

This instruction can detach the servo tool without power off.

Format : -
TOOL DETACH[i]

i : Tool number (1 –> 6)
Specify the tool number which you want to detach.
R[ ] and AR[ ] can be used.

Ex)

```
TOOL DETACH[1]
TOOL DETACH[R[2]]
TOOL DETACH[AR[1]]
```

11.6.2 TOOL ATTACH instruction

This instruction can attach the servo tool without power off. In case of batteryless tool, the mastering and calibration are performed automatically.

Format : -
TOOL ATTACH[i]

i : Tool number (1 –> 6)
Specify the tool number which you want to detach.
R[ ] and AR[ ] can be used.

Ex)

```
TOOL ATTACH[5]
TOOL ATTACH[R[3]]
TOOL ATTACH[AR[1]]
```

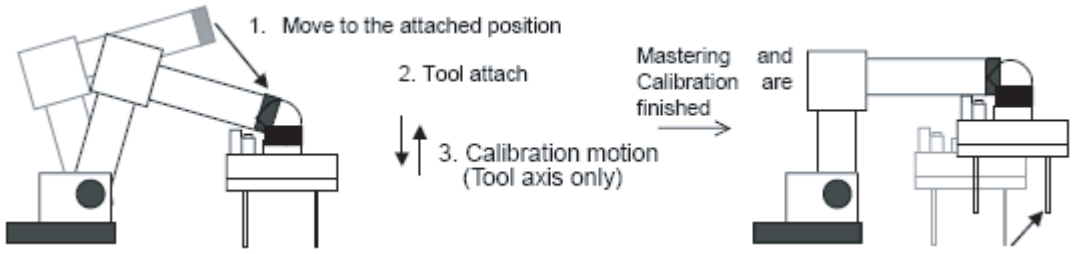
11.6.3 Sample program

```
                                          [Explanation]
 1: J P[1:Approach] 100% FINE             1: Move to approach position
 2: L P[2:TOOL1 SET] 100mm/sec FINE       2: Set tool axis to 0 positio n
 3: TOOL DETACH[1]                        3: Detach tool 1
 4: L P[1:Approach] 100mm/sec FINE        4: Release tool1
              :
11:J P[11: Approach] 100% FINE
12: L P[12:TOOL2 SET] 100mm/sec FINE
13: TOOL ATTACH[2]                        13: Attach tool 2
14: UTOOL_NUM=2 2                         14: Change tool frame number
15: L P[13: Approach] 100mm/sec FINE
:
99: END                                   99: Normal end
```

Note The above is merely a sample program. Create programs to suit the requirements of your system

11.6.4 Foward execution

It is necessary to check that the tool number of instruction is same as the actual tool. And in the case of attach instruction, the calibration motion type must be set correct type. In the case of batteryless tool, the calibration motion (3 in the following figure) is done.



Note In the case of calibration motion type is Type 1 or Type 2, if the motion is stopped by e-stop or HOLD during the calibration motion, the mastering data is lost. If the mastering data is lost, please do the zero mastering and calibration by manual again. Otherwise,

1. The current position on TP is not matched to the actual position.
2. If the motion instruction is executed, the following alarm is occurred. “MOTN-049 Attempt to move w/o calibrated”

3. The TOOL DETACH instruction does not work until manual mastering and calibration are finished.

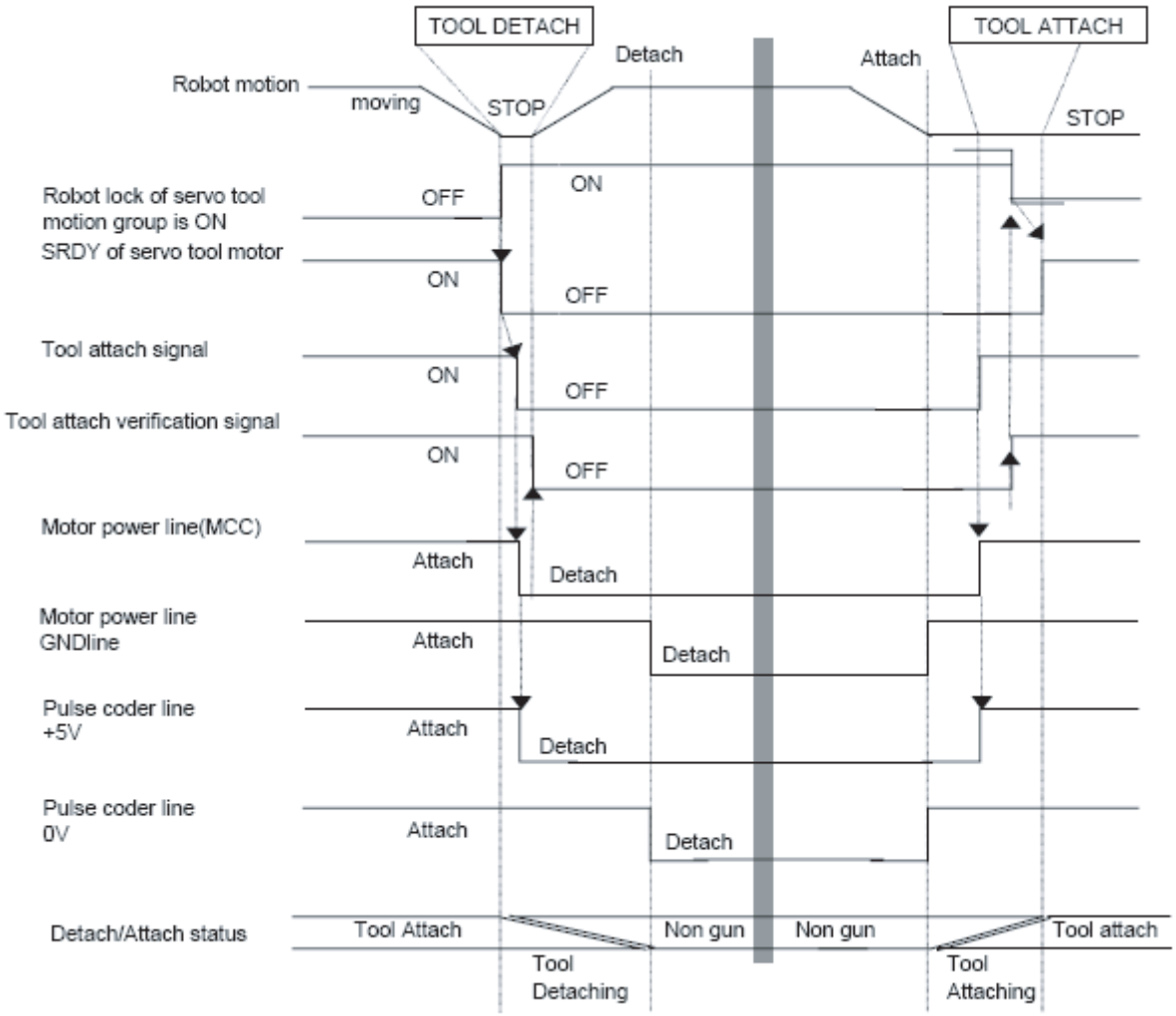
In the case of TOOL DETACH instruction, both robot and servo tool do not move at all.

11.6.5 Backward execution

Both TOOL ATTACH and TOOL DETACH instruction can not be execute by backward execution.

11.7 TOOL CHANGE SEQUENCE

Figure 11–4. Tool Change Sequence



11.8 TOOL CHANGE STATUS

On the tool change status screen, verify that which tool is attached or detached now.

1. Press MENUS key.
2. Select “0 next page”, ”4 STATUS”.
3. Press F1 [TYPE].
4. Select “Tool change”.

```
STATUS TOOL CHANGE  G1  JOINT  10 %
Tool Change / Group:2
 AXS         Comment            Status
  1  TOOL[ 1:Part A     ]       ATTACH
  2  TOOL[ 2:Part B     ]       DETACH
  3  TOOL[ 3:Part C     ]       DETACH
  4  TOOL[ 4:Part D     ]       DETACH
```

The following status are displayed.

Table 11–8. Status Tool Change Screen Items

Item	Description
AXS	The axis number of the motion group which is assigned to the current motion group.
Tool	The tool number which is assigned to the above axis.
Status	Tool status ATTACH : The tool is attached. DETACH : The tool is detached.
Comment	The comment of each tool.

11.9 TEACHING

In the tool change system, please note the folloiwng points.

11.9.1 Notice for teaching

Position data

In the tool change system, the each tool is assigned to the each axis of the servo tool motion group. For example, if three kinds of tool are assiged to tool change system, the servo tool motion group has three axes data. This means three axes data are recorded when servo tool position is taught.

Ex. The position data of tool1 which is assigned to first axis

```
GP1:   UF:0, UT:1,       Config : 'N, 0, 0, 0',
  X =  1879.25 mm,     Y = -1035.37 mm,     Z =   162.65 mm,
  W =  -178.67 deg,    P =  -13.68 deg,     R =  -37.32 deg
GP2:   UF : 0, UT : 1,
 J1 =  10.00 mm    J2 = 0.00 mm   J3 = 0.00 mm
```

Please note that this position data has not only tool 1 data but also tool 2 and tool 3 data. This means this position data can move tool 2 and tool 3. If tool 2 is attached, tool 2 moves to 0mm position. This specification is for the flexibility of the program. But, if the user attaches different tool, it may cause interfer of tool. To prevent this problem, please specify different tool number to the tool close direction for each tool.

By this function, the position data for each tool has the different tool number. Becuse of this, even if the user execute the program with different tool, the program is stopped by "INTP-253 Tool frame number mismatch".

11.9.2 Sample program

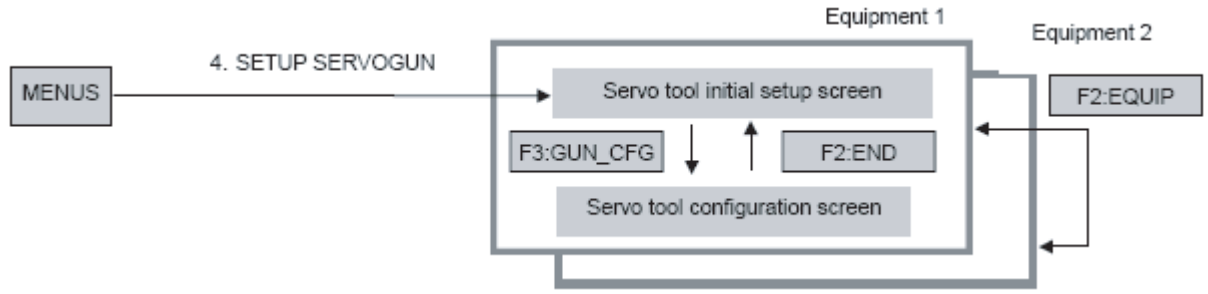
```
Sample program
          :                                    [Explanation]
 2: TOOL ATTACH[1] Move with tool number 2  2:Attach tool1
 3: UTOOL_NUM=1    Move with tool number 2  3 Change tool frame number to 1
          :                                        : :
11:J P[11] 100% FINE                        11:Handle with tool1
12:L P[12] 100mm/sec FINE                   12:Handle with tool1
13:J P[13:Approach] 100% FINE               13:Move to approach position
14:L P[14:Tool1 set] 100mm/sec FINE         14:Set tool1 to tool stocker
15:  TOOL DETACH[1]                         15:Detach tool1
16:L P[13:Approach] 100mm/sec FINE          16:Release tool1
             : :                                 :
21:J P[21:Approach] 100% FINE
22:L P[22:Tool2] 100mm/sec FINE
23:  TOOL ATTACH[2]                         23:Attach tool2
24:  UTOOL_NUM=2                            24:Change tool frame number to 2
25:L P[23:Approach] 100mm/sec FINE
26:J P[24] 100% FINE                        26:Handle with tool2
27:L P[25] 100mm/sec FINE                   27:Handle with tool2
             : :                                 :
 99:  END                                    99:Normal end
```

Note This is sample program. Create the program related to your system.

11.10 CONSTRUCTION OF SERVO TOOL CHANGE SCREEN

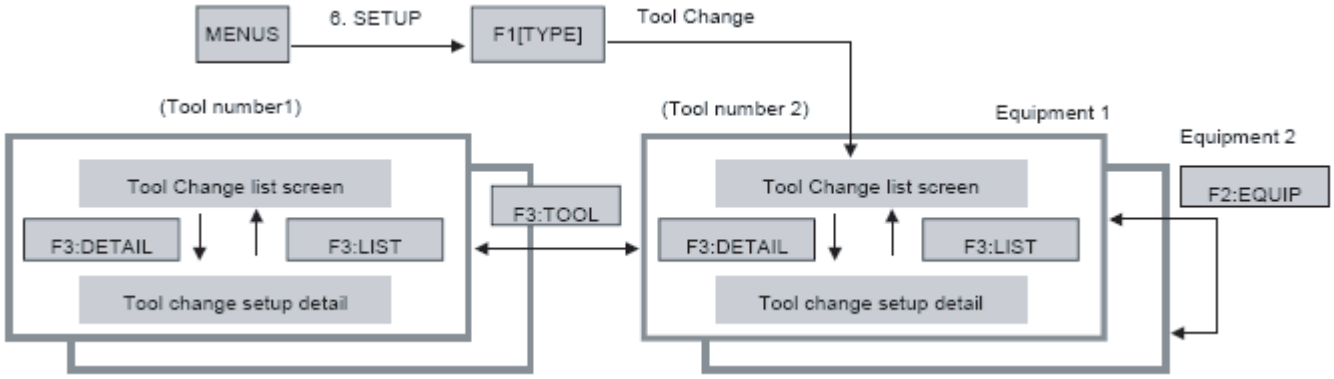
11.10.1 INITIAL SETUP SCREEN This screen can be displayed at CONTROL START Menu.

Figure 11–5.



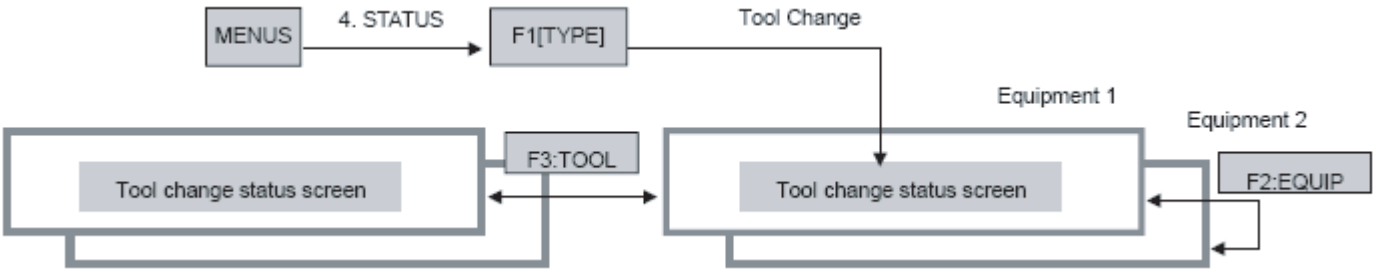
11.10.2 SETUP SCREEN

Figure 11–6.



11.10.3 - STATUS SCREEN

Figure 11–7.



11.11 TOOL CHANGE INITIAL SETUP

The following explains how to make initial settings required for tool change system installation.

Note The setup procedure explained below is required just once when the system is installed. Once settings are made, they need not be changed in ordinary use.

11.11.1 Setting Motion Parameters For Servo Tool Axes

Set tool axes used for the tool change function within each operation group according to the procedure below.

- 1st to 6th axes: Independent Axes setting procedure
- Assigning Tool Numbers to Servo Tool Axes

Open the servo tool initial setup screen from the control start menu, and set the following:

1. Turn off the power. While holding the PREV key and the F–> key, turn on the power.
2. After a while, a screen as shown below appears. Select 3, Controlled start.

```
--------CONFIGURATION MENU--------
1 Hot start
2 Cold start
3 Controlled start
4 Maintenance
Select >
```

3. After a while, the control start screen as shown below appears.

```
FANUC Handling tool                1/1
1 F Number                   F000000
```

4. Press the MENUS key to display the screen menu.
5. Select 0, NEXT then 4, TOOL CHANGE. The servo tool initial setup screen appears.

```
SETUP TOOL CHANGE
  1  Motion Group:              2
  2  Tool Change:             ENABLE
 [ TYPE ]  GROUP  TOOL_CFG
```

Table 11–9. Servo Tool Change Initial Setup Screen Items

Item	Description
Motion Group:	Input the operation group number with which servo tools are set.
Tool Change:	Enable or disable the tool change function.

The tool change function switches servo tools that are set within the same operation group. Note that it is impossible to switch between tools that are set in different operation groups.

Set the items listed below for the tool axes included in the selected operation group.
If there is no problem with using the standard settings, the standard settings need not be modified.

```
SETUP TOOL CHANGE  G2  JOINT  10 %
Tool Change / Group:2
    AXS        Comment
  1  J1  TOOL [ 1: Part A     ]
  2  J2  TOOL [ 2: Part B     ]
  3  J3  TOOL [ 3: Part C     ]
[ TYPE ]   END
```

Table 11–10. Servo Tool Change Setup Screen Items

Item	Description
TOOL: :	Set a number from 1 to 6.
Comment:	Set a comment on the tool number. You can also set a comment on the tool change setup screen after cold start.

11.12 TOOL CHANGE REFERENCE POSITION SETUP METHOD (BATTERY-LESS TYPE)

11.12.1 Reference Position Setup for Calibration Types 3 and 4

- **Reference Position Setup Program**

  Create the following program to calculate the reference position based upon the sign of the calculated touch torque.

  Note Only the servo tool axis operation group can be used as the operation group of the program to be used for setting the reference position.

  Example) When the servo tool axis operation group is the 2nd group:
  The operation group becomes [*,1,*,*,*].

  Sample Program Operation Group=[*,1,*,*,*]

Figure 11–8.

```
1:J P[1:ZERO POS] 10%
positioning
(when sign of touch torque is
plus)
2:SKIP CONDITION
  $MISC[g].$HPD_TRQ[ga] > 2 *1)
  (when sign of touch torque is
  minus)
2:SKIP CONDITION
  $MISC[g].$HPD_TRQ[ga] <-2 *1)
3:J P[2:TOOL OPEN END] 1% *2)
  FINE ACC 100 *3)SKIP,
  LBL[1],PR[1]=JPOS
4:END
```

```
[Explanation]
1:Servo tool axis 0 position
(The position need not necessarily
be the 0 position.)
2:As the touch torque in the skip
condition, set the threshold value
of the torque from which arrival
at the limit can be detected when
the tool moved in the negative direction
of manual feed touches the limit. Pay
attention to the sign of the touch torque.
3:Move the tool in the negative
direction of manual feed
to touch the limit.  If detection
```

```
5:LBL[1]                          fails, jump to LBL[1].
6:UALM[1]                         4:Normal end
                                  5:Jump to here if torque cannot be
                                  detected.
                                  6:Generate user alarm.
```

Note The above is merely a sample program. Create programs to suit the requirements of your system.
g: Operation group number of currently attached tool
ga: Axis number of currently attached tool
If the currently attached servo tool axis is taken as the 1st axis of the 2nd group, then g and ga are as follows: g=2, ga=1

Program Operation Adjustment Method

Adjust the value at the shaded section in the sample program to adjust program operation. Operation adjusted here is used as calibration motion that is carried out when attaching the tool.

1. Touch torque
   Increase the value from about ±2, and calculate the value at which erroneous detection of contact occurs. The sign of touch torque is the sign that is currently displayed at Touch torque at the Calibration Setup screen.

```
SETUP TOOL CHANGE G1     JOINT  10 %
Tool Change / Group:2 Tool:1            1/9
  1  Comment:                    [          ]
  2  Attach signal:                  SDI[  1]
  3  Identify signal:                SDI[  2]
  4  Battery:                        OFF
  5  Calibration motion type:        [TYPE 4]
     Calibration motion
  6    Calibration Speed(%):             100
  7    Acc value:                        100
  8    Position register index:      PR[  1]
  9    Touch torque(%):         ( + )  5.000

[ TYPE ]          LIST
```

Note After confirming the sign in this screen, teach it to the program. Do not make a mistake when entering the sign. If you do so, contact with the open limit may not be detected.

2. Program operation speed
   To set the same calibration motion that is carried out when attaching the tool, execute this program at the same override (normally, 100%) as when the attach instruction is executed for the tool used during actual production. For this reason, reduce the program operation speed so that contact is made with the open limit at a safe speed even if the

program is executed at a high override. For added safety, increase the program operation speed value successively from 1%.

3. Accelerate rate
Decrease the accelerate rate if motion is sluggish even if the program operation speed is 1%.

- **Reference Position Setup**

After you have adjusted operation in the program, execute the program to bring the servo tool axis into contact with the open limit.

Set up the reference position according to the procedure described in Section 11.12.3 with the servo tool contacting the open limit.

Tool Change Setup

To set operation in the newly adjusted program to the same motion as calibration motion that is carried out when attaching the tool, reflect the program operation speed, accelerate rate and touch torque adjustment values in the tool change setup.

1. Press the SELECT SCREEN key to display the Screen menu.
2. Select 6 SETUP.
3. Press F1 TYPE to display the Screen Selection menu.
4. Select TOOL CHANGE. The Tool Change Setup List screen appears.
5. Select the tool to be adjusted from this list, and press F3 DETAIL to display the Tool Change Details Setup screen.
6. Set the operation speed, accelerate rate and touch torque adjustment values at the appropriate sections in the following screen:

```
SETUP TOOL CHANGE G1     JOINT  10 %
Tool Change / Group:2 Tool:1            1/9
  1  Comment:                [          ]
  2  Attach signal:              SDI[  1]
  3  Identify signal:            SDI[  2]
  4  Battery:                         OFF
  5  Calibration motion type:    [TYPE 3]
     Calibration motion
  6    Calibration Speed(%):          100   <-
  7    Acc value:                     100   <-
  8    Position register inde     PR[  1]
  9    Touch torque(%):      ( + )  5.000   <-
```

11.12.2 Reference Position Setup for Calibration Types 5 and 6

- **Provision of Detection Signal**

  With calibration types 5 and 6, the position where the detection signal is input is taken as the reference position during quick mastering. The user must install a limit switch for providing the detection signal capable of assuring the absolute position of the tool.

- **Reference Position Setup Program**

  Create the following program to calculate the reference position based upon the sign of the calculated touch torque.

  Note Only the servo tool axis operation group can be used as the operation group of the program to be used for setting the reference position.
  Example) When the servo tool axis operation group is the 2nd group: The operation group becomes [*,1,*,*,*].

  Sample Program Operation Group=[*,1,*,*,*]

```
                                [Explanation]
1:J P[1:ZERO POS] 10%           1:Servo tool axis 0 position (The position
  FINE                            need not necessarily be the 0 position.)
2:SKIP CONDITION DI[ 1*1)]      2:Set turning ON of the provided detection
  =ON                             signal as the scan condition.
3:J P[2:SWITCH POS] 1%          3:Move tool to position where detection
  *2) FINE ACC100 *3)SKIP,        signal turns ON. (Jump to label[1] if the
  LBL[1],PR[1]=JPOS               detection signal does not turn ON.)
4:END                           4:Normal end
5:LBL[1]                        5:Jump to here if detection signal does not
6:UALM[1]                         turn ON.
                                6: Generate user alarm
```

  Program Operation Adjustment Method

  Adjust the value at the shaded section in the sample program to adjust program operation. Operation adjusted here is used as calibration motion that is carried out when attaching the tool.

  1. Detection Signal
     Assign the provided detection signal to SDI[ ] or RDI[ ].
  2. Program Operation Speed
     To set the same calibration motion that is carried out when attaching the tool, execute this program at the same override (normally, 100%) as when the attach instruction is executed for the tool used during actual production. For this reason, reduce the program operation speed so that the switch will not be broken even if the program is executed at a high override. For added safety, increase the program operation speed value successively from 1%.

3. Accelerate rate
   Decrease the accelerate rate if motion is sluggish even if the program operation speed is 1%.

- **Reference Position Setup**

After you have adjusted operation in the program, execute the program to stop the servo tool axis at the position where the detection signal turns ON.

- **Tool Change Setup**

To set operation in the newly adjusted program to the same motion as calibration motion that is carried out when attaching the tool, reflect the program operation speed, accelerate rate and detection signal adjustment values in the tool change setup.

1. Press the SELECT SCREEN key to display the Screen menu.
2. Select 6 SETUP.
3. Press F1 TYPE to display the Screen Selection menu.
4. Select TOOL CHANGE. The Tool Change Setup List screen appears.
5. Select the tool to be adjusted from this list, and press F3 DETAIL to display the Tool Change Details Setup screen.
6. Set the operation speed, accelerate rate and detection signal adjustment values in the following screen:

```
SETUP TOOL CHANGE G1     JOINT  10 %
Tool Change / Group:2 Tool:1            1/9
  1  Comment:                   [          ]
  2  Attach signal:                 SDI[  1]
  3  Identify signal:               SDI[  2]
  4  Battery:                       OFF
  5  Calibration motion type:       [TYPE 5]
     Calibration motion
  6    Calibration Speed(%):            100   <-
  7    Acc value:                       100   <-
  8    Position register index:     PR[  1]
  9    Detect signal:              SDI[  1]   <-
```

11.12.3 Quick Mastering Reference Position Setup

Calibration (quick mastering data) is automatically carried out when the servo tool is changed. However, the reference points must be set to each of the tools to enable automatic calibration. The reference points are set with the robot in a cold start state. Follow the procedure below to set the reference points:

1. Press the AUX key to select 3 CHANGE GROUP. Change the operation group to the desired servo tool group (e.g. 2nd group).
2. Press the SELECT SCREEN key to display the Screen menu.
3. Press 0 NEXT and 6 SYSTEM.
4. Press F1 TYPE to display the Screen Selection menu.
5. If CALIBRATE is not displayed in the menu, select Variables to display the System Variables screen. Set the $MASTER_ENB system variable to "1".
6. Select CALIBRATE at 4. to display the Calibration screen.

```
SYSTEM Master Cal     G2    JOINT 10 %
                      TORQUE = [ON ]
  1 FIXTURE POITION MASTER
  2 ZERO POSITION MASTER
  3 QUICK MASTER
  4 SINGLE AXIS MASTER
  5 SET QUICK MASTER REF
  6 CALIBRATE

   Press 'ENTER' or number key to select.
```

7. Select 5 SET QUICK MASTER REFERENCE and F4 YES.
8. The message "Quick master reference point is set." is displayed. This indicates that setting of the reference point is completed.

11.13 TROUBLESHOOTING

11.13.1 The Attach Instruction Is Executed When the Tool Is Not Attached.

- **Phenomenon**

  The tool is not actually attached. So, a communications alarm such as "SRVO-068 DTERR alarm" occurs on the pulse coder. Correct this by Remedy 1 or Remedy 2 below according to the present conditions.

- **Remedy 1**

  1. Turn the machine lock ON in the servo tool operation group (normally, group 2) in the Test Cycle screen.
  2. Move the robot by manual feed so that the robot and tool are physically attached.
  3. Turn the controller OFF then ON again.

4. Battery-less tool (calibration motion types 1, 2)

   The tool pulse values and mastering data are lost. So, first move the tool by manual feed to cancel the "SRVO-075 Pulse not established" alarm.

5. Then, carry out tool 0 position mastering and calibration manually.

6. Battery-less tool (calibration motion types 3, 4, 5, 6)

   The tool pulse values and mastering data are lost. So, move the tool by manual feed to cancel the "SRVO-075 Pulse not established" alarm.

7. Carry out tool 0 position mastering and calibration manually. The tool 0 position may be an approximate position.

8. Execute the detach instruction to temporarily detach the tool. Also, physically detach the tool from the robot. Otherwise, the tool will not be calibrated at the next attach operation.

9. Execute the attach instruction again to attach the tool. Calibration this time is carried out at the reference position to restore the position.

10. Battery mounted tool

    The tool is attached and positioned, so it can be used as it is.

- **Remedy 2**

  1. Directly change the system variable to detach the tool. To do this, set the following system variable to "0": $SCR_GRP[servo tool operation group No. (normally, 2)].$AXISORDER[all 1 to 9] = 0
  2. Turn the controller power OFF then back ON.
  3. The robot starts up with the tool in a detached state. Move the robot by manual feed to physically attach the robot and tool.
  4. Execute the attach instruction to attach the tool.
  5. The tool can be used as it is after it is attached regardless of the battery type.

11.13.2 The Robot Stopped During Calibration.

- **Phenomenon**

  This phenomenon occurs only on battery-less tools. Note that the phenomenon and remedy vary according to the preset calibration type.

  Battery-less tool (calibration motion types 1, 2) The "SVGN-012 Tool mastering data is lost" alarm has occurred.

  Battery-less tool (calibration motion types 3, 4, 5, 6) Alarm has not occurred.

- **Remedy**

Calibration motion types 1, 2

1. The tool pulse values and mastering data are lost. So, first move the tool by manual feed to cancel the "SRVO-075 Pulse not established" alarm.

   This alarm may not be occurring. If so, skip this step and proceed to step 2.

2. Then, carry out tool 0 position mastering and calibration manually.

Calibration motion types 3, 4, 5, 6

1. Tool operation can be resumed as it is.

11.13.3 Calibration Motion Failed.

- **Phenomenon**

  The "SVGN-020 Calibrate motion is failed" alarm occurred. This phenomenon occurs only when the calibration motion type is 3, 4, 5 or 6 on a battery-less tool. This alarm occurs when the touch torque or detection signal conditions are not satisfied during calibration.
  In both of the following Remedies 1 and 2, the tool is calibrated again. So, calibration motion will fail again if the touch torque or detection signal conditions are not satisfied. Before you remedy this phenomenon, make sure that the touch torque or detection signal conditions are correct.

- **Remedy 1**

  Carry out steps 6 to 9 of G-1 Remedy 1.

- **Remedy 2**

  Same as Remedy 2 of G-1.

11.13.4 A Different Tool from That Specified by the Attach Instruction Is Attached.

- **Phenomenon**

  The tool is attached to the robot, but the tool's current position is the wrong value. On a battery-less tool, an alarm may cause the robot to stop during calibration.

- **Remedy**

  The following describes an example where tool 2 (actually attached tool) is attached to tool 1 (tool No. taught by the attach instruction).

1. In this state, the tool cannot be detached from the robot by the detach instruction. So, directly change the system variable to detach the tool. To do this, set the following system variable to "0":

   $SCR_GRP[servo tool operation group No. (normally, 2)].$AXISORDER[all 1 to 9] = 0

2. Turn the controller power OFF then back ON.
3. The tool starts up in a detached state.
4. Execute the attach tool[2] instruction to attach tool 2. (In this state, tool 2 should be physically connected to the robot.)

   When tool 2 is a battery-less tool and the calibration motion type is 1 or 2:
   If tool 2 is moved when attached by mistake instead of tool 1, tool 2 will move and be at a position different to when it was detached. So, tool 2 may be calibrated to the wrong position by the attach instruction executed in step 4 above, resulting is misalignment between its actual and desired positions. In this case, carry out tool 0 position mastering and calibration manually.

   When tool 2 is other than the above:
   Tool 2 can be used as it is.
5. Tool 1 can be used as it is by executing the attach instruction to attach the tool.

11.13.5 The Attached Tool Has Been Detached by Mistake (Without Using the Detach Instruction).

- **Phenomenon**

  The tool is not actually attached. So, a communications alarm such as "SRVO-068 DTERR alarm" occurs on the pulse coder. Correct this by Remedy 1 or Remedy 2 below according to the present conditions.

- **Remedy 1**

  Same as Remedy 1 of Section 11.13.1 .

- **Remedy 2**

  Same as Remedy 2 of Section 11.13.1 .

11.13.6 TheTool Axis of a Detached Tool Has Moved.

- **Phenomenon**

Battery-less tool (calibration motion types 1, 2): Misalignment has occurred. Misalignment occurs as a result of calibration at the wrong position when the tool is attached with the tool axis having moved to a different position. Note, however, that misalignment does not occur when tool axis movement is within the allowable errors indicated below. The allowable error must fall within the respective following ranges for the direct-acting axis and rotary axis:

Direct-acting axis: ±(gear ratio/2)mm or less around the detach position
Rotary axis: ±((360/gear ratio)/2)deg or less around the detach position

Other instances:
No problem has occurred.

- **Remedy**

Battery-less tool (calibration motion types 1, 2):

1. Change the battery with the tool attached.
2. The tool can be used as it is.

11.13.7 The Battery Ran Low While the Tool Was Detached.

- **Phenomenon**

The "SRVO-062 BZAL" and "SRVO-075 Pulse not established" alarms occurred when the tool was attached.

- **Remedy**

When the reference position has not been set:

— Press F3:RESET_PULSE in the Calibration screen, and select YES.

— Turn the controller OFF then ON again.

— Move the tool by manual feed to cancel the "SRVO-075 Pulse not established" alarm.

— Carry out tool 0 position mastering and calibration manually.

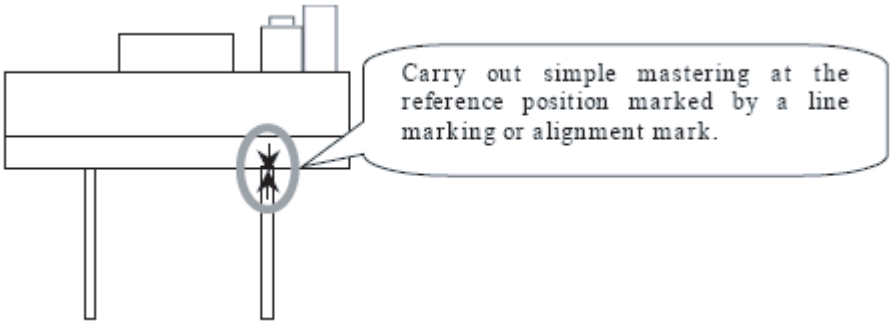
When the reference position has been set:

3. Press F3:RESET_PULSE in the Calibration screen, and select YES.
4. Turn the controller OFF then ON again.
5. Move the tool by manual feed to cancel the "SRVO-075 Pulse not established" alarm.
6. Move the tool by manual feed to the reference position marked by a line marking or alignment mark.
7. Execute 3 QUICK MASTERING in the Calibration screen.

*1) Touch torque

Increase the value from about ±2, and calculate the value at which erroneous detection of contact occurs. The sign of touch torque is the sign that is currently displayed at Touch torque at the Calibration Setup screen .

Note After confirming the sign in this screen, teach it to the program. Do not make a mistake when entering the sign. If you do so, contact with the open limit may not be detected.



Chapter 12

FANUC SENSOR INTERFACE

Contents

Chapter 12	FANUC SENSOR INTERFACE	12–1
12.1	OVERVIEW	12–2
12.2	COMMUNICATION PROTOCOL	12–2
12.2.1	Overview	12–2
12.2.2	Flow Control Mechanism	12–2
12.2.3	Data Format	12–3
12.2.4	Handshaking	12–5
12.3	PROGRAMMING	12–8
12.4	SENSOR SETUP AND HARDWARE CONNECTIONS	12–9

12.1 OVERVIEW

The FANUC Sensor Interface is an optional feature that enables the controller to communicate with an external sensor device through the RS-232-C serial port. The sensor device provides the controller with position offset data. The controller uses the position offset data for robot motion compensation.

Usually, a sensor has a coordinate system that differs from the coordinate system used by the robot. Because of this difference, the offset data must be transformed so the result reflects a valid offset for the coordinate system of the robot.

After the controller receives the offset data, you can use the offset data to compensate any taught position in your program.

Refer to Section 12.3 for more information about programming.

12.2 COMMUNICATION PROTOCOL

12.2.1 Overview

The FANUC Sensor Interface has a proprietary software flow control mechanism and uses its own data format. This communication protocol is similar to Xmodem protocol, but uses a very simple control sequence. It has a fixed data format and is equipped with a block checksum character for error detection.

12.2.2 Flow Control Mechanism

Four different flow control codes are available:

- ENQ
- ACK
- NAK
- EOT

ENQ (enquiry) 0x05 in hex representation

For every transmitted packet, the sending end sends an ENQ request to the receiving end to begin the data exchange. The sending end waits for the receiving end to respond with an ACK signal. If the sending end does not receive any acknowledgement from the receiving end within a predefined period of time, an error occurs.

ACK (acknowledgement) 0x06 in hex representation

The receiving end sends ACK when it is ready for data transmission. If the receiving end receives an ENQ but is not ready for data exchange, it sends a NAK.

NAK (not acknowledged) 0x95 in hex representation

The receiving end sends a NAK if it is not ready for data exchange when an ENQ arrives.

EOT (end of transmission) 0x84 in hex representation

After the sending end sends all the data in a packet, it notifies the receiving end that the packet sending session is finished by sending an EOT.

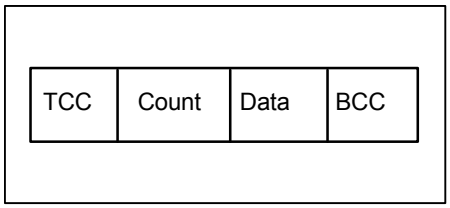
12.2.3 Data Format

The FANUC Sensor Interface recognizes five different fixed length packet formats:

- 0X55
- 0X5A
- 0X36
- 0X3A
- 0X6C

These packet formats are composed as shown in Figure 12–1 and are detailed in Table 12–1 .

Figure 12–1. Composition of FANUC Sensor Data Packet



TCCTransfer Control Character1 Byte

The receiving end uses the TCC to interpret the corresponding data in the packet.

Count1 Byte

The Count is the number of bytes that follows in the Data section. (BCC is not included.)

DataMultiple Bytes

The Data section represents the data to be exchanged. The Data section allows multiple bytes, but the number of bytes is fixed for each TCC.

BCCBlock Checksum Character1 Byte

The BCC is computed by taking Exclusive-Or (XOR) of all bytes in the Count and data section. This is an even parity checking mechanism for error detection.

Table 12–1. FANUC Sensor Data Formats

Function	TCC	Type	Count	Data	Units
Send the data contained in a register to the sensor	0X55	Output	3	Byte 1 to 3 is integer data.	None
Receive data for a register	0X5A	Input	4	Byte 1 is the register number. Bytes 2 to 4 are data for the register.	None
Receive data for a position register	0X36	Input	12	Byte 1 is the position register number. (PR[1] - PR[5]) Byte 2 to 4 is the x component. Byte 5 to 7 is the y component. Byte 8 to 10 is the z component. Byte 11 and 12 is the orientation. Orientation is not used.	The units for the x, y, and z components are 0.01 mm and the units for rotation are 0.01 deg. For example, X=0X00012C is equal to 4 mm.

Table 12–1. FANUC Sensor Data Formats (Cont'd)

Function	TCC	Type	Count	Data	Units
Receive 3 point data	0X3A	Input	28	Byte 1 indicates whether the data is the taught position or is the offset data. Byte 1-0: taught position Byte 1-1 offset data Byte 2 to 28 contain positional data for the x, y, z components of three positions.	0.01mm
Receive transformation matrix	0X6C	Input	36	Contains the elements of the homogenous transformation matrix: a11 a12 a13 a14 a21 a22 a23 a24 a31 a32 a33 a34 Each element occupies 3 bytes. The order of transmission is a11, a21, a31, ..., a14, a24, a34.	a*1, a*2, a*3 (*=1,2,3): 0.00001a*4 (*=1,2,3) : 0.01 mm

12.2.4 Handshaking

There are five handshaking sequences for each packet:

- Send the contents of a register to the sensor.
- Receive a register number and its data.
- Receive data for a position register.
- Receive three point data.
- Receive the transformation matrix.

Figure 12–2 through Figure 12–6 illustrate each of these handshaking sequences.

Figure 12–2. Handshaking When the Contents of a Register is Sent to the Sensor

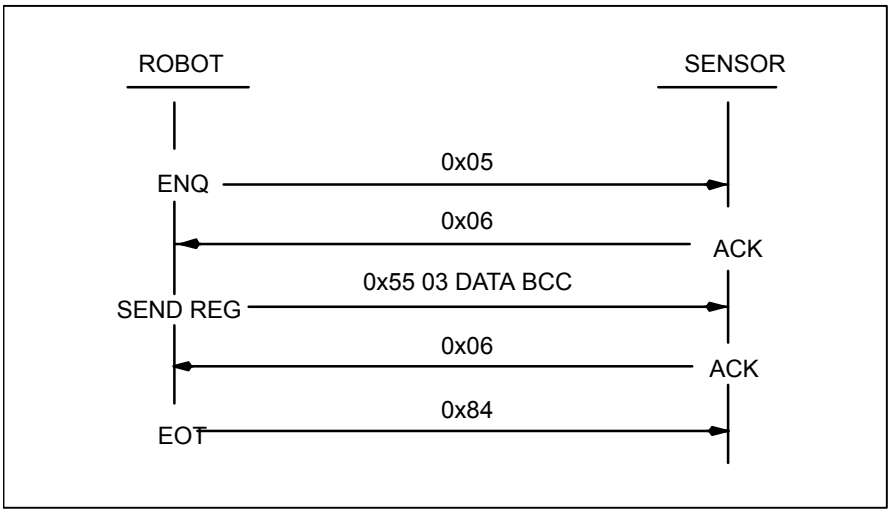


Figure 12–3. Handshaking When a Register Number and Its Data is Received

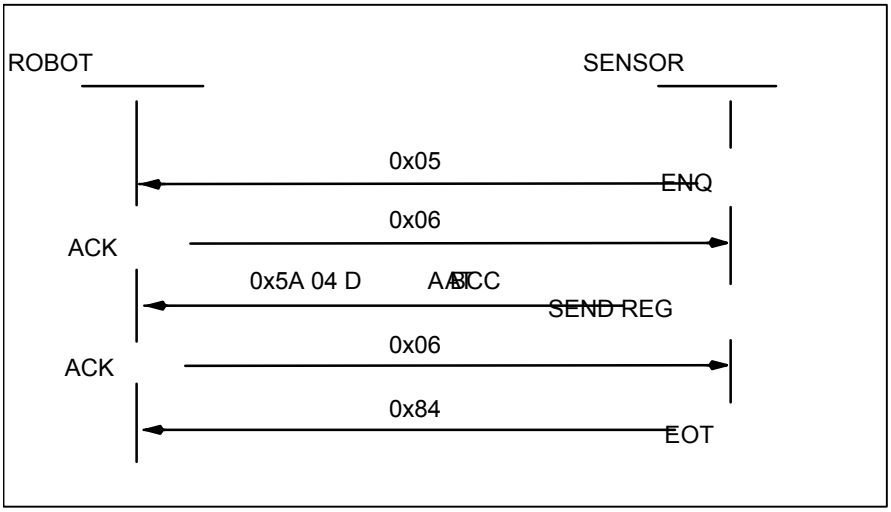


Figure 12–4. Handshaking When Position Register Data is Received

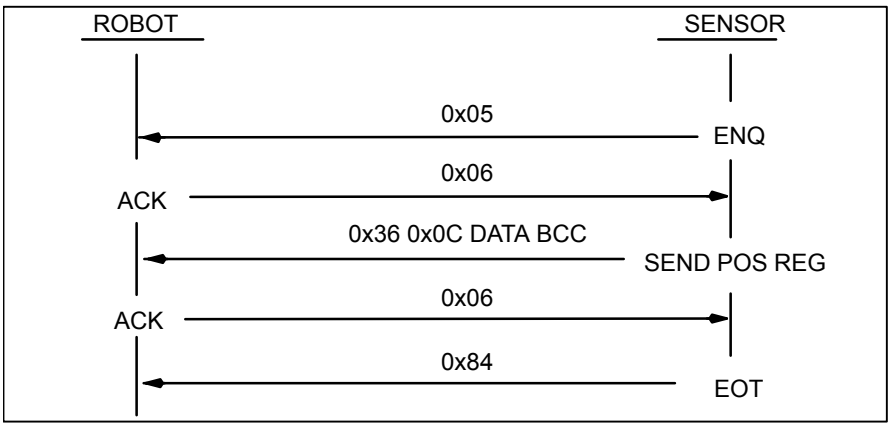
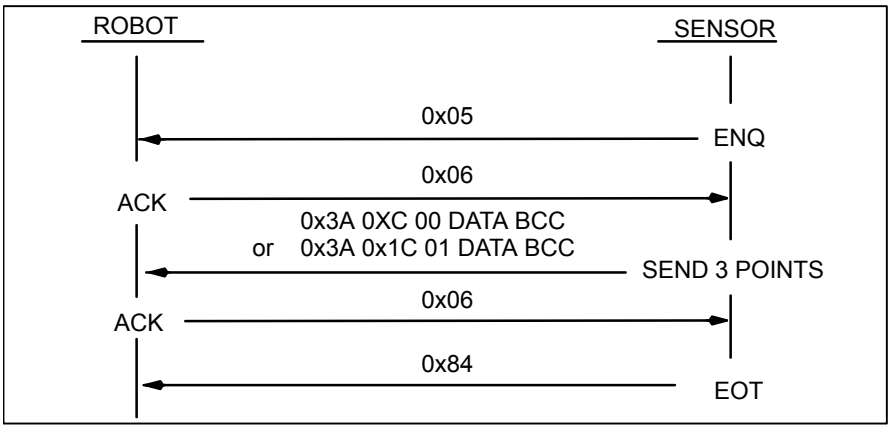
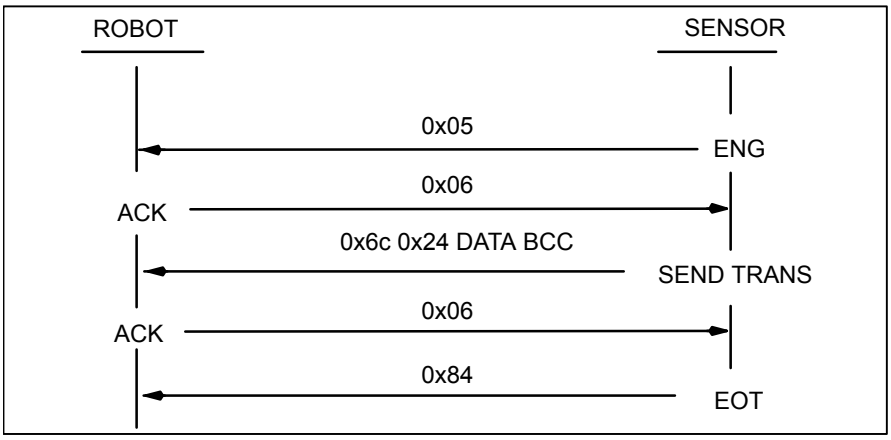


Figure 12–5. Handshaking When Three Point Data is Received



three point data transmitted from the sensor (V-120 vision system) is set in the position registers specified by the system variables $SENS_IF[ ].

Figure 12–6. Handshaking When the Transformation Matrix is Received



is set in the position register specified by the $SENS_IF[ ] system variables. Refer to Section 8.7 and the *Software Reference Manual* for more information on system variables.

12.3 PROGRAMMING

There are four program instructions associated with the FANUC sensor interface:

- SEND R[*]
- RCV R[*], LBL[*]
- RCV R[*] LBL[x] TIMEOUT, LBL[y]
- CALMATRIX

SEND R[*] sends out a packet with TCC = 0X5A. RCV R[*], LBL[*] waits for input until the register is set, then continues to the OFFSET condition instruction. If a communication error occurs, program execution will jump to LBL[1]. CALMATRIX computes the transformation matrix for offset data.

The OFFSET program element can then be used in your program to compensate the motion of any recorded robot position. Refer Chapter 6 *PROGRAM ELEMENTS* for more information about program instructions.

In the programming example shown in FANUC Sensor Interface Program Example , sensor will send all the offset data to the position registers. At the end, sensor set R[2] to 1 so program execution will continue. At that point, P[1], P[2], and P[3] will be offset by the sensor input.



Caution

Recorded positions are not affected by UFRAME and UFRAME has no affect during playback. However, **position registers are recorded with respect to UFRAME.** If you change UFRAME, any recorded position registers will also change.

FANUC Sensor Interface Program Example

```
1:  R[1] = 2
  2:  R[2] = 0
  3:  !Notify sensor to use R[2] to terminate comm
  4:  SEND R[1]
  5:  !Get the input, at the end set R[2] to 1
  6:  RCV R[2], LBL[1]
  7:  OFFSET CONDITION PR[1]
  8:  J P[1] 100% FINE OFFSET
  9:  J P[2] 100% FINE OFFSET
```

```
10:  J P[3] 100% FINE OFFSET
```

12.4 SENSOR SETUP AND HARDWARE CONNECTIONS

To use an external sensor device with the controller, you must:

- Set up the RS-232-C port.
- Connect the RS-232-C port to an external sensor.

You set up the port using the Port Init screen in the SETUP menu. The external sensor device uses the following default communications setting:

- 4800 baud
- 1 stop bit
- Odd parity

Refer to Section 9.1.4 for information on how to set up a port.

You use a NULL modem cable to connect the RS-232 port to a sensor device. Refer to Table 12–2 for the pin connector layout.

Table 12–2. NULL Modem Cable Pin Connector Layout

RS-232 Port	External Sensor
Pin 2 (TXD)	Pin 3 (RXD)
Pin 3 (RXD)	Pin 2 (TXD)
Pin 4 (RTS)	Pin 5 (CTS)
Pin 5 (CTS)	Pin 4 (RTS)
Pin 6 (DSR)	Pin 20 (DTR)
Pin 20 (DTR)	Pin 6 (RSD)

Chapter 13

MATERIAL HANDLING OPTION

Contents

Chapter 13	MATERIAL HANDLING OPTION	13–1
13.1	OVERVIEW	13–2
13.2	SETTING UP THE MATERIAL HANDLING TOOLS	13–3
13.2.1	Overview	13–3
13.2.2	Selecting the Handling Config (SpotTool+ only)	13–3
13.2.3	Setting Up Tooling Valves	13–6
13.2.4	Setting Up Tooling Valve Signals	13–12
13.2.5	Configuring Tooling Valve I/O	13–17
13.2.6	Controlling Valve I/O	13–19
13.3	CONTROLLING A TOOLING VALVE IN MANUAL MODE	13–22
13.4	MATERIAL HANDLING VALVE MACRO PROGRAMS	13–23
13.4.1	Examples	13–24
13.4.2	Grip Part	13–25
13.4.3	Release Part	13–27
13.4.4	Part Present	13–29
13.4.5	Check No Part	13–30
13.4.6	Prepare to Pick	13–31
13.4.7	Clear to Proceed	13–32
13.4.8	Turn ON Vacuum	13–33
13.4.9	Turn OFF Vacuum	13–36
13.4.10	Turn OFF Blowoff	13–37
13.4.11	Set CurrentValve	13–39
13.4.12	Set MH Tool	13–40
13.5	PRODUCTION	13–41
13.5.1	MH Tryout Mode	13–41
13.5.2	Disable for 20 Cycles	13–41
13.5.3	Alarms	13–42

13.1 OVERVIEW

Note Some descriptions in this chapter might apply only to the use of this option with the SpotTool+ product and not the HandlingTool product.

The Material Handling option provides the controller with the ability to set up, configure, and control the functioning of vacuum or clamp-style material handling valves. You define the tooling valve type, characteristics, and I/O. Then you can control the tooling valve using specific macro commands that are called from within a teach pendant program, run from the Manual Functions screen, or run by pressing a teach pendant user key.

The following terms are used in the descriptions of setting up and operating the Material Handling Tool. Review them for a better understanding of the Material Handling Option.

Tooling Valve

This is a "gripper-like" mechanism.

Vacuum Tool Valve

This is a mechanism that picks up objects using vacuum suction cups.

Clamp Tool Valve

This is a mechanism that picks up objects using clamps.

Part Present

This is a sensor that indicates whether an object is being held by a tool valve.

Vacuum Input

This is a sensor that indicates whether the vacuum suction is on.

Valve n to A

This is a signal that indicates whether the tooling valve is in the A position. When the signal is on, the tooling valve is in the A position. The A and B positions must be opposite.

Valve n to B

This is a signal that indicates whether the tooling valve is in the B position. When the signal is on, the tooling valve is in the B position. The A and B positions must be opposite.

13.2 SETTING UP THE MATERIAL HANDLING TOOLS

13.2.1 Overview

You set up the Material Handling Tool by doing the following:

1. Select Handling Config during Controlled start (for SpotTool+ only).
2. Set up tooling valve information.
3. Set up tooling valve signals.
4. Configure tooling valve I/O.

After you have set up this information, you can also control valve I/O.

13.2.2 Selecting the Handling Config (SpotTool+ only)

If you have SpotTool+, before you can set up and use the Material Handling Tools, you must do the following:

- Select and configure the material handling application at Controlled start - Procedure 13-1
- Select the material handling application as the current application for robot operation - Procedure 13-2

Procedure 13-1 Selecting and Configuring the Handling Config at Controlled Start

Steps

1. Perform a Controlled start.

a. **If the controller is turned on**, press FCTN and select CYCLE POWER. Select YES and press ENTER. **If the controller is OFF**, turn on the power disconnect circuit breaker.

b. When the BMON screen is displayed on the teach pendant, press and hold the PREV and NEXT keys. You will see a screen similar to the following.

```
---------- Configuration Menu ---------
1 Hot start
2 Cold start
3 Controlled start
4 Maintenance
Select >
```

c. Select Controlled start and press ENTER. You will see a screen similar to the following.

```
SPOT CONFIG             CONTROLLED START MENUS
                                          1/50
 1 F Number:                            F00000 >
 2 Load Spottool Macros:                DISABLED
 3 Number of equipments:                1
 4 Number of weld controllers:          1
 5 Current equipment, gun               1, 1
 6   Weld interface:                    DIGITAL
 7   Number of guns:                    1
 8   Studwelder:                        DISABLED
 9     Enable Backup gun:               DISABLED
10  Tool changer gun:                   DISABLED
11  Soft touch gun:                     DISABLED
Press FCTN then START (COLD) when done
```

2. Select the application:

a. Press MENUS.

b. Select Handling Config. You will see a screen similar to the following.

```
Appl selection CONTROLLED START MENUS  1/3
1 Spot Weld                         TRUE
2 Handling                          FALSE
3 Dispense                          FALSE
```

c. Move the cursor to Handling and press F4, TRUE.**To disable an application**, press F5, FALSE.

Note If you have the Multiple Application option in SpotTool+, you can enable up to three loaded applications on the controller.

Note After you have enabled an application, you can use the enabled application instructions in a teach pendant program.

3. Configure the Handling application:

a. Press MENUS.

b. Select Handling Config. You will see a screen similar to the following.

```
Handling Signals          CTRL START MENU
1 Number of MH Tools:                1  <--See NOTE
2 Number of Valves per Tool          4  <--See NOTE
3 Number of Clamp Inputs:              34
4 Number of Part Present Inputs:       12
5 Number of Vacuum Inputs:         4
6 Use Active Tool for Arguments:       NO
```

Note The combination of these two items is restricted by size and can be determined by multiplying one by the other. Note, however, that the resulting value cannot exceed 352. For example, 29 tools by 12 valves (348) would be the limit for that setup. As a second example, 22 tools by 16 valves (352) would be the limit for that setup.

c. Set each item as desired, and press ENTER. Refer to Table C–5 .

Table 13–1. Handling Config Menu Options

Item	Description
Number of MH Tools	Use this item to configure the number of tools you need for your system. The maximum number of tools is 29. At cold start, you will then have the specified number of tools available in the MH Valves menu under Setup.
Number of Valves Per Tool	Use this item to configure the number of valves you have in your tool. The maximum number of valves is 29. At cold start, you will then have the specified number of valves available in the **MH Valves** menu under Setup. You can then configure any valve to be a vacuum or clamp style tool and set up the required I/O for each.
Number of Clamps	Use this item to configure the number of clamps you need for your tool. The maximum number of clamps is 64. At Cold start, you will then have the specified number of clamps available in the MH Valves menu under Setup to assign to a valve.
Number of Part Present Inputs	Use this item to configure the number of Part Present Inputs you need for your tool. the maximum number of Part Present Inputs is 29. At cold start, you will then have the specified number of Part Present Inputs available in the MH Valves menu under Setup to assign to a valve.
Number of Vacuum Inputs	Use this item to configure the number of Vacuum Inputs you need for your tool. the maximum number of Vacuum Inputs is 29. At cold start, you will then have the specified number of Vacuum Inputs available in the MH Valves menu under Setup to assign to a valve.
Use Active Tool for Arguments	Use this item to change the behavior of the tool macros when no parameters are passed to them. When this is set to YES, the number of valves set in the tool SETUP are used to operate that number of valves for that tool. When set to NO, they act like the original specification.

Procedure 13-2 Selecting the Current Application for Robot Operation

Conditions

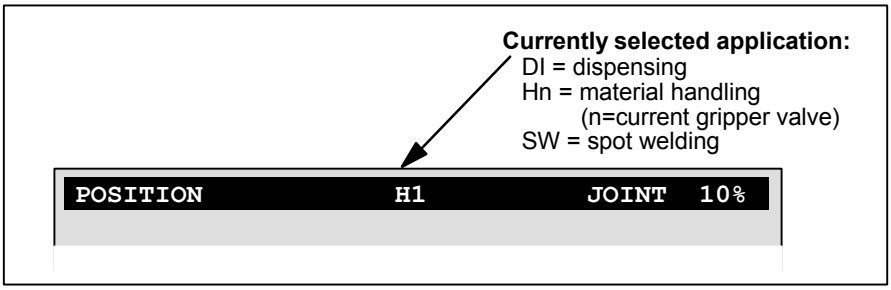
- The applications you want have been enabled at controlled start. Refer to the *Software Installation Manual* for more information.

Steps

1. Cold start the controller.
   a. Press FCTN and select CYCLE POWER. Select YES, and press ENTER.
   b. **When the BMON screen is displayed on the teach pendant**, press and hold the SHIFT and RESET keys. Or, **on the operator panel** , press and hold RESET.
   c. Release all of the keys.
2. Press the FCTN key.
3. Select CHANGE APPL-TOOL.

   The next enabled application will be selected. The currently selected application is displayed in the middle of the screen name line on the teach pendant screen, as shown in Figure 13–1 .

Figure 13–1. Currently Selected Application



4. Repeat Step 2 and Step 3 until the material handling application is selected.

13.2.3 Setting Up Tooling Valves

You can use from 1 to *x* Tooling valves, depending on how many you enabled at Controlled start (Procedure 13-1). Table 13–2 lists the items you set up. Use Procedure 13-3 to set up tool valves.

Note You cannot change any of the tooling valve setup information when the controller is in AUTO mode.

Table 13–2. Material Handling Setup Screen Items

ITEM	DESCRIPTION
Tool	This item indicates which tool you are using.
Vacuum Valve	
Valve #*n*	This item is the number and name of the valve you are setting up.
Valve Type	This item specifies the type of valve, which is VACUUM.
Valve Type	This item specifies the type of valve, which is VACUUM.
No. of Vacuum Sensor min: 0 max: 18	This item specifies the number of Vacuum sensors.
Vac Feedback default: NO	This item indicates whether the valve has a feedback sensor. • YES means the valve has a feedback sensor • NO means the valve does not have a feedback sensor.
Ignore Feedback min: 0 ms max: 10000 ms default: 750 ms	When Vac Feedback is YES, this item specifies the amount of time to wait after the vacuum was turned on to start checking whether feedback has been received.
Continuous Check default: NO	When Vac Feedback is YES, this item specifies whether to check that feedback stays on until the vacuum is turned off. • YES checks that feedback stays on until the vacuum is turned off. • NO does not check that feedback stays on until the vacuum is turned off.
Max Feedback Delay min: 0 ms max: 10000 ms default: 750 ms	When Vac Feedback is YES, this item specifies the amount of time to wait before giving an alarm if no feedback has been received. When Vac Feedback is NO, this item specifies the amount of time to wait after turning on or off the vacuum to make sure the vacuum is in the desired state before continuing.
Blow off time min: 0 ms max: 10000 ms default: 1000 ms	When releasing a part manually, this item specifies the amount of time to pulse the vacuum valve to position B.

Table 13–2. Material Handling Setup Screen Items (Cont'd)

ITEM	DESCRIPTION
No. of parts present min: 0 max: 6 default: 0	This item specifies the number of parts present sensors used by this valve. **NOTE:** If you change the number of parts present, the Material Handling Valve I/O setup will be changed. Increasing the number of parts present will add an entry with default data on the tool I/O screen. The existing data will be left unchanged. The newly added data will remain unchanged. Decreasing the number of parts present will remove the last entry for that data type. The rest of the data will remain unchanged.
Continuous Check default: NO	After a part has been picked up, this item specifies whether to check that the part present sensor stays on until the part is put back down. • YES checks that part present sensor stays on until a picked up part is put back down. • NO does not check that the part present sensor stays on until the part is put back down.
Clamp Valve	
Tool	This item indicates which tool you are using.
Valve #*n*	This item is the name of the valve you are setting up.
Valve Type	This item specifies the type of valve, which is CLAMP.
No. of clamps min: 0 max: 4 default: 0	This item specifies the number of clamp sensor pairs used by this clamp valve. **NOTE:** If you change the number of clamps, the Material Handling Valve I/O setup will be changed. Increasing the number of clamps will add an entry with default data on the tool's I/O screen. The existing data will be left unchanged. The newly added data will remain unchanged. Decreasing the number of clamps will remove the last entry for that data type. The rest of the data will remain unchanged.
Check Opened default: NO	Before a part has been picked up and after a part has been put down, this item specifies whether to automatically check that the valve's clamps are all open.
Check Closed default: NO	After a part has been picked up, this item specifies whether to automatically check that the valve's clamps are closed.
Operation Delay min: 0 ms max: 10000 ms default: 750 ms	After a part has been picked up or put down, this item specifies the amount of time to wait to check the clamp state.

Table 13–2. Material Handling Setup Screen Items (Cont'd)

ITEM	DESCRIPTION
No. of parts present min: 0 max: 6 default: 0	This item specifies the number of parts present sensors used by this valve. **NOTE:** If you change the number of parts present, the Material Handling Valve I/O setup will be changed. Increasing the number of parts present will add an entry with default data on the tool's I/O screen. The existing data will be left unchanged. The newly added data will remain unchanged. Decreasing the number of parts present will remove the last entry for that data type. The rest of the data will remain unchanged.
Continuous Check default: NO	After a part has been picked up, this item specifies whether to check that the part present sensor stays on until the part is put back down. • YES checks that part present sensor stays on until a picked up part is put back down. • NO does not check that the part present sensor stays on until the part is put back down.
Undefined Valve (*****)	
Tool	This item indicates which tool you are using.
Valve *#n*	This item is the name of the valve you are setting up.
Valve Type	This item specifies the type of valve, which is NONE.

Procedure 13-3 Setting Up an MH Tooling Valve

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select MH Valves. You will see a screen similar to the following.

```
SETUP MH Tool
     Select tool to setup       1/1
     TOOL    VALVES      NAME
       1        1           TOOL 1
```

5. Select the Tool you are using, and press ENTER. You will see a screen similar to the following.

```
SETUP Mat.Handling
Select valve to setup    Tool: 1          1/3
VALVE     TYPE      VALVE NAME
   1      CLAMP     DOOR RING
   2      VACUUM    HINGE
   3      *****     VALVE3
```

6. Move the cursor to the valve you want to set up and press F2, DETAIL.

If you selected a **vacuum** type valve, you will see a screen similar to the following.

```
SETUP Mat. Handling
Setup Vacuum Information  Tool: 1  1/10

  1 Valve #1 :  [         VALVE 1]
  2 Valve Type:            Vacuum
  3  No. of Vaccum Sensor:  8
  4  Vac Feedback:           NO
  5   Ignore Feedback:       750 (ms)
  6   Continuous Check:      NO
  7  Max Feedback Delay:     750 (ms)
  8  Blow off time:         1000 (ms)
  9  No. of parts present:     4
 10   Continuous Check:      NO
```

If you select a **clamp** type valve, you will see a screen similar to the following.

```
SETUP Mat. Handling
Setup Clamp Information                  1/8
 1  Valve #1     [          DOOR RING]
 2  Valve Type:                Clamp
 3  No. of clamps:                 3
 4  Check Opened:                YES
 5  Check Closed:                YES
 6  Operation Delay:            1000 (ms)
 7  No. of parts present:          2
 8  Continuous Check:            YES
```

If you select a ***** (NONE) type valve, you will see a screen similar to the following.

```
SETUP Mat. Handling
Setup Valve Information                    1/2
  1  Valve #3        [         VALVE 3]
  2  Valve Type:                   None
```

Note You cannot change any of the valve setup information when the controller is in AUTO mode.

7. Move the cursor to each item and set as desired.

Note If you change the number of clamps or number of parts present, the Config setup will be changed when you press APPLY, as follows:

- Increasing the number of clamps or number of parts present will add an entry with default data on the valve's I/O screen. The existing data will be left unchanged. The newly added data will remain unchanged.
- Decreasing the number of clamps or number of parts present will remove the last entry for that data type. The rest of the data will remain unchanged.

8. **When you are finished** making changes to the setup information and want to update the settings, press F3, APPLY.
9. **To display detail information for another valve**, press F2, VALVE#, type the number of the valve you want, and press ENTER.
10. **To save all of the MH valve settings to the MHGRIPDT.VR file on the default device,**

- To continue the save, select YES and press ENTER.
- To cancel the save, select NO and press ENTER.

Note If the file MHGRIPDT.VR file already exists on the default device, you will be asked whether to overwrite the file.

- To overwrite the MHGRIPDT.VR file on the default device, select YES and press ENTER.
- To cancel the save, select NO and press ENTER.

11. **To restore and MHGRIPDT.VR file from the default device to controller memory** (load will overwrite the current MHGRIPDT.VR file in controller memory),
    a. Press NEXT, >, and then press F2, LOAD.
    b. Confirm the restoration of the MHGRIPDT.VR file:
        - To continue the restoration, select YES and press ENTER.

- To cancel the restoration, select NO and press ENTER.

12. **To set up the signals used by the tooling valves,** refer to Section 13.2.4 .

13.2.4 Setting Up Tooling Valve Signals

You need to set up the following valve signals to use the Material Handling Tool:

- Clamp inputs
- Part present inputs
- Vacuum inputs
- Valve outputs

The type of signals you need to set up depends on the type of valve you have and the number of valves you defined at Controlled start.

Table 13–3 and Table 13–4 list these I/O signals. Use Procedure 13-4 to set up valve signals.

Table 13–3. Valve Signals

I/O SIGNAL	DESCRIPTION
Clamp inputs - signals referred to by name on the I/O MatHandling screens	
Clamp*n* open *n* = 1 to 64	This signal indicates that the specified clamp *n* is open. You can give the signal any 16-character name. The signal can be either an RI or DI.
Clamp*n* closed *n* = 1 to 64	This signal indicates that the specified clamp *n* is closed. You can give the signal any 16-character name. The signal can be either an RI or DI.
Part present (PP) inputs - signals referred to by name on the I/O MatHandling screens	
Part present*nn* = 1 to 29	This signal indicates that the part present sensor is on. You can give the signal any 16-character name. The signal can be either an RI or DI.
Vacuum (VacMade) inputs - signals referred to by name on the I/O MatHandling screens	
Vacuum*n* made *n* = 1 to 29	This signal indicates that the vacuum has been made for the specified tool ***n*** . *n* is any of the tools that were defined at Controlled start. You can give the signal any 16-character name.
Valve outputs - signals referred to by name on the I/O MatHandling screens	
Valve*n* to A *n* = 1 to 29	This signal indicates that the valve is in the A position. You can give the signal any 16-character name.
Valve*n* to B *n* = 1 to 29	This signal indicates that the valve is in the B position. You can give the signal any 16-character name.

Table 13–4. Valve DETAIL Screen Items

ITEM	DESCRIPTION
Vacuum Valve	
Valve #*n*	This item is the name of the valve you are setting up.
Valve type	This item specifies the [TYPE] of valve, which is VACUUM.
Vac Feedback Default: NO	This item indicates whether the vacuum valve has a feedback sensor. • YES means the vacuum valve has a feedback sensor. • NO means the vacuum valve does not have a feedback sensor.
Ignore Feedback Min: 0 ms Max: 10000 ms Default: 750 ms	When Vac Feedback is YES, this item specifies the amount of time to wait after the vacuum was turned on to start checking whether feedback has been received.
Continuous Check Default: NO	When Vac Feedback is YES, this item specifies whether to check that feedback stays on until the vacuum is turned off. • YES checks that feedback stays on until the vacuum is turned off. • NO does not check that feedback stays on until the vacuum is turned off.
Max Feedback Delay Min: 0 ms Max: 10000 ms Default: 750 ms	When Vac Feedback is YES, this item specifies the amount of time to wait before giving an alarm if no feedback has been received. When Vac Feedback is NO, this item specifies the amount of time to wait after turning on or off the vacuum to make sure the vacuum is in the desired state before continuing.
Blow off time Min: 0 ms Max: 10000 ms Default: 1000 ms	When releasing a part manually, this item specifies the amount of time to pulse the vacuum valve to position B.
No. of parts present Min: 0 Max: Default: 0	This item specifies the number of parts present sensors used by this valve. **Note** If you change the number of parts present, the Material Handling Valve I/O setup will be changed. Increasing the number of parts present will add an entry with default data on the tool's I/O screen. The existing data will be left unchanged. The newly added data will remain unchanged. Decreasing the number of parts present will remove the last entry for that data [TYPE]. The rest of the data will remain unchanged.

Table 13–4. Valve DETAIL Screen Items (Cont'd)

Continuous Check Default: NO	After a part has been picked up, this item specifies whether to check that the part present sensor stays on until the part is put back down. • YES checks that part present sensor stays on until a picked up part is put back down. • NO does not check that the part present sensor stays on until the part is put back down.
Clamp Valve	
Valve #*n*	This item is the name of the valve you are setting up.
Valve type	This item specifies the [TYPE] of valve, which is CLAMP.
No. of clamps Min: 0 Max: Default: 0	This item specifies the number of clamp sensor pairs used by this clamp valve. **Note** If you change the number of clamps, the Material Handling Valve I/O setup will be changed. Increasing the number of clamps will add an entry with default data on the tool's I/O screen. The existing data will be left unchanged. The newly added data will remain unchanged. Decreasing the number of clamps will remove the last entry for that data [TYPE]. The rest of the data will remain unchanged.
Check Opened Default: NO	Before a part has been picked up and after a part has been put down, this item specifies whether to automatically check that the valve's clamps are all open.
Check Closed Default: NO	After a part has been picked up, this item specifies whether to automatically check that the valve's clamps are closed.
Operation Delay Min: 0 ms Max: 10000 ms Default: 750 ms	After a part has been picked up or put down, this item specifies the amount of time to wait to check the clamp state.
No. of parts present Min: 0 Max: Default: 0	This item specifies the number of parts present sensors used by this valve. **Note** If you change the number of parts present, the Material Handling Valve I/O setup will be changed. Increasing the number of parts present will add an entry with default data on the tool's I/O screen. The existing data will be left unchanged. The newly added data will remain unchanged. Decreasing the number of parts present will remove the last entry for that data [TYPE]. The rest of the data will remain unchanged.
Continuous Check Default: NO	After a part has been picked up, this item specifies whether to check that the part present sensor stays on until the part is put back down. • YES checks that part present sensor stays on until a picked up part is put back down. • NO does not check that the part present sensor stays on until the part is put back down.
Undefined Valve (*****)	

Table 13–4. Valve DETAIL Screen Items (Cont'd)

Valve #*n*	This item is the name of the valve you are setting up.
Valve [TYPE]	This item specifies the [TYPE] of valve, which is NONE.

Table 13–5. SETUP MH Tool Signal DETAIL Screen Items

ITEM	DESCRIPTION
ID #	This item is the number which is assigned to the signal.
Input or Output Signal	This item represents the signal being used.
Type	This item represents the [TYPE] of digital input/output that is assigned.
Number	This item represents the digital input or output number.

Procedure 13-4 Setting Up Tooling Valve Signals

Conditions

- You have defined tools at Controlled Start. Refer to the *Software Installation Manual*

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select MH Tool. You will see a screen similar to the following.

```
SETUP Mat. Handling
Select valve to setup  Tool: 1          1/2
VALVE    TYPE     VALVE NAME
  1      CLAMP    DOOR RING
  2      VACUUM   HINGE
```

5. Press F3, SIGNALS. You will see a screen similar to the following.

```
SETUP MH CONFIG
         I/O TYPES
    1    Clamp Inputs
    2    Part Present Inputs
    3    Vacuum Inputs
    4    Valve Outputs
```

6. Move the cursor to the type of I/O you want to set up and press F2, DETAIL.

- For **clamp inputs**, you will see a screen similar to the following.

```
SETUP Mat. Handling
Clamp Signals for all valves          1/40
ID#      INPUT SIGNAL          TYPE #
  1      Clamp 1 open          DI[ 425]
  1      Clamp 1 closed        DI[ 426]
  2      Clamp 2 open          DI[ 427]
  2      Clamp 2 closed        DI[ 428]
  3      Clamp 3 open          DI[ 429]
  3      Clamp 3 closed        DI[ 430]
  4      Clamp 4 open          DI[   0]
  4      Clamp 4 closed        DI[   0]
  5      Clamp 5 open          DI[   0]
```

- For **part present inputs**, you will see a screen similar to the following.

```
SETUP Mat. Handling
PP Signals for all valves              1/10
ID#      INPUT SIGNAL          TYPE #
  1      Part present 1        DI[ 437]
  2      Part present 2        DI[ 438]
  3      Part present 3        DI[ 439]
  4      Part present 4        DI[   0]
  5      Part present 5        DI[   0]
  6      Part present 6        DI[   0]
  7      Part present 7        DI[   0]
  8      Part present 8        DI[   0]
  9      Part present 9        DI[   0]
```

- For **vacuum inputs**, you will see a screen similar to the following.

```
SETUP Mat. Handling
VacMade Inputs for all valves          1/2
ID#      INPUT SIGNAL          TYPE #
  1      VACUUM 1 MADE         DI[ 440]
  2      VACUUM 2 MADE         DI[   0]
```

- For **valve outputs**, you will see a screen similar to the following.

```
SETUP Mat. Handling
Setup Valve Signals                    1/20
ID#      OUTPUT SIGNAL          TYPE #
  1      Valve 1 to A           DO[ 429]
  1      Valve 1 to B           DO[ 430]
  2      Valve 2 to A           DO[ 425]
  2      Valve 2 to B           DO[ 426]
  3      Valve 3 to A           DO[   0]
  3      Valve 3 to B           DO[   0]
  4      Valve 4 to A           DO[   0]
  4      Valve 4 to B           DO[   0]
  5      Valve 5 to A           DO[   0]
```

7. Select the signals you want to set and set them as desired. You can change the signal name, type, and digital input or output number.
8. After you have set up the signals for each valve, you can configure the I/O for each valve. Refer to Section 13.2.5 .

13.2.5 Configuring Tooling Valve I/O

When you configure tooling valve I/O, you specify which of the signals you set up in Section 13.2.4 will be used with each of the valves you set up in Section 13.2.3 . Use Procedure 13-5 to configure valve I/O.

Procedure 13-5 Configuring Valve I/O

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select MH Valves.
5. Move the cursor to the valve for which you want to configure I/O and press F2, DETAIL. You will see a screen similar to the following.

```
SETUP Mat. Handling
Setup Clamp Information  Tool: 1          1/8
  1  Valve #1          [     DOOR RING]
  2  Valve Type:                Clamp
  3  No. of clamps:                 2
  4  Check Opened:                YES
  5  Check Closed:                YES
  6  Operation Delay:            1000 (ms)
  7  No. or parts present:          1
  8  Continuous Check:            YES
```

6. Press F5, CONFIG.

For an output signal, you will see a screen similar to the following.

```
SETUP MHandling
Valve 1:  DOOR RING
    TYPE     OUTPUT SIGNAL      TYPE #
  1 Valve 1 to A               DO[   14]
  2 Valve 1 to B               DO[   15]
Cannot change a valve's output signal
```

Note The items on the output screen are for display only; you cannot make any changes on this screen.

For an input signal, you will see a screen similar to the following.

```
SETUP MHandling
Valve 1:  DOOR RING
    TYPE     INPUT SIGNAL       TYPE #
  1 CLOPEN:Clamp 1 open        DI[    0]
  2 CLCLSD:Clamp 1 closed      DI[   11]
  3 CLOPEN:Clamp 20 open       DI[    0]
  4 CLCLSD:Clamp 20 closed     DI[    0]
  5 PARTPR:Part present 4      DI[   45]
Select the signals used by this valve
```

In this example screen, the clamp valve has two clamp pairs and one part present.

7. To select the input signal you want to configure,

a. Move the cursor to the input signal you want and press F4, [CHOICE].

A subwindow will be displayed that contains the signal names you set up in Section 13.2.4 . See the following screen for an example.

```
SW
  1 Unassigned PP          5  Part Present 4
  2 Part present 1         6  Part present 5
  3 Part present 2         7  Part present 6
  4 Part present 3         8-- NEXT --
  4 CLCLSD:  Clamp 20 closed        DI[   0]
  5 PARTPR:  Unassigned PP          DI[   0]
Select the signals used by this valve
```

b. Select the appropriate input signal name.

The input TYPE and # columns will be filled in automatically based on the signal name you selected.

13.2.6 Controlling Valve I/O

Afte r you have set up and configured valve I/O, you can monitor, simulate, and force I/O. Use Procedure 13-6 to control valve I/O.

Procedure 13-6 Controlling Valve I/O

Steps

1. Press MENUS.
2. Select I/O.
3. Press F1, [TYPE].
4. Select MH Valves.

- For**clamp inputs**, you will see a screen similar to the following.

```
I/O MatHandling DI
Valve 1:  DOOR RING
    INPUT SIGNAL        TYPE #  SIM  STATUS
  1 Clamp 1 open        DI[ 425] U   ***
  2 Clamp 1 closed      DI[ 426] U   ***
  3 Clamp 2 open        DI[ 427] U   ***
  4 Clamp 2 closed      DI[ 428] U   ***
  5 Clamp 3 open        DI[ 429] U   ***
  6 Clamp 3 closed      DI[ 430] U   ***
  7 Part present 1      DI[ 437] U   ***
  8 Part present 2      DI[ 438] U   ***
```

- For**clamp outputs**, you will see a screen similar to the following.

```
I/O MatHandling DO
Valve 1:  DOOR RING
    OUTPUT SIGNAL        TYPE # SIM  STATUS
  1 Valve 2 to A         DI[425] U   ***
  2 Valve 2 to B         DI[426] U   ***
```

- For**vacuum inputs**, you will see a screen similar to the following.

```
I/O MatHandling DI
Valve 2:  HINGE
    INPUT SIGNAL        TYPE # SIM  STATUS
  1 Vacuum 1 made       DI[440] U   ***
  2 Part present 3      DI[439] U   ***
```

- For**vacuum outputs**, you will see a screen similar to the following.

```
I/O MatHandling DO
Valve 2:  HINGE
    OUTPUT SIGNAL       TYPE # SIM  STAT
  1 Valve 2 to A        DO[425] U   ***
  2 Valve 2 to B        DO[426] U   ***
```

5. **To change the display between inputs and outputs**, press F3, IN/OUT.
6. **To display I/O information for another valve**, press F2, VALVE#, type the number of the valve you want, and press ENTER.

   Note You cannot change the simulation of a signal, or turn the signal ON or OFF from the teach pendant, when the controller is in AUTO mode.

7. **To simulate a signal**,
   a. Move the cursor to the SIM column of the signal you want to simulate.
      - **U** means the signal is not simulated or unsimulated.
      - **S** means the signal is simulated.
   b. Simulate or unsimulate the signal.
      - **To simulate**, press F4, SIMULATE.

- **To unsimulate**, press F5, UNSIM.

c. **To unsimulate all simulated signals** press FCTN and then select UNSIM ALL I/O.

8. To turn a signal ON or OFF,

a. Move the cursor to the STATUS column.

Note If you want to turn ON or OFF an input signal, you must first simulate the input. If you want to turn ON or OFF an output signal, it can be either simulated or unsimulated.

b. Turn ON or OFF the signal:

- **To turn ON the signal**, press F4, ON.

- **To turn OFF the signal**, press F5, OFF.

13.3 CONTROLLING A TOOLING VALVE IN MANUAL MODE

You can operate a tooling valve manually using teach pendant user keys when Handling is the currently selected application. The status screen in Figure 13–2 shows how the user keys are used for HandlingTool.

Figure 13–2. HandlingTool User Keys

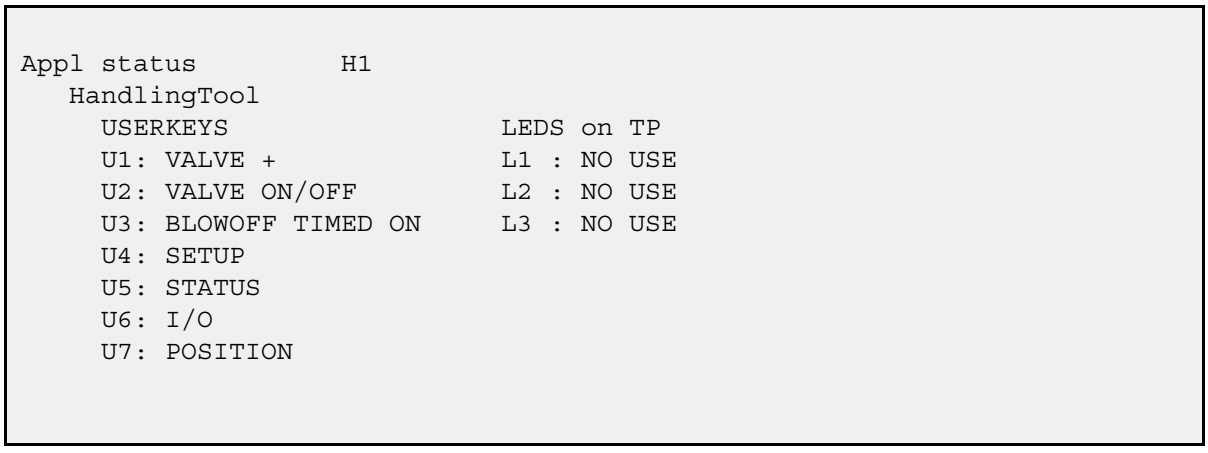


Table 13–6 lists and describes the manual operations you can perform using the user keys.

Table 13–6. Handling Manual Operations

USER KEY	DESCRIPTION OF OPERATION
Teach Pendant Enabled SHIFT + USER KEY 1	The next defined tooling valve becomes the currently selected valve. It is displayed on the teach pendant status line as "H," followed by a number that corresponds to the valve number.
Teach Pendant Enabled SHIFT + USER KEY 2	The state of the currently selected tooling valve is changed from position A to position B, or from position B to position A.
Teach Pendant Enabled SHIFT + USER KEY 3	If the currently selected tooling valve is a vacuum valve, the blowoff is timed on for the Blow off time specified on the SETUP Mat. Handling screen (Section 13.2.3).

13.4 MATERIAL HANDLING VALVE MACRO PROGRAMS

The macro programs listed in Table 13–7 are provided with the Material Handling option. You can call these macros from other teach pendant programs or invoke them from the Manual Functions screen.

Table 13–7. Material Handling Valve Macro Programs

Instruction Name	Program
Grip Part	GRIPPART
Release Part	DROPPART
Part Present	CHKPART
Check No Part	CHKNOPRT
Prepare to Pick	OKTOPICK
Clear to Proceed	CLR2PRCD
Turn ON Vacuum	VACUUMON
Turn OFF Vacuum	VACUUMOF
Turn OFF Blowoff	BLOWOFF
Set CurrentValve	SETVALVE
Set MH Tool	SET_TOOL

Note the following macro information:

- All of the macros are system macros.
- You set up all of the tooling valve control variables on the SETUP Mat. Handling screens (Section 13.2.3).
- You set the current tooling valve using SHIFT USERKEY 1, or using the Set CurrentValve macro. When a macro is called with no parameters, the current tooling valve is used by default.
- You can disable each MH PAUSE alarm using the Disable for 20 Cycles feature (Section 13.5.2).

Note There are no material handling program instructions; macros are used instead. You can combine material handling macros with either SpotTool+ application program instructions or DispenseTool application program instructions if you have one of these plug-ins installed.

Note Refer to Section 3.3 for information on setting up and executing macros.

13.4.1 Examples

Teach Pendant Program with Material Handling Clamp Valve Macros shows an example teach pendant program that uses Material Handling clamp valve macros. Teach Pendant Program with Material Handling Vacuum Valve Macros shows an example teach pendant program that uses Material Handling vacuum valve macros.

Teach Pendant Program with Material Handling Clamp Valve Macros

```
; Move above part 1
Prepare to Pick(1)          ** Check all clamps open for VALVE #1
; Move to part
Part Present(1)             ** Check part present #1 is ON
Grip Part(1)                ** Shift valve for VALVE #1 to A Position
; Move above part 2
Prepare to Pick(2)          ** Check all clamps open for VALVE #2
; Move to part
Part Present(3)             ** Check part present #3 is ON
Grip Part(2)                ** Shift valve for VALVE #2 to A Position
; Move to dropoff
Release Part(1,2)           ** Shift VALVEs #1 and #2 to B Position
; Move above release
Check No Part(1,2,3)        ** Check part presents #1, #2, and #3
to be OFF
; Move to home
```

Teach Pendant Program with Material Handling Vacuum Valve Macros

```
; Move above part 1
Prepare to Pick(1)          ** Check vacuum made for VALVE #1 is OFF
; Move to part
Part Present(4)             ** Check part present #4 is ON
Turn ON Vacuum(1)           ** Turn on VALVE 1A and wait for VacMade
1 to be ON
; Move above part 2
Prepare to Pick(2)          ** Check vacuum made for VALVE #2 is OFF
; Move to part
Turn ON Vacuum(2) ;         ** Turn on VALVE 2A and wait for VacMade
2 to be ON
Part Present(8) ;           ** Check part present #8 is ON
; Move to dropoff
Turn OFF Vacuum(1,2)        ** Turn ON VALVEs 1B and 2B.  Turn OFF
VALVEs 1A and 2A
; Move above part
Turn OFF Blowoff(1,2) ;     ** Turn OFF VALVEs 1B and 2B and wait
for VacMade 1,2
                            ** to be OFF
Check No Part(2,4,8) ;      ** Check part presents #2, #4, and #8
to be off
```

13.4.2 Grip Part

Refer to Table 13–8 for details on the Grip Part macro.

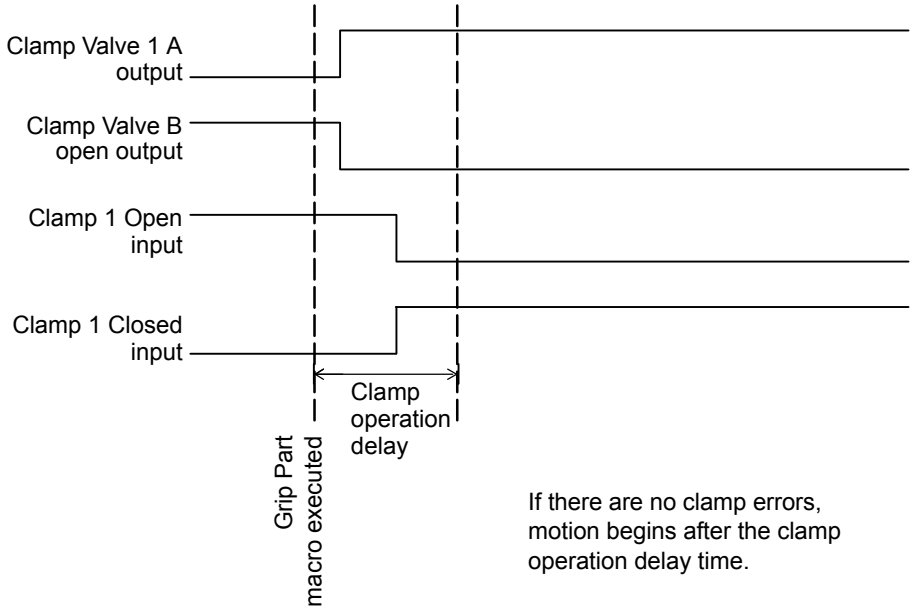
Table 13–8. Grip Part Macro

ITEM	DESCRIPTION
Program Name	**GRIPPART.TP**
Macro Name	**Grip Part**
KAREL Program	**MHGRIPPT.KL**
Description	This macro picks up parts using a CLAMP type valve. For each valve specified, the B side outputs are turned OFF and the A side outputs are turned ON. The macro program then waits for the clamp valve's "Operation Delay" period. If the specified valves use different delays, the macro program will use the longest delay. If "Check Closed" is required, an alarm will alert you if any of the clamp valves failed to close.

Table 13–8. Grip Part Macro (Cont'd)

ITEM	DESCRIPTION
Parameters	Up to nine parameters can be used. Each parameter is an INTEGER with a value from 1 to 9, which defines the CLAMP type valve to use. If no parameter is given, the macro program uses the current valve.
Errors	All errors will cause the program to display an error and pause: MHND-250 ABORT "%s is not an integer" MHND-251 ABORT "%s is not in range of 1 to 9" MHND-252 ABORT "Valve %s is not setup" MHND-246 ABORT "Valve %s is not a clamp gripper" MHND-256 PAUSE "Clamp %s A state error" MHND-257 PAUSE "Clamp %s B state error"

Table 13–8. Grip Part Macro (Cont'd)

ITEM	DESCRIPTION
Program Logic	• Check parameters • Close requested CLAMP valves • Turn on A side valve and turn OFF B side valve • Wait CLAMP "Operation Delay" period • If CLAMP "Check Closed" is YES • Check that each of the valve's Clamp open inputs are OFF and Clamp closed inputs are ON • Send an alarm if an error occurs
Timing Diagram	

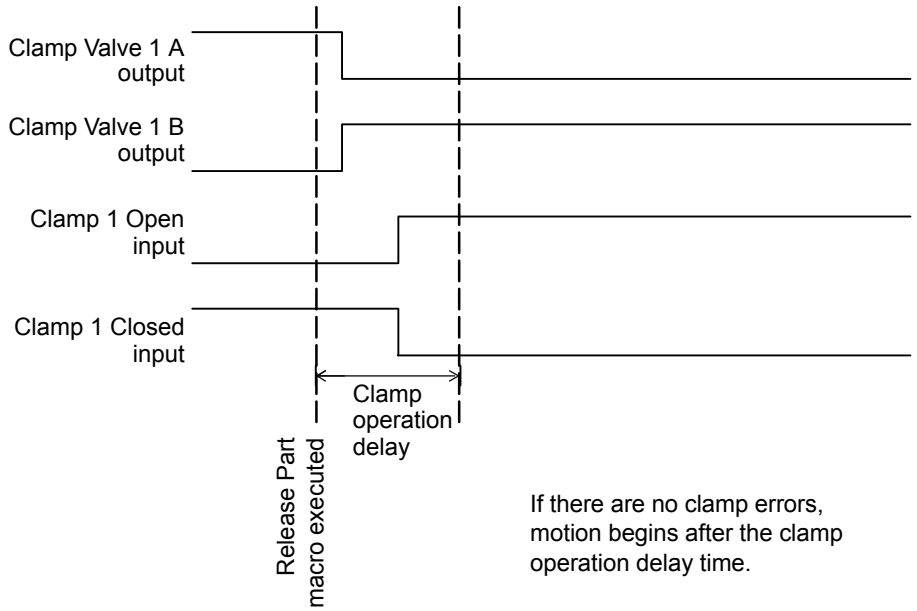
13.4.3 Release Part

Refer to Table 13–9 for details on the Release Part macro.

Table 13–9. Release Part Macro

ITEM	DESCRIPTION
Program Name	**DROPPART.TP**
Macro Name	**Release Part**
KAREL Program	**MHDROPPT.KL**
Description	This macro drops off parts using a CLAMP type valve. For each valve specified, the A side outputs are turned OFF and the B side outputs are turned ON. The macro program then waits for the clamp valve's "Operation Delay" period. If the specified valves use different delays, the macro program will use the longest delay. If "Check Opened" is required, an alarm will alert you if any of the valves failed to open.
Parameters	Up to 9 parameters can be used. Each parameter is an INTEGER with a value from 1 to 9, which defines the CLAMP type valve to use. If no parameter is given, the macro program uses the current tool.
Errors	All errors will cause the program to display an error and pause: MHND-250 ABORT "%s is not an integer" MHND-251 ABORT "%s is not in range of 1 to 9" MHND-252 ABORT "Valve %s is not setup" MHND-246 ABORT "Valve %s is not a clamp gripper" MHND-256 PAUSE "Clamp %s A state error" MHND-257 PAUSE "Clamp %s B state error"

Table 13–9. Release Part Macro (Cont'd)

ITEM	DESCRIPTION
Program Logic	• Check parameters • If the continuous check of part presents is active, turn it off • Close the requested CLAMP valves • Turn on the B side valve and turn off the A side valve • Wait CLAMP "Operation Delay" period • If CLAMP "Check Opened" is YES • Check that each of the valve's Clamp open inputs is ON and Clamp closed inputs is OFF • Send an alarm if an error occurs
Timing Diagram	

13.4.4 Part Present

Refer to Table 13–10 for details on the Part Present macro.

Table 13–10. Part Present Macro

ITEM	DESCRIPTION
Program Name	**CHKPART.TP**
Macro Name	**Part Present**
KAREL Program	**MHPTPRES.KL**
Description	This macro checks that specified Part Present inputs are ON. If $MH_TRYOUT is FALSE, for each part present input specified, this macro checks that the input is ON. If $MH_TRYOUT is TRUE, for each part present input specified, this macro checks that the input is OFF.
Parameters	Up to ten parameters can be used. Each parameter is an INTEGER with a value from 1 to 10, which defines the Part Present input to check. If no parameter is given, an error occurs.
Errors	MHND-250 ABORT "%s is not an integer" MHND-251 ABORT "%s is not in range of 1 to 9" MHND-253 PAUSE "No parameters passed to routine" MHND-254 PAUSE "Part present %s not reading part" MHND-248 PAUSE "TRYOUT MODE: %s reading part"
Program Logic	• Check parameters • For each specified Part Present input: • If the input if OFF, $MH_TRYOUT is FALSE, and the alarm is not disabled, send the alarm "signal not reading part" • If the input is ON, $MH_TRYOUT is TRUE, and the alarm is not disabled, send the alarm "TRYOUT MODE: signal reading part" • If Part Present "Continuous Check" is required, turn on a condition handler to watch that Part Present inputs stay on.

13.4.5 Check No Part

Refer to Table 13–11 for details on the Check No Part macro.

Table 13–11. Check No Part Macro

ITEM	DESCRIPTION
Program Name	**CHKNOPRT.TP**
Macro Name	**Check No Part**
KAREL Program	**MHNOPART.KL**
Description	This macro checks that the specified Part Present inputs are OFF . $MH_TRYOUT is not used by this macro. This macro handles any errors encountered if they have not been disabled by the user.
Parameters	Up to ten parameters can be used. Each parameter is an INTEGER with a value from 1 to 10, which defines the Part Present input to check. If no parameter is given, an error occurs.
Errors	MHND-250 ABORT "%s is not an integer" MHND-251 ABORT "%s is not in range of 1 to 9" MHND-252 ABORT "Valve %s is not setup" MHND-253 PAUSE "No parameters passed to routine" MHND-255 PAUSE "Part present %s still reads part"
Program Logic	• Check parameters • For each specified Part Present input: — If the input is ON and the alarm is not disabled, send the alarm "signal still reading part"

13.4.6 Prepare to Pick

Refer to Table 13–12 for details on the Prepare to Pick macro.

Table 13–12. Prepare to Pick Macro

ITEM	DESCRIPTION
Program Name	**OKTOPICK.TP**
Macro Name	**Prepare to Pick**
KAREL Program	**MHOKPICK.KL**
Description	This macro checks the required conditions before picking up a part for both CLAMP and VACUUM type valves. For each valve specified, if "Check Opened" is required, an alarm will alert you if any of the clamps are not open. For each valve specified, an alarm will alert you if any of the valve's Part Present inputs are on. Then, for ease of customization, the teach pendant program CLR2PRCD.TP is called.

Table 13–12. Prepare to Pick Macro (Cont'd)

ITEM	DESCRIPTION
Parameters	Up to nine parameters can be used. Each parameter is an INTEGER with a value from 1 to 9, which defines the tooling valves to check. If no parameter is given, the current valve is used.
Errors	MHND-250 ABORT "%s is not an integer" MHND-251 ABORT "%s is not in range of 1 to 9" MHND-252 ABORT "Valve %s is not setup" MHND-246 ABORT "Valve %s is not a clamp gripper" MHND-256 PAUSE "Clamp %s A state error" MHND-257 PAUSE "Clamp %s B state error" MHND-259 PAUSE "Vacuum %s feedback is not off" MHND-255 PAUSE "Part present %s still reads part"
Program Logic	• Check parameters • For each specified valve — If CLAMP "Check Opened" is YES, check that each of the valve's Clamp open inputs are ON and Clamp closed inputs are OFF — If VACUUM "Vac Feedback" is YES, check that each of the valve's Vacuum Made inputs is OFF — Check that the valve's Part Present inputs are OFF • Call CLR2PRCD.TP

13.4.7 Clear to Proceed

Refer to Table 13–13 for details on the Clear to Proceed macro.

Table 13–13. Clear to Proceed Macro

ITEM	DESCRIPTION
Program Name	**CLR2PROCD.TP**
Macro Name	**Clear to Proceed**
KAREL Program	**N/A**
Description	This macro is a teach pendant program you customize to define any other conditions that need to be checked before picking up the part. This macro is called from the Prepare to Pick macro (OKTOPICK.TP). This macro is used by both types of tooling valves.

Table 13–13. Clear to Proceed Macro (Cont'd)

ITEM	DESCRIPTION
Parameters	None
Errors	N/A (You define the UALMs.)
Program Logic	TPE defined by user /PROG CLR2PRCD Macro 1: !This macro is called by oktopick ; 2: !to allow the user to add any ; 3: !additional criteria for the ; 4: !robot to wait on before picking ; 5: !up the part. ; 6: ; 7: !Does not use the parameters ; 8: !passed to oktopick. ;

13.4.8 Turn ON Vacuum

Refer to Table 13–14 for details on the Turn ON Vacuum macro.

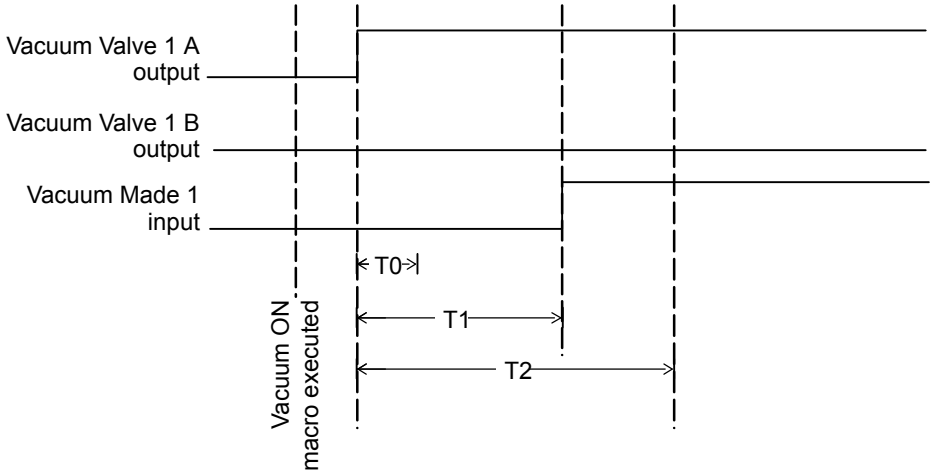
Table 13–14. Turn ON Vacuum Macro

ITEM	DESCRIPTION
Program Name	**VACUUMON.TP**
Macro Name	**Turn ON Vacuum**
KAREL Program	**MHVACON.KL**
Description	This macro picks up parts using a VACUUM type valve. For each valve specified, the B side outputs are turned OFF and the A side outputs are turned ON. If "Vac Feedback" is YES for any selected valve, the logic waits for the valve's "Ignore Feedback" period. If the specified valves use different delays, the macro program will use the longest delay. Then the macro program will wait for up to the "Feedback Timeout" for the vacuum made inputs to turn ON. As soon as the inputs are ON, the macro program continues. If the inputs never turn ON, an alarm is given. If "Vac Feedback" is NO for all selected valves, the macro program waits for the valve's "Feedback Timeout" period. If "Continuous Check" is required, an alarm will alert you if the vacuum made input turns OFF .

Table 13–14. Turn ON Vacuum Macro (Cont'd)

ITEM	DESCRIPTION
Parameters	Up to two parameters can be defined. Each parameter is an INTEGER with a value from 1 to 2, which defines the VACUUM type valve to use. If no parameter is given, the macro program uses the current valve.
Errors	MHND-250 ABORT "%s is not an integer" MHND-247 ABORT "Parm %s is not between 1-2" MHND-252 ABORT "Valve %s is not setup" MHND-245 ABORT "Valve %s is not a vacuum gripper" MHND-258 PAUSE "Vacuum %s feedback is not on"

Table 13–14. Turn ON Vacuum Macro (Cont'd)

ITEM	DESCRIPTION
Program Logic	• Check parameters • Close the required VACUUM valves — Turn on the A side valve and turn off the B side valve • If "Vac Feedback" is YES — Wait for VACUUM Ignore Feedback — For up to ["Feedback Timeout" - "Ignore Feedback"] ms, watch for the Vacuum Made input to turn ON — If the Vacuum Made input does not turn ON, give an error • If "Vac Feedback" is NO — Wait for VACUUM Feedback Timeout • If Vac Feedback "Continuous Check" is required and $MH_TRYOUT is FALSE, turn on a condition handler to watch that the Vacuum Made inputs stay ON
Timing Diagram	 T0 □ Time to ignore the Vacuum Made signal T1 □ Time to achieve vacuum feedback (Maximum Feedback Timeout) T2 □ Time to achieve vacuum with no feedback (Feedback Timeout) Motion begins after T1 or T2, depending on the □Vac Feedback□ setting

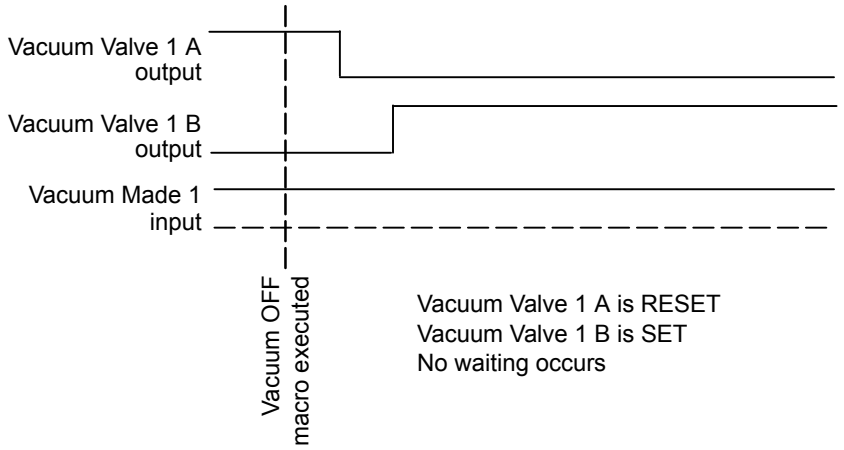
13.4.9 Turn OFF Vacuum

Refer to Table 13–15 for details on the Turn OFF Vacuum macro.

Table 13–15. Turn OFF Vacuum Macro

ITEM	DESCRIPTION
Program Name	**VACUUMOF.TP**
Macro Name	**Turn OFF Vacuum**
KAREL Program	**MHVACOFF.KL**
Description	This macro puts down parts using a VACUUM type valve. If "Continuous Check" is active, the macro disables the condition that is watching for the Vacuum Made input to turn OFF. For each valve specified, the B side outputs are turned ON and the A side outputs are turned OFF.
Parameters	Up to two parameters can be defined. Each parameter is an INTEGER with a value from 1 to 2, which defines the VACUUM type valve to use. If no parameter is given, the macro program uses the current valve.
Errors	MHND-250 ABORT "%s is not an integer." MHND-247 ABORT "%s is not in the range of 1 to 2." MHND-252 ABORT "Valve %s is not setup." MHND-245 ABORT "Valve %s is not a vacuum gripper" MHND-259PAUSE "Vacuum %s feedback is not off"

Table 13–15. Turn OFF Vacuum Macro (Cont'd)

ITEM	DESCRIPTION
Program Logic	• Check parameters • If the continuous checking of part presents is active, turn it off • If the continuous checking of vacuum made is active, turn it off • Open the requested vacuum valves — Turn on the B side valve and turn off the A side valve
Timing Diagram	

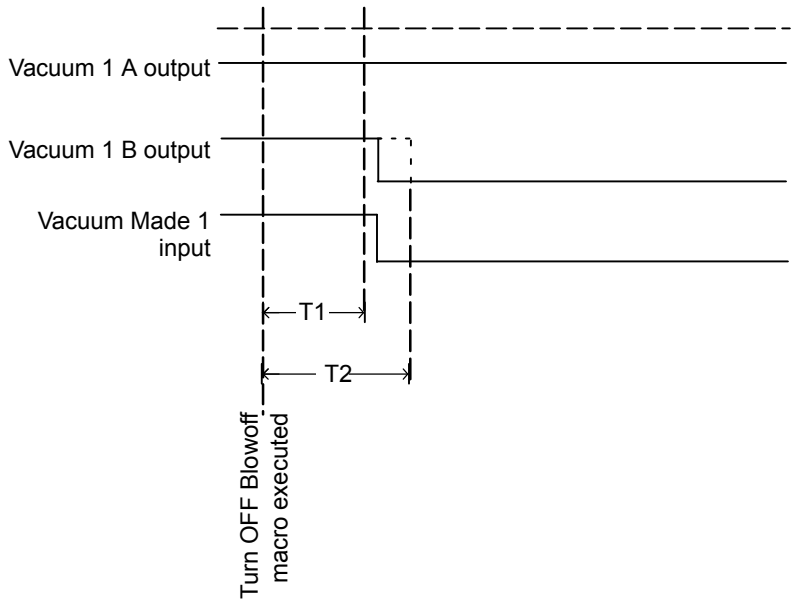
13.4.10 Turn OFF Blowoff

Refer to Table 13–16 for details on the Turn OFF Blowoff macro.

Table 13–16. Turn OFF Blowoff Macro

ITEM	DESCRIPTION
Program Name	**BLOWOFF.TP**
Macro Name	**Turn OFF Blowoff**
KAREL Program	**MHBLOWOF.KL**
Description	This macro turns off the blowoff output when using a VACUUM type valve. If "Vac Feedback" is YES for any selected valve, the macro program will wait for up to the valve's "Blow off time" period, for the Vacuum Made inputs to go OFF. When the inputs are OFF, the macro program continues. If the inputs never go OFF, an alarm is issued. If "Vac Feedback" is NO for all selected valves, the macro program waits for the valve's "Blow off time" period.
Parameters	Up to two parameters can be defined. Each parameter is an INTEGER with a value from 1 to 2, which defines the VACUUM type valve to use. If no parameter is given, the macro program uses the current valve.
Errors	MHND-250 ABORT "%s is not an integer." MHND-247 ABORT "Parm %s is not between 1-2" MHND-252 ABORT "Valve %s is not setup." MHND-245 ABORT "Valve %s is not a vacuum gripper"

Table 13–16. Turn OFF Blowoff Macro (Cont'd)

ITEM	DESCRIPTION
Program Logic	• Check parameters • If "Vac Feedback" is YES — For up to "Blow off time" ms, watch for the Vacuum Made input to turn OFF — If the Vacuum Made input does not turn OFF, issue an error • If "Vac Feedback" is NO — Wait for VACUUM "Blow off time" ms • Turn OFF Blow off output — Turn off B side valve
Timing Diagram	 T1 □ Time for vacuum made to be lost (maximum T2) T2 □ Time for vacuum to be released without feedback (Blowoff Tim Motion begins after T1 or T2, depending on □Vac Feedback□ setting

13.4.11 Set CurrentValve

Refer to Table 13–17 for details on the Set CurrentValve macro.

Table 13–17. Set CurrentValve Macro

ITEM	DESCRIPTION
Program Name	**SETVALVE.TP**
Macro Name	**Set CurrentValve**
KAREL Program	**MHSETVLV.KL**
Description	This macro sets the "Current Valve," which is used as a default by most the Material Handling macros.
Parameters	One macro can be used. The parameter is an INTEGER with a value from 1 to 9, which defines the valve to use as a default.
Errors	MHND-250 ABORT "%s is not an integer." MHND-251 ABORT "%s is not in the range of 1 to 9." MHND-252 ABORT "Valve %s is not setup." MHND-253 ABORT "No parameters passed to routine"

13.4.12 Set MH Tool

Refer to Table 13–18 for details on the Set MH Tool macro.

Table 13–18. Set MH Tool

ITEM	DESCRIPTION
Program Name	**SET_TOOL.TP**
Macro Name	**Set MH Tool**
KAREL Program	**MHSETGRP.KL**
Description	This macro sets the "Current Tool", which is used as a default by most the Material Handling macros.
Parameters	The parameter is an INTEGER with a value from 1 to 29, which defines the tool to use as a default.
Errors	MHND-250 ABORT "%s is not an integer." MHND-251 ABORT "%s is not in the range of 1 to 9." MHND-252 ABORT "Valve %s is not setup." MHND-253 ABORT "No parameters passed to routine"

13.5 PRODUCTION

When you have the Material Handling Plug-in, production operation has the following features, if they have been enabled:

- Tryout mode
- Disable for 20 cycles
- Alarms

13.5.1 MH Tryout Mode

MH Tryout Mode enables and disables the ability of the robot to execute material handling commands in a tryout, or dry run, mode. You enable and disable MH Tryout Mode on the TEST CYCLE Spot Weld menu after you set up the cell interface I/O. Refer to Section 4.10.2 .

You can also enable MH Tryout Mode by setting the system variable $SHELL_WRK.$tryout_mode to TRUE.

When MH Tryout Mode is enabled, it does the following:

- Disables checking for feedback signals.
- Changes the checking for part present signals. When MH Tryout Mode is enabled on the TEST CYCLE Spot Weld menu, the part present signals must never be on.
- Allows the robot to run the program with all other material handling alarms and conditions enabled.

13.5.2 Disable for 20 Cycles

When you use the Disable Fault Checking feature, you can specify that the robot ignore tooling valve inputs, clamps, and part present alarms for 20 cycles. After 20 cycles, an error message will be re-enabled and, if the alarm occurs again, will pause the program until it is reset by the operator.

You can use the Disable Fault Checking feature to disable each tooling valve PAUSE alarm for 20 cycles. When a material handling pause alarm occurs, you will be able to choose Disable Alarm on the error recovery screen to disable this particular alarm for 20 cycles. Refer to Section 7.2.5 for more information about using the Disable Fault Checking feature to do this.

13.5.3 Alarms

All Material Handling alarms turn on the MH Fault output, if the controller is not in MH Tryout Mode. The output will be turned off when a FAULT RESET is given.

The MH Alert output turns ON when a fault that has been disabled for 20 cycles occurs. The output will be turned OFF when a FAULT RESET is given.

Chapter 14

MACHINE TOOL OPTIONS

Contents

Chapter 14 MACHINE TOOL OPTIONS 14–1
14.1 OVERVIEW 14–2
14.1.1 Overview 14–2
14.2 MACHINE TOOL EASY SETUP 14–2
14.2.1 Overview 14–2
14.2.2 User Interface 14–5
14.2.3 Machine Tool Programs 14–10
14.2.4 Machine Tool Data 14–18
14.2.5 Customizing Machine Tool Easy Setup 14–25

14.1 OVERVIEW

14.1.1 Overview

The Machine Tool options are a family of products for use in a system in which a Machine Tool equipped with CNC and Robot are connected. Refer to Table 14–1 for information about the functions each option performs.

Table 14–1. Relation of Machine Tool Options

Machine Tool Software Option	Documentation	Function
R727 Machine Tool Easy Setup	This manual	Graphical interface to set up the robot for machine tool automation.
J984 Machine Tool Connection	Robot and Machine Tool Integration Function Operator's Manual, MAROCMTOL08091E	Closely coupled integration of robot with specific FANUC CNC models. Please refer to manual for details. • Function to connect CNC with robot. • Function to display status and operate the robot at CNC. • Function to display status and operate the CNC at robot.
J980 Initial Setup Function	Initial Setup Function Operator's Manual, MAROCINSE06091E	Software to enter settings for robot initial setup. • Settings are entered on PC using SFTROBINI750CD ROBOT INITIAL SETUP • Settings are loaded to the robot. • Machine Tool Easy Setup includes wizard to load settings.

14.2 MACHINE TOOL EASY SETUP

14.2.1 Overview

Machine Tool Easy Setup runs on the iPendant and provides graphics-based programming. Wizards

will guide you through simple step-by-step procedures to create complete, custom robot programs for the desired machine tending process. An optional touch screen interface is provided. In lieu of a touch screen, cursor keys and enter can select the buttons. Function keys are used for navigating the wizards. Machine Tool Easy Setup is not available on the Monochrome Teach Pendant.

Machine Tool Easy Setup provides the new features listed in the following tables.

Table 14–2. Process Navigator

Feature	Description
Process Navigator	Guides you through each step in the process. • Define the Cell is used to setup the robot and machine tool cell. • Teach Cell is used to teach home, maintenance, and PNS programs. • Run Production is used to monitor production.
Color Buttons	The background color of a button can be used to distinguish the links in the Process Navigator: • Green - Wizard button • Yellow - Navigator button • Cyan - Shortcut to other menu • Lavender - Shortcut to TP Editor with specified program
Wizard Function Keys	When a wizard is in progress, the function keys are defined as follows: • F2 BACK will go back one step. • F3 NEXT will continue to the next step. • F3 FINISH will finish the wizard and return to the Process Navigator. • F5 CANCEL will cancel the wizard and return to the Process Navigator. Sometimes CANCEL is on F10 so you may need to press NEXT to get the next page of function keys. • F4 DIRECT ENTRY is used to directly enter the tool frame or user frame values. • F5 SKIP is used to skip the tool frame or user frame and directly go to the program step.
Wizard is Remembered	When a wizard is in progress, the current step is saved when the Machine Tool menu is exited. It will resume the wizard when the Machine Tool menu is reentered. If this is not what you want, please press CANCEL to return to the Process Navigator.

Table 14–2. Process Navigator (Cont'd)

Feature	Description
Machine Tool Shortcuts	When MENUS key is pressed, the Machine Tool menu can be selected by pressing F1 in the Menu Favorites dialog. While in the TP Editor, the function key labelled MTOOL will shortcut to the Machine Tool menu.
Password Protection	If passwords are setup, then SETUP level is required to change any of the settings in the Machine Tool menu. All wizards still operate, but will not let you change any data. The shortcuts are still available. While in the TP Editor, then PROGRAM level is required.

Table 14–3. Define the Cell

Function	Purpose
Setup Machine List	Wizard to setup the number of machine tools and configure I/O. Up to 4 machine tools can be connected to the robot. When Machine Tool Connection option is loaded, the wizard will contain additional steps, such as selecting the I/O device, loading the standard configuration, and setting data communications.
Setup Gripper	Wizard to setup single or dual gripper, create the tool frame(s) and edit the gripper open and close macros.
Setup Part Infeed Station	Wizard which selects the station from a predefined list, such as fixture, fixed grid, flexible location using iRVision, or drawers. The wizard will create the user frame(s) and edit the pick program.
Add a Pre Process Operation	Wizard to add a pre process operation and edit the pre process program.
Setup Machine Station	Wizard to create the user frame and edit the load/unload program for each machine tool in the cell.
Add a Post Process Operation	Wizard to add a post process operation and edit the post process program.
Setup Finish Part Station	Wizard which selects the station from a predefined list, such as fixture or fixed grid. The wizard will create the user frame(s) and edit the place program.
Setup Machine Diagnostics	Shortcut to SETUP Visual Diagnostics if R702 Visual Diagnostics option is loaded. Refer to the Visual Diagnostics section in the *HandlingTool Setup and Operations Manual* for more information.

Table 14–4. Teach Cell

Functions	Purpose
Teach Home Position	Shortcut to TP Editor with the home position macro.
Teach Maintenance Position	Shortcut to TP Editor with the maintenance program.
Program Number Selection	Wizard to create and edit PNS programs which can be started by CNC.
Teach Reference Position	Shortcut to the Reference Position menu. When using Machine Tool Connection option, the robot program can be started by CNC only when the robot is at reference position 1.

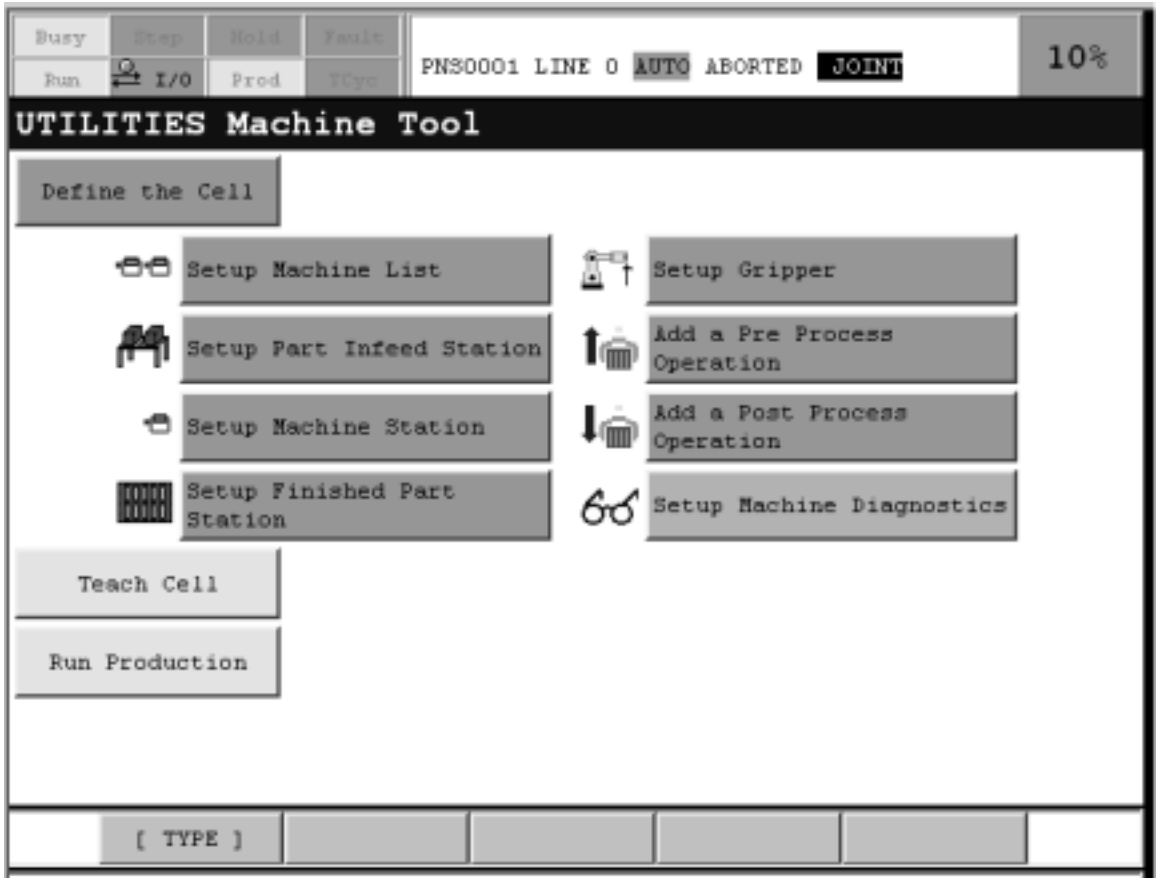
Table 14–5. Run Production

Function	Purpose
Production Status	Shortcut to a Production Status screen. This screen can be changed using the BROWSER Panel Wizard menu. Start the Panel Wizard and select Panel 1.
CNC Status	Shortcut to CNC Status screen if Machine Tool Connection option is loaded. There are several screens which can be displayed: CNC Actual Position, CNC Alarm, CNC Message, CNC Running Program, and I/O data connected with CNC. If Machine Tool Connection option is not loaded, this will link to the Browser FAVORITES menu. If the CNC has a web server, a link can be made to display its information.
Manual CNC Operation	Shortcut to Machine Tool Operation screen. This is only shown if Machine Tool Connection option is loaded. This screen enables you to operate jog feed of the connected machine tool.
View Machine Diagnostics	Shortcut to STATUS Visual Diagnostics if R702 Visual Diagnostics option is loaded. Refer to the Visual Diagnostics section in the *HandlingTool Setup and Operations Manual* for more information.

14.2.2 User Interface

Machine Tool Easy Setup is started by pressing the MENUS key on the iPendant and selecting Machine Tool from UTILITIES or pressing the F1 key in the Menu Favorites dialog. The Process Navigator will open “Define the Cell” as shown in Figure 14–1 .

Figure 14–1. Define The Cell



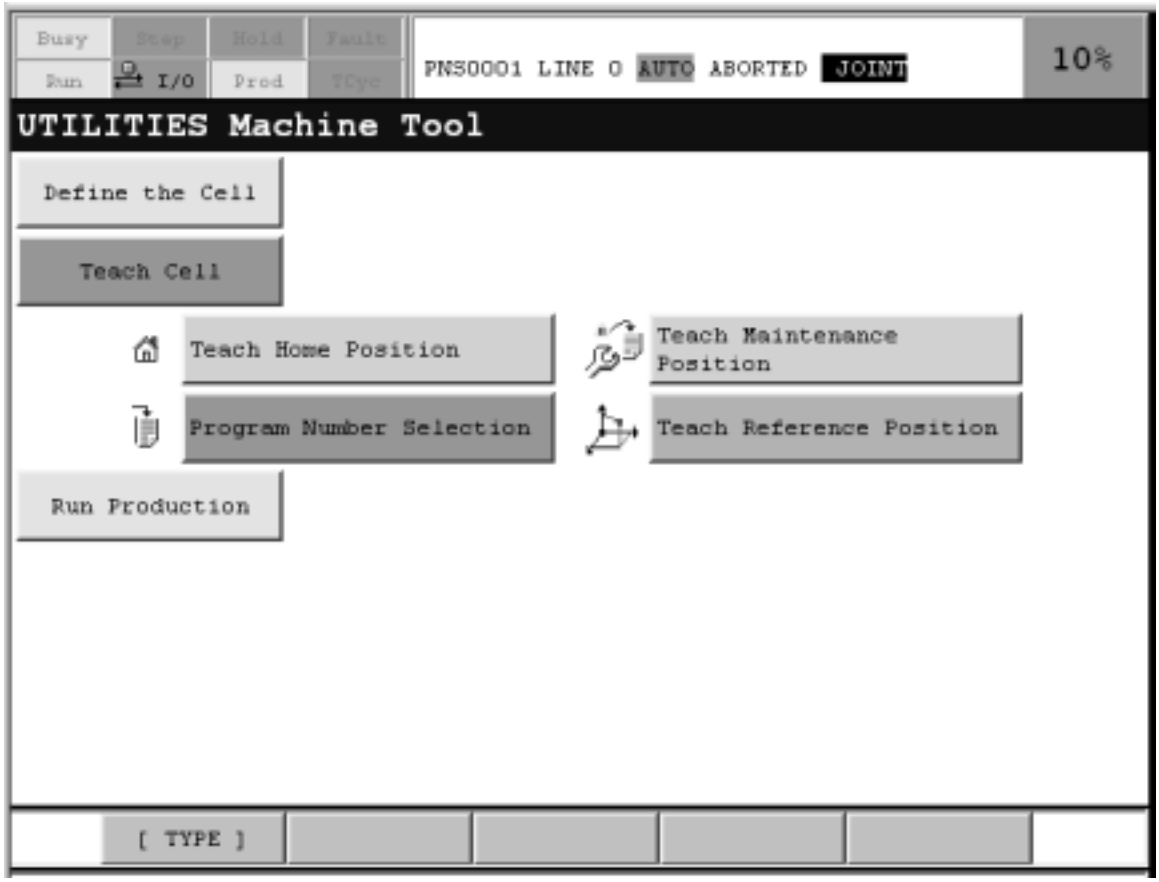
Each green button will open a wizard to guide you in that setup, such as the one shown in Figure 14–2 .

Figure 14–2. Setup Machine List



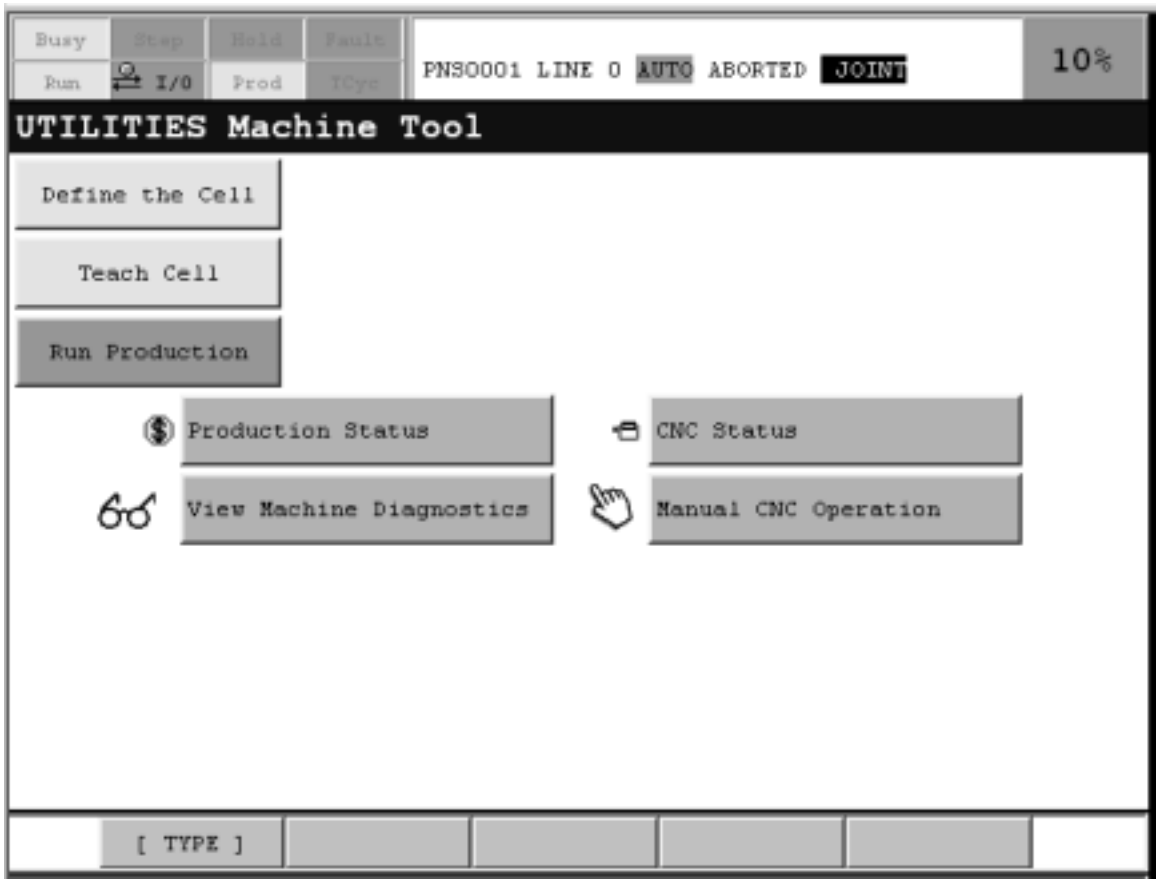
Once the cell is defined, the programs which are started from the CNC can be taught by opening "Teach Cell" as shown in Figure 14–3 .

Figure 14–3. Teach Cell



When production is running the status can be viewed by opening “Run Production” as shown in Figure 14–4 . This page is shown when using FANUC CNC with Machine Tool Connection option. Some of the links may not be available for your CNC connection.

Figure 14–4. Run Production



14.2.3 Machine Tool Programs

Figure 14–5. CNC Tending Flowchart

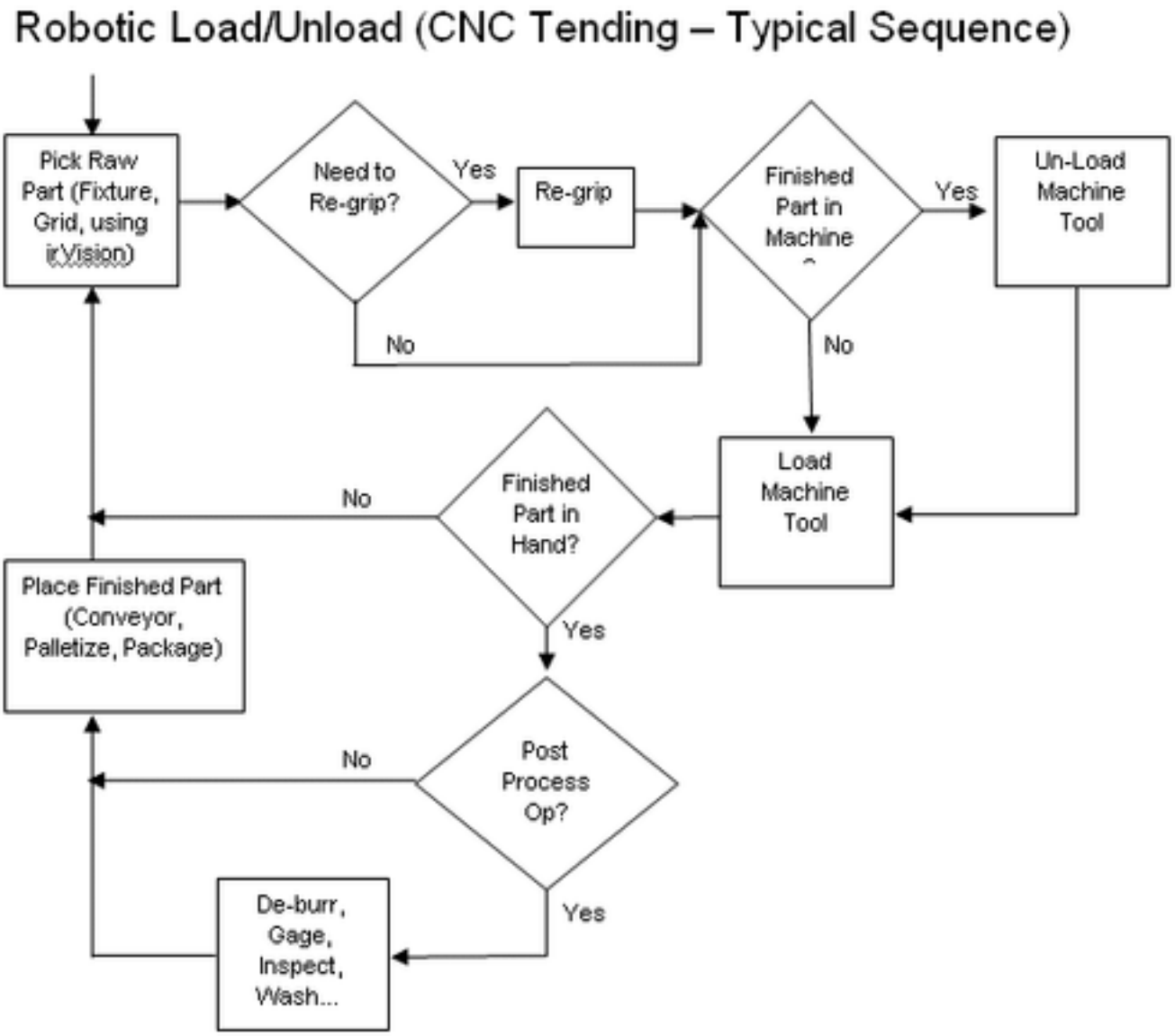


Table 14–6. Teach Pendant Programs

Teach Pendant Program Name	Gripper Type	Description
GRIPOPEN.MR Open Gripper	Single	This macro opens the gripper.
GRIPCLOS.MR Close Gripper	Single	This macro closes the gripper.
RAWOPEN.MR Open Gripper	Dual	This macro opens the raw part gripper.
RAWCLOSE.MR Close Gripper	Dual	This macro closes the raw part gripper.
FINOPEN.MR Open Fin Grip	Dual	This macro opens the finished part gripper.
FINOPEN.MR Close Fin Grip	Dual	This macro closes the finished part gripper.
MTSHOME.MR Move to Home		Machine Tool Connection option only. This macro sends the robot to the home position.
MT_HOME.MR Move to Home		When Machine Tool Connection option is not loaded. This macro sends the robot to the home position.
MT_MAINT.MR		This macro sends the robot to the maintenance position.
MT_MAIN.TP		When Machine Tool Connection option is not loaded. This program initializes the machine tending process. If CNC will not signal a PNS program to run, then this program can be changed to handle the tending process. Please refer to the Program Number Selection templates for an example.
MT_PIK01.TP		This program picks the raw part from a fixed location/fixture.
MT_PIK02.TP		This program picks the raw part from a flexible location using iRVision 2D with fixed overhead camera.
MT_PIK03.TP		This program picks the raw part from a fixed 2D grid (e.g. tray, conveyor, single drawer etc.).
MT_PIK04.TP		This program picks the raw part from a flexible grid using iRVision 2D with fixed overhead camera (e.g. parts from a tray, table, conveyor, drawer etc.).

Table 14–6. Teach Pendant Programs (Cont'd)

Teach Pendant Program Name	Gripper Type	Description
MT_PIK05.TP		This program picks the raw part from a fixed 2D grid with multiple drawers.
MT_PRP01.TP		This program pre processes the raw part. It contains commands for reorient and regrip. It can be modified to handle any pre processing necessary.
MT_LUL11.TP	Single	This program loads/unloads machine 1.
MT_LUL12.TP	Single	This program loads/unloads machine 2.
MT_LUL13.TP	Single	This program loads/unloads machine 3.
MT_LUL14.TP	Single	This program loads/unloads machine 4.
MT_LUL21.TP	Dual	This program loads/unloads machine 1.
MT_LUL22.TP	Dual	This program loads/unloads machine 2.
MT_LUL23.TP	Dual	This program loads/unloads machine 3.
MT_LUL24.TP	Dual	This program loads/unloads machine 4.
MT_POP01.TP		This program post processes the finished part. This is an empty program. It can be modified to handle any post processing necessary.
MT_NOPR.TP		This program is used when pre or post processing is not required.
MT_PUT11.TP	Single	This program places the finished part to a single fixture.
MT_PUT12.TP	Single	This program places the finished part to a grid (e.g. table, tray, conveyor or drawer).
MT_PUT21.TP	Dual	This program places the finished part to a single fixture.
MT_PUT22.TP	Dual	This program places the finished part to a grid (e.g. table, tray, conveyor or drawer).
MT_PNS11n.LS MT_PNS12n.LS MT_PNS13n.LS MT_PNS14n.LS	Single	Program Number Selection template for normal CNC start with CNC 1-4. This template will be ascii uploaded to PNSnnnn.TP based on the number you select in the Program Number Selection wizard.

Table 14–6. Teach Pendant Programs (Cont'd)

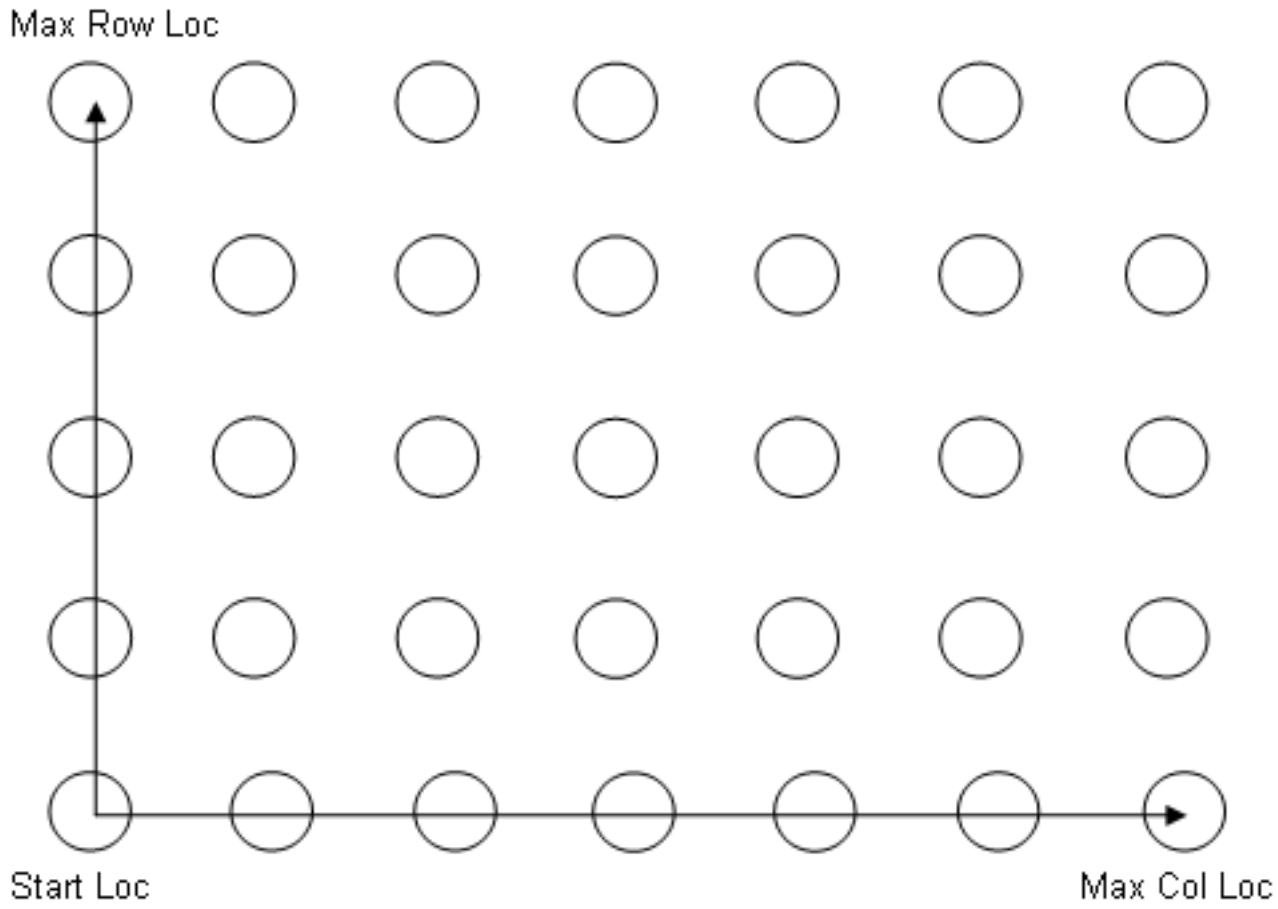
Teach Pendant Program Name	Gripper Type	Description
MT_PNS11o.LS MT_PNS12o.LS MT_PNS13o.LS MT_PNS14o.LS	Single	Program Number Selection template for overlap CNC start with CNC 1-4. This template will be ascii uploaded to PNSnnnn.TP based on the number you select in the Program Number Selection wizard.
MT_PNS21n.LS MT_PNS22n.LS MT_PNS23n.LS MT_PNS24n.LS	Dual	Program Number Selection template for normal CNC start with CNC 1-4. This template will be ascii uploaded to PNSnnnn.TP based on the number you select in the Program Number Selection wizard.
MT_PNS21o.LS MT_PNS22o.LS MT_PNS23o.LS MT_PNS24o.LS	Dual	Program Number Selection template for overlap CNC start with CNC 1-4. This template will be ascii uploaded to PNSnnnn.TP based on the number you select in the Program Number Selection wizard.

Pick Programs

This section provides an overview of the pick programs.

- MT_PIK01
  - — Recommended for large parts on a fixture.
- MT_PIK02
  - — Recommended for large parts (e.g. Auto Wheels) within FOV of a fixed camera.
- MT_PIK03

Figure 14–6. Grid Example



— Recommended for picking parts from a fixed 2D grid.

— Teach UFRAME[11] with actual part and gripper.

  – Make origin as Start Location (row 1, col 1)

  – X-direction at Max Col Location (row 1, max col)

  – Y-direction at Max Row Location (max row, col 1)

  All 3 positions must have identical gripper orientation.

— First pick up position will have null position.

— Setup Procedure for 5 rows x 7 Columns grid:

  – Calculate distance Dx between Start Loc to Max Col Loc

  – Calculate distance Dy between Start Loc to Max Row Loc

  – Manually set:

    – R[18:Pick max row] = 5

- R[19:Pick max col] = 7
- R[16:Pick row offset] = Dy/(Max row - 1)
- R[17: Pick col offset] = Dx/(Max col - 1)

— Test run part pickup at all part pickup positions.

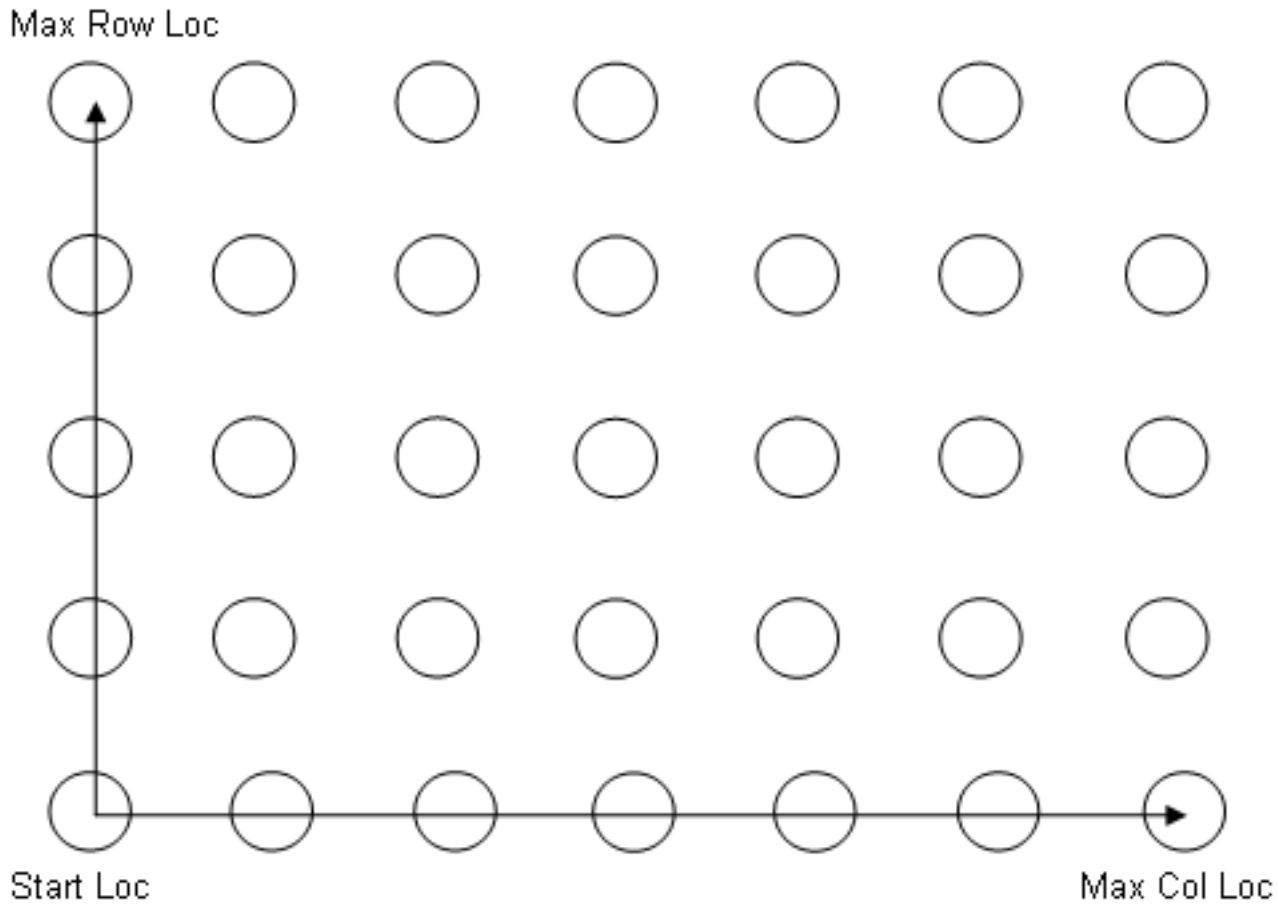
- MT_PIK04
  - — Fixed Camera should have FOV to cover all parts in the grid. Correct size lens and correct camera height is essential for successful operation.
  - — Recommended that parts are sorted row-wise or column-wise. Lighting, lens filter should be used to find all parts in the grid.
  - — Parts not found will be skipped by robot.
  - — Part pickup will be random within a sorted row/col.
- MT_PIK05
  - — Recommended for applications that require robot to pick parts from multiple drawers. Up to a maximum of 10 drawers are supported. Robot is programmed to open the drawer, pick up the parts one by one and then close the door before proceeding to the next drawer. The drawers are set up to be serviced from top to bottom.
  - — Setup requires each drawer to be setup as an independent reference frames as shown in MT_PIK03.
  - — Part pickup order is row by row.

Place Programs

This section provides an overview of the place programs.

- MT_PUT11 and MT_PUT21
  - — Recommended for placing large parts on a fixture.
  - — MT_PUT11 is for single gripper.
  - — MT_PUT21 is for dual gripper.
- MT_PUT12 and MT_PUT22

Figure 14–7. Grid Example



— Recommended for placing parts to a fixed 2D grid.

— MT_PUT12 is for single gripper.

— MT_PUT22 is for dual gripper.

— Teach UFRAME[21] with actual part and gripper.

  – Make origin as Start Location (row 1, col 1)

  – X-direction at Max Col Location (row 1, max col)

  – Y-direction at Max Row Location (max row, col 1)

  All 3 positions must have identical gripper orientation.

— First place position will have null position.

— Setup Procedure for 5 rows x 7 Columns grid:

  – Calculate distance Dx between Start Loc to Max Col Loc

  – Calculate distance Dy between Start Loc to Max Row Loc

- Manually set:
  - R[24:Place max row] = 5
  - R[25:Place max col] = 7
  - R[22:Place row offset] = Dy/(Max row - 1)
  - R[23: Place col offset] = Dx/(Max col - 1)

— Test run part placement at all part place positions.

Load/Unload Programs

This section provides an overview of the load/unload programs for CNC 1 - CNC 4.

- MT_LUL11 - MT_LUL14
  - — Designed for a single gripper.
  - — Two parameters are passed to these programs.
    - AR[1] = 1, AR[2] = 0, Load operation
    - AR[2] = 0, AR[2] = 1, Unload operation
    - Otherwise UARM[1] “Incorrect Service” is posted
- MT_LUL21 - MT_LUL24
  - — Designed for a dual gripper.
  - — Two parameters are passed to these programs.
    - AR[1] = 1, AR[2] = 0, Load operation
    - AR[2] = 0, AR[2] = 1, Unload operation
    - AR[2] = 1, AR[2] = 1, Exchange
    - Otherwise UARM[1] “Incorrect Service” is posted

PNS Programs

This section provides an overview of the PNS template programs for CNC 1 - CNC 4.

- MT_PNS11n - MT_PNS14n (Normal Start)
  - — Designed for a single gripper
  - — GI[1] defines the service type for CNC
    - GI[1] = 1, The robot picks up a work piece from the Infeed Station and loads a work piece to machine tool. In this case, the robot executes only the load of a work piece because a processed work piece is not in the machine tool.
    - GI[3] = 1, The robot unloads a processed work piece from the machine tool and places the work piece at the Finished Part Station.

– Otherwise UARM[1] “Incorrect Service” is posted

- MT_PNS11o - MT_PNS14o (Overlap Start)
  - — Same as MT_PNS1*n except provides an Overlap Start. In case that only unloading a work piece is executed, the robot moves to the position in front of the machine tool and waits until Service Call signal is turned on. When Service Call signal is turned on, the robot moves into the machine tool and the designated process is executed. After the process executed into the machine tool is ended, when the robot moves to the position out of the machine tool, the robot turns off Service Completed signal to release waiting in CNC program. Service completed signal is turned off when Service Call Signal is turned off.
- MT_PNS21n - MT_PNS24n (Normal Start)
  - — Designed for a dual gripper
  - — GI[1] defines the service type for CNC
    - – GI[1] = 1, The robot picks up a work piece from the pallet and loads a work piece to machine tool at cycle start. In this case, the robot executes only the load of a work piece because a processed work piece is not in the machine tool.
    - – GI[2] = 1, The robot moves into the machine tool with holding an unprocessed work piece, then unloads a processed work piece at first, and loads an unprocessed work piece.
    - – GI[3] = 1, The robot unloads a processed work piece from the machine tool at cycle end.
    - – Otherwise UARM[1] “Incorrect Service” is posted
- MT_PNS21o - MT_PNS24o (Overlap Start)
  - — Same as MT_PNS2*n except provides an Overlap Start. In case that both loading and unloading a work piece are executed in the same cycle and only unloading a work piece is executed at ending cycle, the robot moves to the position in front of the machine tool and waits until Service Call signal is turned on. When Service Call signal is turned on, the robot moves into the machine tool and the designated process is executed. After the process executed into the machine tool is ended, when the robot moves to the position out of the machine tool, the robot turns off Service Completed signal to release waiting in CNC program. Service completed signal is turned off when Service Call Signal is turned off.

14.2.4 Machine Tool Data

The wizard will teach the following User Tools:

Table 14–7. User Tools

User Tool	User Tool #	User Tool Comment
Single Gripper	1	Gripper

Table 14–7. User Tools (Cont'd)

User Tool	User Tool #	User Tool Comment
Dual Gripper - Raw Part	1	Raw part gripper
Dual Gripper - Finish Part	2	Finished part gripper

The wizard will teach the following User Frames:

Table 14–8. User Frames

User Frame	User Frame #	User Frame Comment
Machine Station 1	1	Machine 1
Machine Station 2	2	Machine 2
Machine Station 3	3	Machine 3
Machine Station 4	2	Machine 4
Vision Calibration Frame	9	
Part Infeed Station	11	Part Infeed 1
Part Infeed Drawers 1 up to 10	11 - 20	Part Infeed 1 - Part Infeed 10
Finished Part Station	21	Finished 1
Finished Part Drawers 1 up to 10	21 - 30	Finished 1 - Finished 10

The wizard sets macros for use in the TP programs:

Table 14–9. Macros

Macro	Description
Open Gripper	For single or dual gripper. Used by pick, load/unload, and place programs. Can be executed in Tool 1 menu.
Close Gripper	For single or dual gripper. Used by pick, load/unload, and place programs. Can be executed in Tool 1 menu.
Open Fin Grip	For dual gripper. Used by place program. Can be executed in Tool 2 menu.
Close Fin Grip	For dual gripper. Used by load/unload program. Can be executed in Tool 2 menu.
Move to Home	Can be executed in Tool 1 menu.
Pick Program	Used by PNS programs to call pick program selected by the wizard.
Pre Proc Prog	Used by PNS programs to call pre process program selected by the wizard.

Table 14–9. Macros (Cont'd)

Macro	Description
Load/Unload CNC1	Used by PNS programs to call load/unload CNC1 program selected by the wizard.
Load/Unload CNC2	Used by PNS programs to call load/unload CNC2 program selected by the wizard.
Load/Unload CNC3	Used by PNS programs to call load/unload CNC3 program selected by the wizard.
Load/Unload CNC4	Used by PNS programs to call load/unload CNC4 program selected by the wizard.
Post Proc Prog	Used by PNS programs to call post process program selected by the wizard.
Place Program	Used by PNS programs to call place program selected by the wizard.

The template TP programs use registers as shown Table 14–10 . In addition, the wizard sets registers starting at 100 for use in the TP programs.

Table 14–10. Registers

Register #	Description
R[1:]	Not used
R[2:Pick Counter]	Incremented by pick program after picking up the part from the Part Infeed Station.
R[3:Dropped Parts]	Incremented by load/unload programs after unload operation if Robot Input for part presence gripper is OFF.
R[4:Reject Counter]	Not used
R[5:Finished Parts]	Incremented by place program after placing the part on the Finished Part Station.
R[6:Drawer number]	Used internally by pick programs that use drawers.
R[7:]	Not used
R[8:]	Not used
R[9:Re_pick]	Used by pre process program to reorient and repick the raw part. Please set this value to 1 if required.
R[10:Vis pick order]	Incremented by pick program is irVision is used.
R[11:Load]	Used internally by load/unload programs.
R[12:Unload]	Used internally by load/unload programs.
R[13:Index reg]	Used internally by pick and place programs that use a grid.

Table 14–10. Registers (Cont'd)

Register #	Description
R[14:Cur pick row]	Used internally by pick programs that use a grid.
R[14:Cur pick row]	Used internally by pick programs that use a grid.
R[15:Cur pick col]	Used internally by pick programs that use a grid.
R[16:Pick row offset]	Used internally by pick programs that use a grid.
R[17:Pick col offset]	Used internally by pick programs that use a grid.
R[18:Pick max row]	Used by pick programs that use a grid. Please set this value to the maximum number of rows.
R[19:Pick max col]	Used by pick programs that use a grid. Please set this value to the maximum number of columns.
R[20:Cur place row]	Used internally by place programs that use a grid.
R[21:Cur place col]	Used internally by place programs that use a grid.
R[22:Place row offset]	Used internally by place programs that use a grid.
R[23:Place col offset]	Used internally by place programs that use a grid.
R[24:Place max row]	Used by place programs that use a grid. Please set this value to the maximum number of rows.
R[25:Place max col]	Used by place programs that use a grid. Please set this value to the maximum number of columns.
R[100:Num Grippers]	1 - Single Gripper 2 - Dual Gripper
R[101:Infeed Type]	1 - Fixed location/fixture 2 - Flexible location using iRVision 2D 3 - Fixed grid (tray, single drawer, etc) 4 - Flexible grid using iRVision 2D 5 - Fixed grid with multiple drawers
R[102:Pre Process]	0 - Pre Process disabled 1 - Pre Process enabled
R[103:Post Process]	0 - Post Process disabled 1 - Post Process enabled
R[104:Machine No]	The current machine selected by the wizard.
R[105:Num Machines]	1 - 4 based on the number of machines.
R[106:Finish Type]	1 - Fixed location/fixture 2 - Fixed grid (tray, single drawer, etc)
R[107:Num Infeed]	1 - 10 based on the number of Part Infeed drawers.
R[108:Num Finish]	1 - 10 based on the number of Finished Part drawers.

The template TP programs will use the following position registers:

Table 14–11. Position Registers

Register #	Description
PR[5:Pick ref loc]	First pick position will have null position for Part Infeed Station.
PR[6:Place ref loc]	First place position will have null position for Finished Part Station.
PR[10:Pick ap offset]	Approach offset for pick position for Part Infeed Station.
PR[11:Pick pos]	Pick position for Part Infeed Station.
PR[12:Pick rt offset]	Retreat offset for pick position for Part Infeed Station.
PR[13:Re_pick pos]	Re_pick position for pre process program.
PR[14:Re_orient]	Re_orient position for pre process program.
PR[15:Place ap offset]	Approach offset for place position for Finished Part Station.
PR[16:Place pos]	Place position for Finished Part Station.
PR[17:Place rt offset]	Retreat offset for place position for Finished Part Station.
CNC 1	
PR[30:Inside C1 T1]	Position inside CNC 1 for loading the raw part. Also used for unloading the finished part if single gripper.
PR[31:Inside C1 T2]	Position inside CNC 1 for unloading the finished part if dual gripper.
PR[32:Outside C1 T1]	Position just outside CNC 1 for loading the raw part. Also used for unloading the finished part if single gripper.
PR[33:Outside C1 T2]	Position just outside CNC 1 for unloading the finished part if dual gripper.
PR[34:C1 place pos]	Place position inside CNC 1 for loading the raw part.
PR[35:C1 place ap ofst]	Approach offset for place position inside CNC 1 for loading the raw part.
PR[36:C1 place rt ofst]	Retreat offset for place position inside CNC 1 after loading the raw part.
PR[37:C1 pick pos]	Pick position inside CNC 1 for unloading the finished part.
PR[38:C1 pick ap ofst]	Approach offset for pick position inside CNC 1 for unloading the finished part.
PR[39:C1 pick rt ofst]	Retreat offset for pick position inside CNC 1 after unloading the finished part.
CNC 2	
PR[40:Inside C2 T1]	Position inside CNC 2 for loading the raw part. Also used for unloading the finished part if single gripper.
PR[41:Inside C2 T2]	Position inside CNC 2 for unloading the finished part if dual gripper.
PR[42:Outside C2 T1]	Position just outside CNC 2 for loading the raw part. Also used for unloading the finished part if single gripper.
PR[43:Outside C2 T2]	Position just outside CNC 2 for unloading the finished part if dual gripper.

Table 14–11. Position Registers (Cont'd)

Register #	Description
PR[44:C2 place pos]	Place position inside CNC 2 for loading the raw part.
PR[45:C2 place ap ofst]	Approach offset for place position inside CNC 2 for loading the raw part.
PR[46:C2 place rt ofst]	Retreat offset for place position inside CNC 2 after loading the raw part.
PR[47:C2 pick pos]	Pick position inside CNC 2 for unloading the finished part.
PR[48:C2 pick ap ofst]	Approach offset for pick position inside CNC 2 for unloading the finished part.
PR[49:C2 pick rt ofst]	Retreat offset for pick position inside CNC 2 after unloading the finished part.
CNC 3	
PR[50:Inside C3 T1]	Position inside CNC 3 for loading the raw part. Also used for unloading the finished part if single gripper.
PR[51:Inside C3 T2]	Position inside CNC 3 for unloading the finished part if dual gripper.
PR[52:Outside C3 T1]	Position just outside CNC 3 for loading the raw part. Also used for unloading the finished part if single gripper.
PR[53:Outside C3 T2]	Position just outside CNC 3 for unloading the finished part if dual gripper.
PR[54:C3 place pos]	Place position inside CNC 3 for loading the raw part.
PR[55:C3 place ap ofst]	Approach offset for place position inside CNC 3 for loading the raw part.
PR[56:C3 place rt ofst]	Retreat offset for place position inside CNC 3 after loading the raw part.
PR[57:C3 pick pos]	Pick position inside CNC 3 for unloading the finished part.
PR[58:C3 pick ap ofst]	Approach offset for pick position inside CNC 3 for unloading the finished part.
PR[59:C3 pick rt ofst]	Retreat offset for pick position inside CNC 3 after unloading the finished part.
CNC 4	
PR[60:Inside C4 T1]	Position inside CNC 4 for loading the raw part. Also used for unloading the finished part if single gripper.
PR[61:Inside C4 T2]	Position inside CNC 4 for unloading the finished part if dual gripper.
PR[62:Outside C4 T1]	Position just outside CNC 4 for loading the raw part. Also used for unloading the finished part if single gripper.
PR[63:Outside C4 T2]	Position just outside CNC 4 for unloading the finished part if dual gripper.
PR[64:C4 place pos]	Place position inside CNC 4 for loading the raw part.
PR[65:C4 place ap ofst]	Approach offset for place position inside CNC 4 for loading the raw part.
PR[66:C4 place rt ofst]	Retreat offset for place position inside CNC 4 after loading the raw part.
PR[67:C4 pick pos]	Pick position inside CNC 4 for unloading the finished part.

Table 14–11. Position Registers (Cont'd)

Register #	Description
PR[68:C4 pick ap ofst]	Approach offset for pick position inside CNC 4 for unloading the finished part.
PR[69:C4 pick rt ofst]	Retreat offset for pick position inside CNC 4 after unloading the finished part.

The template TP programs use the following robot input/output:

Table 14–12. Robot Input/Output

Robot Input/Output #	Description
RI[1:Gripper Closed]	Signal that gripper is closed.
RI[2:Gripper Opened]	Signal that gripper is opened.
RI[3:Finish Closed]	Signal that finished part gripper is closed.
RI[4:Finish Opened]	Signal that finished part gripper is opened.
RI[5:Part presence gripper 1]	Signal that part is present in gripper.
RI[6:Part presence gripper 2]	Signal that part is present in finished part gripper.
RO[1:Gripper Open]	Signal to open gripper.
RO[2:Gripper Close]	Signal to close gripper.
RO[3:Finish Grip Open]	Signal to open finished part gripper.
RO[4:Finish Grip Close]	Signal to close finished part gripper.

There are 100 user alarms which can be defined. The template TP programs will use UALM statements as follows:

Table 14–13. User Alarms

User Alarm	Alarm Message	Severity	Status
UALM[1]	Incorrect Service		
UALM[2]	Wait timed out		Not currently used
UALM[3]	Part not in gripper	PAUSE	
UALM[4]	Part not at pickup position		Not currently used
UALM[5]	Part not at place position		Not currently used
UALM[6]	Drawers not locked in position		
UALM[7]	Invalid drawer		
UALM[8]	Gripper malfunction	PAUSE	

Table 14–13. User Alarms (Cont'd)

User Alarm	Alarm Message	Severity	Status
UALM[9]	No parts to pick		
UALM[10]	Part not found by vision		

14.2.5 Customizing Machine Tool Easy Setup

You may customize the Machine Tool Easy Setup as follows:

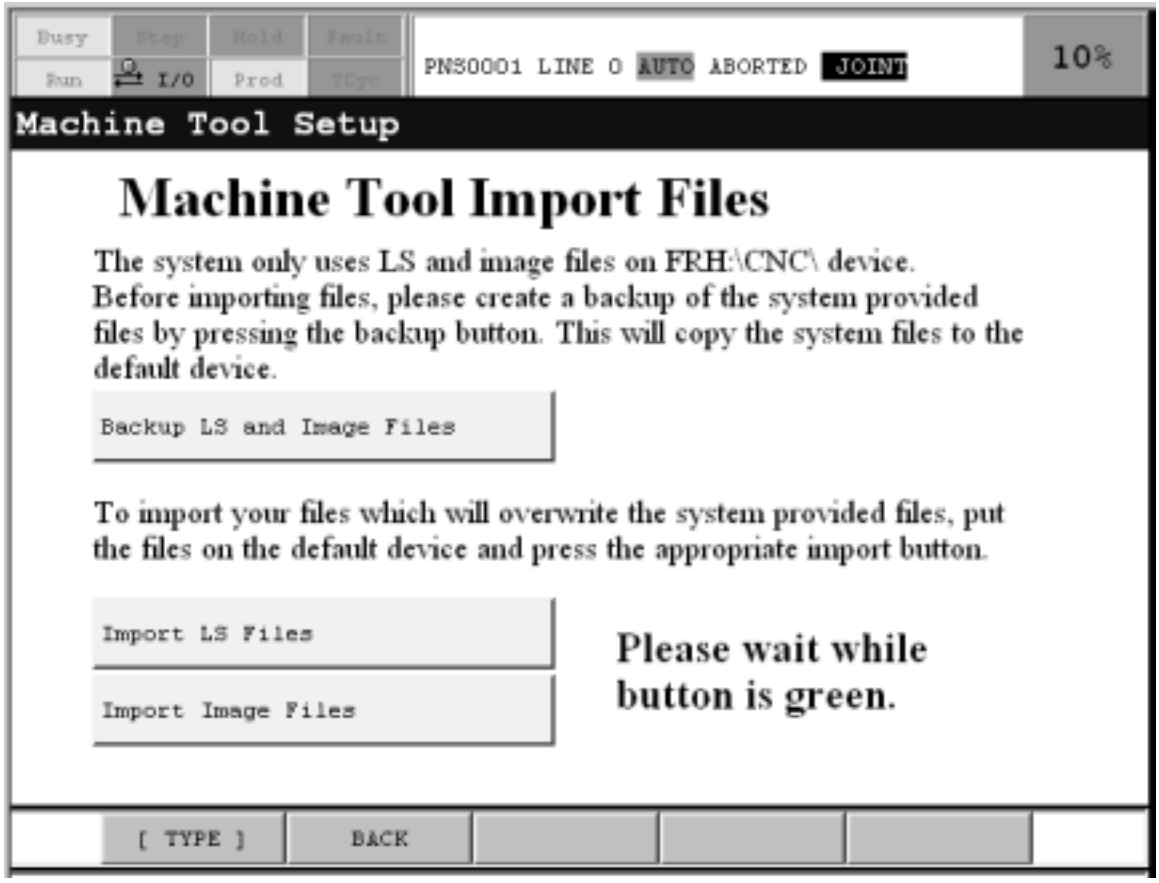
- Modify the TP programs which are used as templates.
- Modify the images shown in the wizards.

The TP programs are stored as LS files on a protected device. The image files are stored as GIF, JPG, and PNS files on a protected device. Before importing files, please create a backup of the system provided files using Procedure 14-1 .

Procedure 14-1 Backup TP Programs and Images

1. Insert an empty memory card or USB stick.
2. Make this the default device using the File Menu.
3. Press MENUS.
4. Select SETUP.
5. Press F1,[TYPE]
6. Select iPendant Setup. You will see the iPendant General Setup Screen.
7. Select Machine Tool Setup. You will see a screen similar to the following.

Figure 14–8. Machine Tool Setup



8. **To backup the files** , select the button labelled “Backup LS and Image Files”.
9. Wait for confirmation popup.
10. Store the memory card or USB stick in a safe place.

14.2.5.1 Modify the TP programs

One of the easiest ways to customize Machine Tool Easy Setup is to modify the TP programs which are used as templates. The first time Machine Tool Easy Setup is entered, the default programs will be loaded from FRH:\CNC\ device using ASCII Upload (.LS files). The benefit of using Ascii Upload is that Register comments found in the .LS files are also uploaded. As selections occur in the wizard, the appropriate programs are loaded from FRH:\CNC\ device. Therefore, by importing your files on FRH:\CNC\ device, your template programs will be loaded. Once a template is loaded, it is expected that the customer will modify the program.

The LS programs will not overwrite any existing TP programs which are loaded. They are used for templates when the TP program is not already loaded. You may need to delete a program in order to see your new template.

Please refer to Table 14–6 which shows the list of TP programs used by the wizard.

To import your LS files which will overwrite the system provided files, use Procedure 14-2 .

Procedure 14-2 Import TP programs

Conditions

- You have backed up the system provided files, using Procedure 14-1

Steps

1. Insert memory card or USB stick which contains only your changed LS files.
2. Set this as the default device using the File Menu.
3. Select the Machine Tool Setup menu, see procedure above.
4. **To import the LS files** , select the button labelled “Import LS Files”.
5. Wait for confirmation popup.

14.2.5.2 Modify the Images

Another easy way to customize Machine Tool Easy Setup is to modify the images which are used in the wizards. To import your image files which will overwrite the system provided files, use Procedure 14-3 .

Procedure 14-3 Import Image Files

Conditions

- You have backed up the system provided files, using Procedure 14-1

Steps

1. Insert memory card or USB stick which contains only your changed image files.
2. Set this as the default device using the File Menu.
3. Select the Machine Tool Setup menu, see procedure above.
4. **To import the image files** , select the button labelled “Import Image Files”.
5. Wait for confirmation popup.
6. The iPendant caches the images, so you may need to recycle power to see your images.

The following tables show the list of images used in the wizards.

Table 14–14. Process Navigator

Web Page	Description	Images
mtnav1.stm	Define the Cell	tbmach.gif tbgrip.gif tbinfeed.gif tbprepr.gif tbmach.gif tbpostpr.gif tbpallet.gif tbvisual.gif
mtnav1.stm	Teach Cell	tbhome.gif tbmaint.gif tbpns.gif tbrefpos.gif tbmain.gif
mtnav3.stm	Run Production	tbprod.gif tbmach.gif tbvisual.gif tbmanual.gif

Table 14–15. Setup Machine List

Web Page	Description	Images
mtcell.stm	Setup Machine List	mtcnc.jpg
mtiocf.stm	Configure I/O	tbfiles.gif
mtfanu.stm	Machine Tool Connection	
mtio.stm	Select I/O Device	mtiolink.jpg
mtiold.stm	Load I/O Configuration	
mthcom.stm	Setup Data Communication	
mthtag.stm	Setup Data Communication Client Tags	

Table 14–16. Setup Gripper

Web Page	Description	Images
mtgrip.stm	Setup Gripper: Select Single or Dual	mtgrip.jpg
mtutba.stm	Setup Tool Frame: Show Base Frame	mtutbase.png
mtutno.stm	Setup Gripper Frame: Show Base Frame and Select Gripper Number	mtutbase.png

Table 14–16. Setup Gripper (Cont'd)

Web Page	Description	Images
mtuta1.stm	Setup Tool Frame: Record First Approach Point	mtutapr1.png
mtuta2.stm	Setup Tool Frame: Record Second Approach Point	mtutapr2.png
mtuta3.stm	Setup Tool Frame: Record Third Approach Point	mtutapr3.png
mtuto1.stm	Setup Tool Frame: Record Orient Origin Point	mtutorg1.png
mtuto2.stm	Setup Tool Frame: Record X Direction Point	mtutorg1.png
mtuto3.stm	Setup Tool Frame: Record Z Direction Point	mtutorg1.png
mtutoj.stm	Setup Tool Frame: Jog the robot in the frame	mtutorg1.png
mtutds.stm	Setup Tool Frame: Enter Comment and Show Frame	
mtutde.stm	Setup Tool Frame: Enter Comment and Direct Entry of Frame	
mtuted.stm	Teach Gripper Macros	

Table 14–17. Setup Part Infeed Station

Web Page	Description	Images
mtinty.stm	Setup Part Infeed Station: Select the type of station	
mtinba.stm	Setup Part Infeed Station: Show Base Frame	mtufbase.png
mtinno.stm	Setup Part Infeed Station: Show Base Frame and Select Drawer Number	mtufbase.png
mtined.stm	Teach Part Infeed Program	

Table 14–18. Add a Pre Process Operation

Web Page	Description	Images
mtpre.stm	Add a Pre Process Operation	
mtpred.stm	Teach Pre Process Program	

Table 14–19. Setup Machine Station

Web Page	Description	Images
mtmsba.stm	Setup Machine Station: Show Base Frame	mtufbase.png
mtmsno.stm	Setup Machine Station: Show Base Frame and Select Machine Number	mtufbase.png
mtmsed.stm	Teach Machine Load/Unload Program	

Table 14–20. Add a Post Process Operation

Web Page	Description	Images
mtpost.stm	Add a Post Process Operation	
mtpoed.stm	Teach Post Process Program	

Table 14–21. Setup Finished Part Station

Web Page	Description	Images
mtfity.stm	Setup Finished Part Station: Select the type of station	
mtfiba.stm	Setup Finished Part Station: Show Base Frame	mtufbase.png
mtfino.stm	Setup Finished Part Station: Show Base Frame and Select Drawer Number	mtufbase.png
mtfied.stm	Teach Finished Part Program	

Table 14–22. Setup User Frame

Web Page	Description	Images
mtufp1.stm	Setup User Frame: Record Origin Point	mtuforg.png
mtufp2.stm	Setup User Frame: Record X Direction Point	mtufx.png
mtufp3.stm	Setup User Frame: Record Point on X-Y Plane	mtufxy.png
mtufp4.stm	Setup User Frame: Record System Origin Point	mtuforg.png
mtufpj.stm	Setup User Frame: Jog the robot in the frame	mtuforg.png
mtufds.stm	Setup User Frame: Enter Comment and Show Frame	
mtufde.stm	Setup User Frame: Enter Comment and Direct Entry of Frame	

Chapter 15

*i*RCALIBRATION OPTIONS

Contents

Chapter 15	*i*RCALIBRATION OPTIONS	15–1
15.1	OVERVIEW	15–3
15.1.1	Overview	15–3
15.1.2	Torque Observer Sensor and Electrical Continuity Sensing	15–5
15.1.3	*i*RCalibration TP Instructions	15–6
15.2	*i*RCALIBRATION MASTER SET	15–8
15.2.1	Overview	15–8
15.2.2	Preliminary Setup	15–9
15.2.3	*i*RCalibration Master Set Setup	15–12
15.2.4	Creating a Robot Calibration Teach Pendant Program	15–22
15.2.5	Running the *i*RCalibration Master Set Program	15–35
15.2.6	Updating the Master Count	15–37
15.2.7	Viewing the Calibration Result	15–38
15.2.8	Troubleshooting *i*RCalibration Master Set	15–40
15.3	*i*RCALIBRATION MASTER RECOVERY	15–42
15.3.1	Overview	15–42
15.3.2	Preliminary Setup	15–43
15.3.3	*i*RCalibration Master Recovery Setup	15–46
15.3.4	Creating a Robot Calibration Teach Pendant Program	15–57
15.3.5	Running the *i*RCalibration Master Recovery Program	15–70
15.3.6	Single Axis Mastering Recovery	15–72
15.3.7	Updating the Master Count	15–73
15.3.8	Viewing the Calibration Result	15–74
15.3.9	Troubleshooting *i*RCalibration Mastering	15–76
15.4	iRCALIBRATION TCP SET	15–78
15.4.1	Overview	15–78
15.4.2	Preliminary Setup	15–78
15.4.3	*i*RCalibration TCP Set Setup	15–83
15.4.4	Running the *i*RCalibration TCP Set Program	15–103
15.4.5	Displaying the UTool Log Screen	15–105

15.4.6 *i*RCalibration TCP Set Error Recovery 15–107
15.4.7 *i*RCalibration TCP Set Troubleshooting 15–108
15.5 iRCALIBRATION TCP SHIFT 15–109
15.5.1 Overview 15–109
15.5.2 Preliminary Setup 15–109
15.5.3 *i*RCalibration TCP Shift Setup 15–113
15.5.4 Running the *i*RCalibration TCP Shift Program 15–133
15.5.5 Displaying the UTool Log Screen 15–135
15.5.6 *i*RCalibration TCP Shift Error Recovery 15–137
15.5.7 *i*RCalibration TCP Shift Troubleshooting 15–138
15.6 *i*RCALIBRATION FRAME SET 15–139
15.6.1 Overview 15–139
15.6.2 Preliminary Setup 15–140
15.6.3 *i*RCalibration Frame Set Setup 15–140
15.6.4 Running the *i*RCalibration Frame Set Program and Updating the User Frame 15–156
15.6.5 Using *i*RCalibration Frame Set with Nominal Data 15–158
15.6.6 Displaying the UFrame Log Screen 15–158
15.6.7 Troubleshooting *i*RCalibration Frame Set 15–160
15.7 iRCALIBRATION FRAME SHIFT 15–163
15.7.1 Overview 15–163
15.7.2 Preliminary Setup 15–164
15.7.3 *i*RCalibration Frame Shift Setup 15–167
15.7.4 Running the *i*RCalibration Frame Shift Program and Updating the User Frame 15–188
15.7.5 Using *i*RCalibration Frame Shift with Dynamic-Uframe 15–191
15.7.6 Displaying the UFrame Log Screen 15–191
15.7.7 Troubleshooting *i*RCalibration Frame Shift 15–193

15.1 OVERVIEW

15.1.1 Overview

The *i*RCalibration options are a family of products that relate to robot accuracy. Refer to Table 15–1 for information about the functions each option performs.

Table 15–1. Relation of *i*RCalibration Options

***i*RCalibration Software Option**	**Documentation**	**Setup**	**Production (Suitable for Automatic Operation)**	**Function**
Vision TCP Set	Vision	Initial TCP	n/a	Initial TCP
Vision Shift	Vision	Adjust program points	n/a	Adjust program points
Vision Frame Set	Vision	Initial User Frame	n/a	Initial user Frame
Vision Frame Shift	Vision	Recover User Frame	Maintain User Frame	Recover user Frame
Vision Mastering	Vision	Initial mastering (J2–J5)	n/a	Initial mastering (J2–J5)
TCP Set (formerly TCPCal)	This manual	Initial TCP**	n/a	Initial TCP
TCP Shift (formerly TCPCal)	This manual	Recover TCP	Maintain TCP	Recover TCP
Frame Set	This manual	Initial user frame	n/a	Initial user frame
Frame Shift (formerly CellCal)	This manual	Recover user frame	Maintain user frame	Recover user frame
Master Set(formerly RobotCal)	This manual	Initial mastering (J2–J5)	n/a	Initial mastering (J2–J5)

Table 15–1. Relation of *i*RCalibration Options (Cont'd)

***i*RCalibration Software Option**	**Documentation**	**Setup**	**Production (Suitable for Automatic Operation)**	**Function**
Master Recovery (formerly RobotCal)	This manual	Master recovery	n/a	Recover mastering
Multi-Arm Frame Set (formerly MultiCal)	MultiARM manual	Coordinated Motion calibration	n/a	Coordinated Motion calibration

** Frame Shift, TCP Shift, and Master Recovery functions for production maintenance and recovery require some operation during the setup phase.

Use the following steps to set up and use the *i*RCalibration options:

1. Install robot and workcell hardware and software, including an *i*RCalibration Mastering or *i*RCalibration TCP fixture (you can use the same fixture or different fixtures for *i*RCalibration Mastering and *i*RCalibration TCP).
2. Use *i*RCalibration Master Set for initial Mastering; Use *i*RCalibration Master Recovery for mastering reference. *i*RCalibration Master Set can also compute the initial TCP value.
3. Use *i*RCalibration TCP Set for initial TCP setting; Use *i*RCalibration TCP Shift for TCP referencing.*i*RCalibration Mastering and *i*RCalibration TCP do not necessarily need to use the same tooling and fixturing, so a separate referencing of *i*RCalibration TCP is needed.
4. Use *i*RCalibraiton Frame Set to set the initial user frame; Use *i*RCalibration Frame Shift to set the frame referencing.*i*RCalibration Frame Shift can also be used as a quick check to detect misaligned fixtures or tooling.
5. Teach all required robot programs, using *i*RCalibration TCP Shift periodically to verify and correct the TCP whenever inadvertent collisions occur during jogging and test running of programs.
6. During production, execute *i*RCalibration TCP Shift in automatic mode to maintain an accurate TCP or to signal when the TCP becomes inaccurate.
7. Use *i*RCalibration Frame Shift to quickly check the location of one or more points on the tooling, fixture, or part.
8. When recovering from a crash or major hardware maintenance:
   - If mastering is lost on one or more axes, use *i*RCalibration Master Recovery to recover mastering.
   - If the TCP becomes mis-aligned, recover the TCP with *i*RCalibration TCP Shift.

- If the cell fixturing becomes mis-aligned, recover the user frame with *i*RCalibration Frame Shift.

15.1.2 Torque Observer Sensor and Electrical Continuity Sensing

The Torque Observer Sensor (TOS) can detect the contact between a robot tool and an object without the use of external sensors, signals, or I/O. This technology uses motor torque signals and other information to detect such contact using a very light contact force. TOS is especially useful for applications such as waterjet cutting where other detection methods might not be practical. There are two kinds of TOS: TOS Wrist and TOS All Axes.

TOS All Axes is the default method and is recommended when the I/O method is not used.

TOS All Axes monitors the motor torque signals for all axes. This method requires more contact force for contact detection, but it does not have the limitation of TOS Wrist regarding the TCP placement. TOS All axes is recommended for all standard six-axis articulated robots, but naturally, it is generally much more sensitive and requires less force to detect contact with small robots than with large robots. TOS is not available for all robots. Refer to the Software Release Note for specifics. The torque threshold setting in the detection schedule may need to be adjusted in some cases

The TOS All Axes method performs a small learning motion before each search, so the tooling should have clearance to allow this learning motion to complete without contact. The learning speed and distance can be changed on the Detection Schedule Setup Screen. Refer to Table 15–10 , Table 15–16 , or Table 15–25 .

TOS Wrist monitors the signals from the wrist axes so that contact between the tool and an object is detected very quickly using very light contact force. For this reason, if the line between the contact point on the tool and the wrist center is parallel to the search direction, the detection requires much more force. Normally this situation does not occur for tooling such as waterjet cutting where the TCP is away from the faceplate (J6) axis and where the tool shaft is not parallel to the faceplate (J6) axis, but care should be taken if you are using TOS with a tool that does not meet this constraint. TOS Wrist works best with six-axis, smaller-payload robots.

TOS Wrist is very sensitive and can be more prone to false detections in certain cases than the more robust TOS All Axes method. TOS Wrist is not recommended except for special cases where it can verified that false detections will not occur.

All of the applications that use TOS technology can use I/O as an alternate signal to TOS. These applications can work on any robot regardless of tooling configuration or payload by using the I/O detection method. Many processes, such as arc welding, some laser cutting, and some spot welding, have an I/O contact detection means as part of the process equipment. For these processes, use the I/O method instead of the TOS unless some of the contact points (tool orientation, for example) do not have the I/O contact detection capability.

If the I/O signal detection method is used by the *i*RCalibration option, then a corresponding output signal is provided so that user circuitry can enable the input signal in the situations where it is desired

to have the I/O active only while the *i*RCalibration option is being used. The I/O method can be used by charging the tooling (for example, for arc welding or laser cutting) and detecting the change from a being in a charged state to being in a grounded state where the contact plate is grounded to the I/O ground. Alternatively, the contact plate can be electrically isolated and charged and the contact detected by grounding the contact plate through the normal grounding capability on the robot tooling. The advantage of the charged plate method is that no additional circuitry needs to be on the robot arm itself.

Note Because of compliance, robot dressout, axis configuration, and other factors, the Torque Observer Sensor may not yield the same results in different areas of the workcell with the same threshold setting. This is generally not an issue for *i*RCalibration TCP Shift and *i*RCalibration Master Recovery because they use the same configuration for mastering and recovery.

For initial TCP and initial Robot Calibration, each detection motion has a similar and opposite detection motion, thereby minimizing any such effects. However, for *i*RCalibration Frame Shift where large configuration changes of the robot axes may exist in some cases between mastering and recovery motions, care should be taken in the choice of detection motion. For this case, having symmetric and opposing detections will minimize the error.

15.1.3 *i*RCalibration TP Instructions

```
- Calib Start[n]
- Utool Start[n]
- Uframe Start[n]
- Detect Point[m] frame[dir]
- Detect Circle[m,xyoff,zoff] frame[dir]
- Detect Joint[m] axis[num,dir]
- Utool End type
- Uframe End type
- Update Frame
- Update TCP
- Calib End
```

The *i*RCalibration Options all use TP programs as a basis of their execution. Each *i*RCalibration Option has its own start instruction, Calib Start for *i*RCalibration Mastering, UTool Start for *i*RCalibration TCP, and UFrame Start for *i*RCalibration Frame. All the start instructions include a schedule number that allows different acceptance tolerances, limits, and register numbers associated with that option.

Detection Instructions

Two detection instructions, **Detect Point** and **Detect Circle,** are used by all the *i*RCalibration Options. A third instruction, **Detect Joint,** is supported only by *i*RCalibration Mastering for single axis recovery.

The **Detect Point** instruction has three parameters:

- The first parameter is a detection schedule index
- The second parameter is the search frame, one of WORLD, UFRAME, or UTOOL.
- The last parameter is search direction, plus or minus XYZ in the specified search frame. During the search motion the robot will move along the search direction until contact is detected or until the search distance specified in the detection schedule is reached.

The **Detect Circle** instruction has five parameters:

- The first parameter is the detection schedule index.
- The second parameter is the XY Offset which is the distance perpendicular to the search direction the robot will move in order to detect the surface.
- The third parameter is the Z offset which is the offset past the plate surface that the robot will move before performing the circle search.
- The fourth parameter is the search frame, one of WORLD, UFRAME, or UTOOL.
- The fifth parameter is the search direction, plus or minus XYZ in the specified search frame. This is the direction the robot will move to detect the plate surface if XY Offset and Z offset are non-zero. If the XY Offset and Z offset are zero then the robot must be taught inside the hole and no search for the plate surface is performed.

Update Instructions

The **Update Tool** instruction will apply a partial update to a TCP calibration recovery program. All the detections before the Update Tool are applied. This is useful in determining a Tool Z direction, for instance before searching the X or Y directions to ensure the proper Z height for the latter instructions. Multiple Update Tool instructions are allowed.

Update Tool instructions are not included in the generated *i*RCalibration TCP Shift programs, but can be added manually after generation.

The **Update Frame** instruction, similar to the Update Tool instruction will apply a partial update to a UFrame calibration recovery program. All the detections before the Update Frame are applied. This is useful in determining a UFrame Z direction, for instance before searching the X or Y directions to ensure the proper Z height for the latter instructions. You could also do three Z searches then apply the Update Frame instruction to get the partial orientation of that plane. Then the remaining X and Y searches will be in the correct Z distance.

Update Frame instructions are not included in the generated *i*RCalibration Frame Shift programs, but can be added manually after generation.

15.2 *i*RCALIBRATION MASTER SET

15.2.1 Overview

Note *i*RCalibration Master Set is not available for all robots. Refer to Software Release Note for specifics.

*i*RCalibration Master Set, formerly RobotCal, is the 3rd generation of calibration software FANUC Robotics has offered. It follows the AccuSight (1989) and AccuCal (1997) products with updated user interface and TOS technology.

*i*RCalibration Master Set is an optional calibration option that can do initial calibration of a robot.

*i*RCalibration Master Set uses a wizard-style menu. It shows the steps that need be done to complete the robot calibration process. Each step of the process has its own status display that shows where you are in the setup process. You cannot skip steps and you cannot go to the next step without completing the current step. *i*RCalibration Master Set also includes a menu item, View Result, that lets you review the previous calibration result and log history.

- *i*RCalibration Master Set uses teach pendant programs for all execution. These programs are typically generated by *i*RCalibration Master Set, but can also be created or modified by the user.
- *i*RCalibration Master Set supports initial mastering
- *i*RCalibration Master Set supports schedule based sensor setup and can use a different schedule for different searches.
- Because *i*RCalibration Master Set uses very large axis rotations, the TCP and mastering values are very accurate.
- *i*RCalibration Master Set supports all robot motion groups.

It is recommended that *i*RCalibration Master Set calibration be performed on all robots before any points are taught. The axes that normally require calibration are J2, J3, J4, and J5. In general, J1 and J6 are not calibrated under initial calibration because a taught UFrame and UTool will completely correct for any error in these axes.

The same program used for *i*RCalibration Master Set can later be used to recover mastering for any or all robot axes, J1 through J6. If *i*RCalibration Master Set is not performed, but *i*RCalibration Master Recovery mastering is performed, it is recommended to both master and recover at least J2, J3, J4, and J5. It is not recommended to recover a different set of axes than were mastered, except it is OK to recover J1 and J6 as described in Table 15–3 . Also, see the special requirements regarding J2 for robots with linear J1 axis.

Hardware Requirements

The following hardware is required to use *i*RCalibration Master Set:

- Calibration tooling. Refer to Section 15.2.2 .
- Six-axis robot (except as indicated in the Software Release Note)
- User-supplied circuitry for using I/O sensor (for robots with delicate tooling)

If the standard tooling is used, *i*RCalibration Master Set can generate a program to perform initial mastering of robot axes 2 through 5, usually without any user point teaching. Standard tooling is available for most robots to support the default calibration program generated by *i*RCalibration Master Set.

Using *i*RCalibration Mastering

To use *i*RCalibration Master Set, you must do the following:

1. Install the *i*RCalibration Master Set software option and any required hardware.
2. Set up *i*RCalibration Master Set.
3. Create a robot calibration program.
4. Run *i*RCalibration Master Set to set up initial mastering.

15.2.2 Preliminary Setup

Standard *i*RCalibration Master Set Hardware

Standard *i*RCalibration Master Set hardware is available. It includes the following components:

- Calibration Tool to attach to the faceplate of the robot. An adapter plate may be required.
- Calibration Arm with one or two calibration plates to attach to the base of the robot
- I/O cable to attach to the robot EE connector

The standard hardware available includes the following:

- EE connector cable to support the I/O detection method of the Calibration Options. Cables are available to support both 5-pin and 24-pin EE connectors.
  - — EE-5770-020-040 (4 meters).
    Cable Assembly for End Effector Touch Sensor w / 24 Pin Connector
  - — EE-5770-021-025 (2.5 meters).
    Cable Assembly for End Effector Touch Sensor w / 5 Pin Connector
- Calibration tool assembly to mount to the robot faceplate. The calibration tool assembly consists of a base to mount to the faceplate, a supporting tube extending away from the base, an attached nonconducting block, and a calibration pointer supported by the non-conducting block. The angle

between the J6 axis and the calibration pointer is 70 degrees for the short tool and 45 degrees for the long tool. Multiple tools and adapters are available to support most robots.

1. MO-1800-943 Calibration Tool Assembly – Short
2. MO-1800-944 Calibration Tool Assembly – Long
3. MO-1800-945-001 Calib. Tool Adapt. - M710/20L
4. MO-1800-945-002 Calib. Tool Adapt. - R2000
5. MO-1800-945-003 Calib. Tool Adapt. - M900
6. MO-1800-945-004 Calib. Tool Adapt. - Arc Solution Arm
7. MO-1800-945-005 Calib. Tool Adapt. - Arc Tregasis
8. MO-1800-945-006 Calib. Tool Adapt. - Arc Binzel

- Calibration arm to mount to the robot base. The calibration arm assembly includes two calibration plates and a mounting bracket. *i*RCalibration Master Set, depending on robot model and configuration, uses one or both of the plates. Multiple calibration arms are available to support most robots.
  - — MO-1800-940 Calibration Arm Assembly –Standard
  - — MO-1800-941 Calibration Arm Assembly – Short

Note The standard calibration arm, MO-1800-940, can be used on any robot which is supported by the short calibration arm, MO-1800-941.

Table 15–2. Calibration Hardware

Robot Series	Calibration Arm	Calibration Tool	Adapter(s)
M6 (AM100iB)	MO-1800-941	MO-1800-943	
AM100iB - SA			MO-1800-945-004
M6-6S (AM100iB/6S)			
M16 (AM120iB)			
AM120iB - SA			MO-1800-945-004
M16/10L (AM120iB/10L)			
AM120iB/10L - SA			MO-1800-945-004
M710iB			MO-1800-945-002
M710iB/20L	MO-1800-940		
M710iB/20L –SA*			
R-2000 (R & P)		MO-1800-944	MO-1800-945-002
R-2000		MO-1800-943	
R-2000 - SA		MO-1800-944	
M-900 (350)			MO-1800-945-003
M-900 (600)			

*The M710iB/20L-SA does not currently have a Standard Calibration Tool assembly available.

Some robots, for example the top loader series of robots, do not have an available standard calibration arm. For these cases, *i*RCalibration Master Set can still generate the standard *i*RCalibration Master Set program if the user installs the standard calibration tool to the robot faceplate and installs a calibration plate at the proper location for the generated calibration program.

Robots that cannot do FRONT/BACK configuration change or FLIP/NOFLIP configuration change require two calibration plates. **For all robots**, the first calibration plate must be located parallel to the robot World Y-Z plane with the hole centered at the Uframe coordinates.

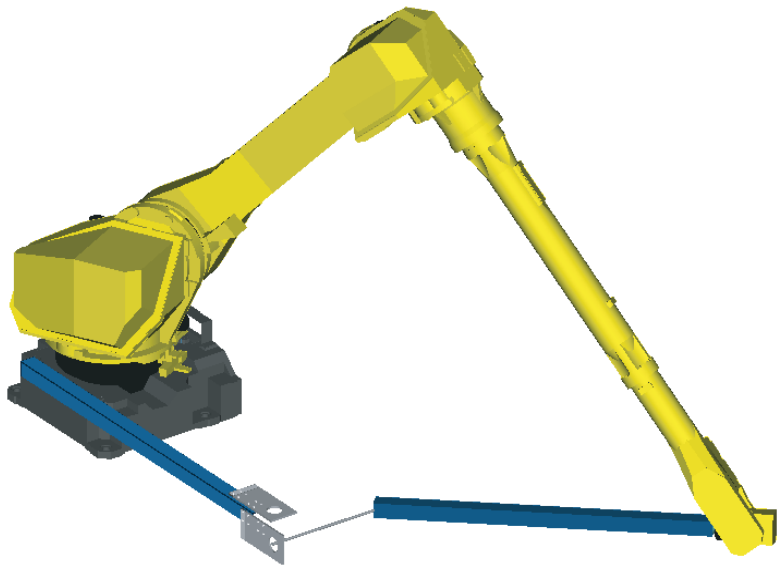
If the robot requires two plates, the second plate must be located in the X-Y plane with the hole centered at these values: (Uframe.x – 56, Uframe.y-82, Uframe.z+57). The two-plate method of initial calibration is more accurate for robots that cannot change configuration than the single plate method. The two-plate method generates 11 calibration points on the first plate and 6 on the second plate.

The Standard *i*RCalibration Master Set calibration plate(s) are attached (by a calibration arm) to the mastering holes located on front of the robot base. The calibration arm is located along the +Y direction of robot World frame. The calibration plates are not precisely located; *i*RCalibration Master Set will fine tune the actual location of the calibration plates.

Refer to Procedure 15-13 for information on installing and aligning the Touch Plate and for setting up the Touch Plate for use with *i*RCalibration Master Set.

Note If there is significant mastering correction or if the generated program positions are not centered on the calibration plate hole, you should run *i*RCalibration Master Set twice.

Figure 15–1. Standard *i*RCalibration Mastering Calibration Plates



15.2.3 *i*RCalibration Master Set Setup

- ***i*RCalibration Master Set.** This provides accurate initial calibration of the robot axes. It will compute the offset of the mastering data. You should update the mastering data after the *i*RCalibration Master Set program completes

Setting up and running *i*RCalibration Master Set includes the following:

- Setting up the Mastering Schedule. Refer to Table 15–3 for descriptions of the items on the Mastering Schedule screen.
- Setting up the Detection Schedule. Refer to Table 15–4 for descriptions of the items on the Detection Schedule screen.
- Creating the robot calibration program. Refer to Section 15.2.4 .
- Running the calibration program. Refer to Section 15.2.5 .
- Updating the master count (Master CT) with the calibration result obtained from running the robot calibration program. Refer to Section 15.2.6 .

Note There are 10 Mastering Schedules and these schedules are shared by iRCalibration Master Set and iRCalibration Master Shift options.

Table 15–3. *i*RCalibration Master Set Schedule Screen Description

ITEM	DESCRIPTION
Robot J1 - J9	The calibration for each axis can be enabled or disabled individually. The default values are good for normal operation For robots with all rotary joints, it is recommended that J2, J3, J4, and J5 all be TRUE. As described in the Calibrate TCP and Calibrate UFrame sections, J1 (and J2 for linear J1 robots), and J6 have special considerations. The normal case for mastering recovery is to use the same settings as for initial mastering. If you need to recover J1 and/or J6 (and/or J2 for linear J1 robots) then you must follow the instructions regarding TCP and UFrame.
Calibration Group Range 1 - 8 Default: 1	This item defines the robot group to be calibrated.
Calibrate TCP True/False Default: True	This item allows calibration software to solve for the TCP value. Normally Calibrate TCP is TRUE unless you are using a previously accurately calibrated TCP. If Calibrate TCP is TRUE, then the calibration field corresponding to J6 must be false. You can never calibrate or recover both the TCP and J6. For mastering recovery, if UTool has been maintained and is still valid at the time of mastering recovery, then Calibrate UTool can be FALSE and J6 can be TRUE.
Calibrate UFRAME True/False Default: True	This item allows calibration software to solve for the user frame. Note that this will make UFRAME be the center of the circle in the calibration plate and will only set uframe XYZ value. Normally, Calibrate UFrame is TRUE unless you are using a previously accurately calibrated calibration plate. If Calibrate UFrame is TRUE, then the calibration field corresponding to J1 must be false. For TopLoaders or other robots with a linear J1 axis then J2 must be false whenever Calibrate UFrame is TRUE You can never calibrate or recover both UFrame and J1. Additionally for robots with a linear J1 axis you can never calibrate or recover both UFrame and J2. Any J1 (and J2 for linear J1 robots) calibration error can be fully corrected by UFrame. If you use *i*RCalibration Frame Shift to maintain UFrame following *i*RCalibration Master Set mastering recovery then it is not necessary to recover J1 (and J2 for linear J1 robots). However, if programming causes these axis to move close to their limits then a run-time limit error may occur if these axes are not recovered. If you want to recover J1 (and J2 ror linear J1 robots) then you MUST mount the calibration plate in a fixed location or in a location that can be exactly duplicated at the time of recovery AND the exact UFrame found during initial mastering must be used at the time of recovery. Normally this is done by dedicating a specific UFrame number for mastering and mastering recovery.

Table 15–3. *i*RCalibration Master Set Schedule Screen Description (Cont'd)

ITEM	DESCRIPTION
Allowable mean error Units: mm Range: 0.01 - 100 Default:	This item is the maximum allowable mean error for the calibration computation.
Mean Error Register Range: 0 - 999 Default: 0	If the number is greater than 0, the calibration software will write the calibration mean error to the register.
Max. Error Register Range: 0 - 999 Default: 0	If this number is greater than 0, the calibration software will update the calibration maximum error to the register.
Use Current TCP True/False Default : FALSE	This item is used for calibration recovery. • **If this value is TRUE, and Calibrate TCP is FALSE**, then the current active UTool will be used for determining the calibration parameters. • **If this value is FALSE and Calibrate TCP is FALSE**, then the current UTool must be the same as was used during calibration mastering. • **If this value is TRUE, and the physical tool has changed by maintenance or other reasons** since*i*RCalibration Master Set was mastered, then TCPCal or some other method should be used to maintain the correspondence between the UTool value and the physical tool.

Detection Schedules

Note The 10 detection schedules are shared among *i*RCalibration TCP, *i*RCalibration Frame, and *i*RCalibration Mastering. Therefore, a change in detection schedules affects all of these *i*RCalibration options you have installed on your controller.

When you set up a detection schedule, you

- Determine the sensor
- Determine the detection motion
- Determine TOS

Note Default values are usually acceptable.

Table 15–4. *i*RCalibration Master Set Detection Schedule Description

ITEM	DESCRIPTION
Sensor Type Values: TOS Wrist, I/O, TOS ALL AXES Default: TOS ALL AXES	This item indicates the kind of sensor used for contact detection. The three modes are: • **TOS Wrist** (Torque Observer Sensor on Wrist) - the contact torque will be monitored, and only the observed torque on the wrist will be used to determine the contact position. When TOS Wrist is selected, no I/O connection is required for contact detection. TOS Wrist is very sensitive and can be more prone to false detections in certain cases than the more robust TOS All Axes method. TOS Wrist is not recommended except for special cases where it can verified that false detections will not occur. • **I/O** - the specified I/O will be monitored and only the state of this input will be used to determine the contact position. Use the detection port for contact detection with I/O. Refer to Detection Port item in this table for more information. • **TOS ALL AXES** (Torque Observer Sensor, All Axes) - The contact torque will be monitored on all robot axes, and only the observed torque will be used to determine the contact position. This is the default value. When TOS ALL AXES is selected, no I/O connection is required for contact detection.
Detection Port Values: RDI, DI, WDI, WSI	This item indicates the desired input port type. The port types are: • **RDI** - Robot Digital Input • **DI** - normal Digital Input • **WDI** - Weld Digital Input • **WSI** - Wire Stick circuit Input. The port type that you specify is followed by the desired port number that will go from OFF to ON when contact is detected. The WDI and WSI ports are typically used for arc welding. You will set this item up if you selected I/O as the Sensor Type. You do not need to set this up if you selected TOS Wrist or TOS All Axes as the Sensor Type.

Table 15–4. *i*RCalibration Master Set Detection Schedule Description (Cont'd)

ITEM	DESCRIPTION
Sensor Enable Values: RDO, DOUT, WDO, or WSO Default: RDO	This item indicates the I/O sensor enable digital output port type. The port types are: • **RDO** - Robot Digital Output • **DOUT** - normal Digital Output • **WDO** - Weld Digital Output • **WSO** - Wire Stick circuit Output. The port type that you specify is followed by the desired port number that will be used to enable the I/O detection circuit. The WDO and WSO ports are typically used for arc welding. You will set this item up if you selected I/O as the Sensor Type. Set the port number to 0 (zero) if you are not using this feature.
Detection Speed Units: mm/sec Range: 1.0 - 100 Default: 10	This item is the search speed for contact detection. This speed is used for both TOS and I/O searches. The value can be increased if cycle time is of great concern (although increasing the speed will also affect the resultant accuracy and repeatability) , or reduced if vibration or other disturbances cause *i*RCalibration Master Set to retry excessively.
Detection Distance Units: mm Range: 1 - 1000 Default: 50	This item is the maximum distance that the robot will move while looking for the signal. If no contact is detected within this distance during a search, an error message will be posted.
Return Speed Units: mm/sec Range: 5 - 5000 Default: 50	This item is the search return speed. This speed is used for both TOS and I/O searches.
Return to Start Default: Yes	When this item is set to YES, the robot will move back to the position before it starts the search motion. This item applies to DETECT POINT instructions only. When this item is set to NO, the robot will move back to a distance specified by the next item (Return distance)
Return Distance Units: mm Range: 1 - 1000 Default: 50	When return to start is set to FALSE, the robot will move back to the distance specified.
Return term type Default: FINE	This item is the termination type for return motion.
Air Cut Speed Units: mm/sec Range: 1 - 1000 Default: 100	This item is the non-search motion speed.

Table 15–4. *i*RCalibration Master Set Detection Schedule Description (Cont'd)

ITEM	DESCRIPTION
Learn Distance Units: mm Range: 0.5– 20.0 Default: 5.0	For "TOS All Axes" method only, there is a short motion before each search to identify dynamic torque values. Learn Distance is the distance in millimeters of this short motion, the direction is determined at the time of the motion.
Learn Speed Units: mm/sec Range: 0.5 – 50.0 Default: 10.0	For "TOS All Axes" method only, this is the speed of the learning motion before each search.
Sensitivity Tolerance Units: max. motor torque/sec. Range: 1 - 999 Default: 10	This item is the primary sensitivity tolerance of TOS Wrist and is not used for TOS All-Axes. It is the rate of change of the disturbance torque. The actual units vary with motor model, gear ratio, and amplifier current, and are affected by tooling length. However, the effect for calibration is largely independent of these factors and a value of 10 is considered normal operation for typical installations. You should not use a value of less than 10 or misdetections might occur. This value can be increased if false detections occur frequently. **Warning** **Be careful when you change this number. Changing the sensitivity tolerance might cause a false detection. This could injure personnel or damage equipment.**
Torque Threshold Units: max. motor torque Range: 1 to 999 Default: 100	For TOS All Axes, this item is the only measure of contact detection. During the learning motion a torque level is memorized and that level plus Torque Threshold must be exceeded for contact to be determined. If false detections are being experienced then this value should be increased. For TOS Wrist, this item is the level of disturbance torque that is used as another means for contact detection. The default value of 100 is considered normal operation for typical installations. If increasing the value of **Sensitivity Tolerance** does not make false detections disappear, then this value should be increased until the false detections disappear; then, reset the value of **Sensitivity Tolerance** .
Fit Tolerance Units: mm Range: 0.1 - 100 Default: 0.5	For Detect Circle instruction: The robot detects 4 points inside the circle and the calibration software will fit a circle to the 4 detected positions. The fit error is defined as the maximum distance from the 4 points to the circle. If the fitting error exceeds the tolerance, the calibration software will post an error.
Radius Tolerance Units: mm Range: 0.1 — 100 Default: 0.5	The Detect Circle instruction stores the circle's radius during mastering. At run time, if the computed circle radius differs from the stored radius value and difference exceeds the tolerance, the calibration software will post an error.

Table 15–4. *i*RCalibration Master Set Detection Schedule Description (Cont'd)

ITEM	DESCRIPTION
Number of Retry Range: 0 - 9 Default:	If the number of retry is >0, the calibration software will re-run the detect circle instruction if the current circle detection has either a fit error or a radius error.
Contact Position Register	If the position register number is any number other than 0, *i*RCalibration software will copy the contact position to the position register.

15.2.3.1 Setting Up *i*RCalibration Master Set

*i*RCalibration Master Set uses a wizard-style menu. It shows the steps that need be done to complete the robot calibration process. Each step of the process has its own status display that shows where you are in the setup process. You cannot skip steps and you cannot go to the next step without completing the current step.

*i*RCalibration Master Set also includes a menu item, View Result, that lets you review the previous calibration result and log history. Both *i*RCalibration Master Set and *i*RCalibration Master Shift share the same calibration log.

*i*RCalibration Master Set should be used:

- Before you teach any points or run any programs on a new robot you should select *i*RCalibration Master Set to generate extremely accurate calibration data. After you run the calibration program you should use the Update menu to update the robot's mastering data.

Procedure 15-1 Setting Up *i*RCalibration Master Set

Conditions

- The *i*RCalibration Master Set software option is loaded.
- Preliminary setup is complete. Refer to Section 15.2.2 .

Steps

1. Press MENUS.
2. Select UTILITIES.
3. Press F1, [TYPE].
4. Select iRCalibration. You will see a screen similar to the following:

```
UTILITIES iRCalibration
       1 TCP Shift
       2 TCP Set
       3 Frame Shift
       4 Frame Set
       5 Master Recovery
       6 Master Set
```

5. Select iRCalibration Master Set and press ENTER. You will see a screen similar to the following:

```
 iRCalibration Master Set
1 Robot: Group [1]  Device: Sensor
2 Calibration Type: New Calibration
3 Setup            (Not Done)
4 Teach Points     (Not Done)
5 Create Program   (Not Done)
6 Run: RCAL_G1     (Not Done)
7 Update Master CT (Not Done)
8 View Result
```

6. Move the cursor to Robot: Group [1], type the group number of the motion group you want to calibrate, and press ENTER.

Note Device is always Sensor, and the Calibration Type is always "New Calibration". You cannot change them.

7. **Specify the Calibration Program Name**. The *i*RCalibration Master Set Setup menu allows you to specify the robot calibration teach pendant program name, select and set the *i*RCalibration Master Set calibration schedule and detection schedule.

Note The Group number is provided for reference in the SETUP menu. You cannot change it in this menu.

a. Move the cursor to Setup, and press ENTER. You will see a screen similar to the following:

```
Mastering Setup
      1 Program name: RCAL_G1      Group[1]
      2 Mastering Schedule:  1
      3 Detect Schedule:    1
```

b. The default program name is RCAL_Gx, where x is the group number. If you want to specify a different program name, press ENTER and use the function keys.

8. Select the Mastering Schedule and set schedule detail items. Move the cursor to Mastering Schedule. Refer to Table 15–3 for information on the *i*RCalibration Master Set Schedule items.

a. To change the number of the Mastering Schedule, move the cursor to the schedule number, type the number of the schedule you want, and press ENTER.

b. To see the detail screen for the selected schedule, press F4, SELECT, or press ENTER. You will see a screen similar to the following:

```
 Mastering Schedule
                                  1/10
  Schedule Number [ 1]
  Robot
    J1:FALSE   J2:TRUE   J3:TRUE
    J4:TRUE    J5:TRUE   J6:FALSE
    J7:FALSE   J8:FALSE  J9:FALSE
Calibration Group:     1
Calibrate TCP:         TRUE
Calibrate UFrame:      TRUE
Allowed Mean Error:    30.00  mm
Mean Error Register    0
Max Error Register    0
Use Current TCP:        TRUE
```

To calibrate a specific joint angle of the robot, select the axis and set it to TRUE:

- Move the cursor to select each item and set it as desired.
- Press F4, CHOICE, to choose from more than one setting for a particular item.

Refer to Table 15–3 for information on the iRCalibration Master Set Schedule items.

When you have finished setting up the *i*RCalibration Master Set schedule, press PREV to return to the *i*RCalibration Master Set Setup Menu:

```
Mastering Setup
       1 Program name: RCAL_G1      Group[1]
       2 Mastering Schedule:  1
       3 Detect Schedule:    1
```

9. Select the Detection Schedule and Set Detection Schedule detail items. Move the cursor to Detect Schedule.

a. To change the number of the Detect Schedule, move the cursor to the schedule number, type the number of the schedule you want, and press ENTER.

b. To set up the detection schedule detail items, press F4, SELECT. You will see a screen similar to the following:

```
Detection Schedule
                                    1/17
  Detection Schedule [ 1]
  Sensor Type:              I/O
  I/O Setup
    Detection Port:         DI   [  1]
    Sensor Enable:          DOUT[  1]
  Motion Setup
    Detection Speed:        10.000  mm/sec
    Detection Distance:    100.000  mm
    Return Speed:           50.00   mm/sec
    Return to Start:       TRUE
    Return Distance:         0.000  mm
    Return Term Type:       Fine
    Learn Speed:            10.000  mm/sec
    Learn distance:          5.000  mm
    Air Cut Speed:         100.000  mm
  TOS Setup
    Sensitivity:            100.000
    Torque Threshold:       100
  Circle Setup
    Fit Tolerance:          1.000
    Radius Tolerance:       .500
    Number of Retry:        2
  Contact Position Register:   0
```

Note The default settings are good for most applications.

Set each item as desired. Move the cursor to the item you want to change and type the value.

Refer to Table 15–4 for information about detection schedule setup detail items.

10. When setup is complete, press F1, DONE. You will see the *i*RCalibration Master Set main menu with Setup (Done) in its status display.

```
iRCalibration Master Set
1 Robot: Group [1]  Device: Sensor
2 Calibration Type: Master Recovery
3 Setup             (Done    )
4 Teach Points      (Not Done)
5 Create Program    (Not Done)
6 Run: RCAL_G1      (Not Done)
7 Update Master CT  (Not Done)
8 View Result
```

Note You must press DONE to change the Setup status. If you press PREV to return to the TCPCal main menu, the Setup status display will not change.

If the teach pendant program already exists when you press DONE, you will see the following screen:

```
Program exists. Skip Teach/Create?
  Press F2 for YES
  Press F3 for NO
```

If the teach pendant program already exists and has been taught, and you want to skip the teach/create step, press F2, YES, and proceed directly to Run. Refer to Section 15.2.5 to run the program.

If the teach pendant program exists but you want to teach it, change it, or use another program, press F3, NO, and continue to the Teach Points menu item.

15.2.4 Creating a Robot Calibration Teach Pendant Program

In order to master a robot with *i*RCalibration Master Set, you must first create a robot mastering teach pendant program. You can create the program manually or you can have *i*RCalibration Master Set create the program for you automatically. When *i*RCalibration Master Set setup is complete, the *i*RCalibration Master Set main menu will be displayed with the cursor at Teach Points.

For most six-axis robots

- you can have *i*RCalibration Master Set create a standard *i*RCalibration Master Set Calibration program automatically,
- you can customize an automatically generated program, or
- you can create your own custom program.

If yours is not a six-axis robot, you can create a custom program, but *i*RCalibration Master Set cannot create a standard *i*RCalibration Master Set program automatically.

If a robot is not supported for the Standard *i*RCalibration Master Set program creation, the menu goes directly to the CUSTOM calibration menu. You can also go to this page by pressing F4, CUSTOM, at the Program Creation menu.

If you cannot use the Standard *i*RCalibration Master Set generated program because of robot limits, workspace restrictions, or other reasons, you should use the CUSTOM program creation method. You can touch up generated points from either the Standard or Custom generated programs to meet the specific restrictions of the workcell. However, the final program must contain significant rotations of all axes in order to get the best results.

To disable the Standard *i*RCalibration Master Set program creation menu, set the system variable $CB_VARS.$STDCAL_ENB = FALSE. If the Standard *i*RCalibration Master Set program creation

menu is disabled, when you select Teach Points from the main menu you will go directly to the custom *i*RCalibration Master Set program creation menu.

Teaching Points

Creating a standard *i*RCalibration Master Set program automatically can usually be done without teaching any points. If you are creating a custom program to perform or recover mastering with *i*RCalibration Master Set, you must first teach several points to be used in your robot mastering teach pendant program. Refer to Section 15.2.4.2 .

15.2.4.1 Creating a Standard *i*RCalibration Master Set Program Automatically

Table 15–5. Program Creation Screen Items for Standard *i*RCalibration Master Set Programs

Item	Description
Program Name	The program name for the *i*RCalibration Master Set program creation cannot be changed in this menu. It can be changed in the Set Up menu. It is usually of the form "RCAL_G" followed by the group number of the robot being calibrated. The current active jog group is used in determining which group will be used.
UTool[10]: Uframe[9]:	*i*RCalibration Master Set program creation always uses UTool[10], UFrame[9]. **Program Defined** will appear if the corresponding frame was not previously set; that is, the frame xyzwpr components were all zero. For this case, when you press F2, CREATE, *i*RCalibration Master Set will set the frames to the identified values. The Uframe and UTool may be different for different robot models. You cannot change the UTool, Uframe, Detection Schedule, or Calibration Schedule number used by Standard *i*RCalibration Master Set. However, you can edit the generated program to use different Detection or Calibration schedules. You can change the Uframe or UTool number after the program has been created by using the Tool offset or Frame offset utilities.
	User Set will appear if *i*RCalibration Master Set had previously generated a program for the current group or if the Uframe or UTool were changed by the user. For this case, *i*RCalibration Master Set will use the current Uframe or UTool and will not change it when the program is generated.
Calibration Schedule[10] Detection Schedule[10]	The *i*RCalibration Master Set program creation always uses Calibration Schedule[10] and Detection Schedule[10].

Procedure 15-2 Creating a Standard *i*RCalibration Master Set Program Automatically

Conditions

- The *i*RCalibration Setup status is Done.
- All required calibration hardware has been installed. Refer to Section 15.2.2 .

- You are using a supported six-axis robot

Steps

1. Press MENUS.
2. Select UTILITIES.
3. Press F1, [TYPE].
4. Select iRCalibration. You will see a screen similar to the following:

```
UTILITIES iRCalibration
       1 TCP Shift
       2 TCP Set
       3 Frame Shift
       4 Frame Set
       5 Master Recovery
       6 Master Set
```

5. Select iRCalibration Master Set and press ENTER. You will see a screen similar to the following:

```
iRCalibration Master Set
1 Robot: Group [1]  Device: Sensor
2 Calibration Type: Master Recovery
3 Setup            (    Done)
4 Teach Points     (Not Done)
5 Create Program   (Not Done)
6 Run: RCAL_G1     (Not Done)
7 Update Master CT (Not Done)
8 View Result
```

6. Move the cursor to Teach Points and press ENTER. You will see a screen similar to the following:

```
Program Creation:
  Program Name:     RCAL_G1
  Robot Group:      1
  UTool[10]:        User Set
    X: -115.000 Y:    0.000 Z:  570.000
    W:    0.000 P:  -70.000 R:    0.000
  UFrame[9]:        User Set
    X:  176.000 Y:  600.000 Z: -335.000

  NOTE:  Calibration requires 1 plate

          DONE          CUSTOM
```

Note **If your robot is not supported**, you will see a screen similar to the following. Refer to Procedure 15-3 to create a custom program.

```
Program Creation

  Program Name:      RCAL_G1
  Robot Group:       1
  Calibration Sched: 1
  Detection Sched:   1
  Plate Direction:   –Y
  Plate Thickness:   3.000   mm
  Circle XY offset:  50.000  mm
  Circle Z offset:   10.000  mm
  Robot Calibration
    J1:FALSE    J2:TRUE     J3:TRUE
```

7. To use the automatically generated standard program, press DONE. To customize the standard program, press CUSTOM. Refer to Section 15.2.4.2 .

If all required positions are taught, the Teach Points status will be Done and the cursor will automatically move to the next step. Otherwise, the cursor will stay at Teach Points and its status will remain unchanged.

```
iRCalibration Master Set
1 Robot: Group [1]  Device: Sensor
2 Calibration Type: Master Recovery
3 Setup             (Done    )
4 Teach Points      (Done    )
5 Create Program    (Not Done)
6 Run: RCAL_G1      (Not Done)
7 Update Master CT  (Not Done)
8 View Result
```

8. With the cursor on Create Program, press ENTER or F4, SELECT to create the program using positions taught in the Teach Points menu.

If the program already exists, you will see the following message:

```
OK to Delete & Re-generate program?
 Press F2 for YES
 Press F3 for NO
```

- **To confirm, and overwrite the existing teach pendant program**, press F2, YES.
- **If you do not want to delete the existing program,** , press F3, NO.

If the program does not exist, you will see the following message:

```
OK to create a new TP program:
 Press F2 for YES
 Press F3 for NO
```

To create the teach pendant program, press F2, YES. To abort, press F3, NO,

The Create Program status on the main menu will change to Done and the cursor will move automatically to Run.

```
iRCalibration Master Set
1 Robot: Group [1]  Device: Sensor
2 Calibration Type: Master Recovery
3 Setup             (Done    )
4 Teach Points      (Done    )
5 Create Program    (Done    )
6 Run: RCAL_G1      (Not Done)
7 Update Master CT  (Not Done)
8 View Result
```

15.2.4.2 Custom Program Creation

The custom program menu will generate programs supporting only the first, vertically oriented, calibration plate. The custom program creation menu does not support generating programs using two calibration plates.

Guidelines for Creating a Custom Robot Calibration Program

Review the following guidelines before you begin to teach points.

- If the starting TCP is all zero, an estimated TCP will be created and applied when the calibration program is created. The seven taught points are also modified to reflect the new estimated TCP. The user must take care that the rotations during program motion do not cause collisions to occur and must insert additional points if needed.
- If the starting TCP is not all zero, then no estimated TCP is used and all program rotations will occur with respect to the starting TCP. The user must take care that the rotations during program motion do not cause collisions to occur and must insert additional points if needed.
- The calibration of Axis 1 and Axis 6 require that both the UFrame and UTool be known. If fixturing with known dimensions is used then the user can set UFrame and UTool to be the correct values and disable solving for UFrame and UTool and enable calibration of J1 and J6.
- If the robot cannot reach a BACK configuration, then the calibration tool dimensions and the location of the touch plate are important. For this case, the calibration tool must be large enough and the touch plate located such that all the axes have significant range of motion (greater than 90 degrees) during the calibration.
- Careful teaching of the calibration points such that they are centered above the touch plate holes and the with the tool shaft oriented perpendicular to the touch plate surface will improve calibration.
- If the initial robot calibration correction is large then repeating the calibration process a second time (after touching up the calibration points) will improve initial calibration.

There are seven taught points. They are described in Table 15–6 .

Refer to Figure 15–5 and Figure 15–6 for illustrations of the first six taught points.

Table 15–6. Robot Calibration Program Taught Points

Taught Point	Description
Approach Position 1	This point must be located such that the robot can move to both Center Position 1 and Approach position 2 without collision. Also, the wrist must be able to rotate freely at this location without collision.

Table 15–6. Robot Calibration Program Taught Points (Cont'd)

Taught Point	Description
Center Position 1	This point must be located on the side of the touch plate that is consistent with the "Plate Direction" setup value. This position should be rotated toward one extreme of orientation about the plate axis. This position should be taught centered over the touch plate circle and about one inch above the plate.
Rotate Position 1	This point must be located on the same side of the touch plate as Center Position 1. This position should be rotated toward the other extreme of orientation about the plate axis, but the total rotation must be less than 180 degrees. This position should be taught centered over the touch plate circle and about one inch above the plate.
Approach Position 2	This point must be located such that the robot can move to both Center Position 2 and Approach position 1 without collision. Also, the wrist must be able to rotate freely at this location without collision.
Center Position 2	This point must be taught on the opposite side of the plate as Center Position 1. This position should be rotated toward one extreme of orientation about the plate axis. This position should be taught centered over the touch plate circle and about one inch above the plate.
Rotate Position 2	This point must be located on the same side of the touch plate as Center Position 2. This position should be rotated toward the other extreme of orientation about the plate axis, but the total rotation must be less than 180 degrees. This position should be taught centered over the touch plate circle and about one inch above the plate.
Orientation position	This point must be taught with the same orientation as Rotate Position 2 but with the tool shaft extending through the touch plate hole. This position defines the orientation of the tool and +Z will extend out from the tool (such as used by waterjet cutting) if the system variable$CB_VARS.$Z_OUT = TRUE, otherwise +Z will go through the tool (such as used by arc welding) if $CB_VARS.$Z_OUT = FALSE. If the Orientation position is the same as Rotate Position 2 or does not extend through the plate by at least 20mm, then the plate vector is used for determining TOOL Z.

Figure 15–2. Example of First Three Positions

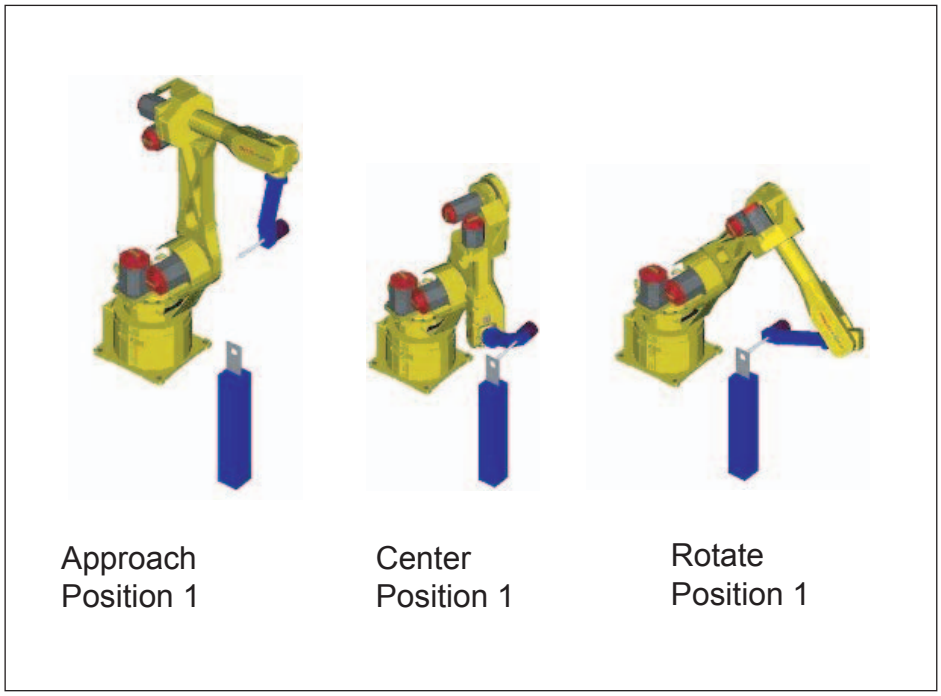
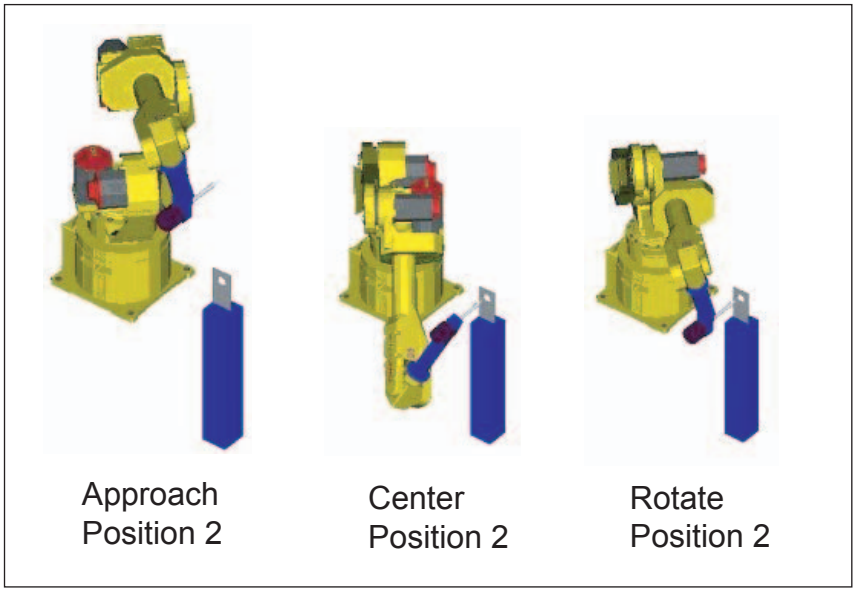


Figure 15–3. Example of Second Three Positions



Note These examples show an M-16 series robot which is capable of the FRONT/BACK and FLIP/NOFLIP configuration change. When the robot can achieve these configuration changes there is a great deal of latitude in the placement of the calibration plate.

The placement shown is not acceptable for robots such as the M-6 series which do not support the FRONT/BACK configuration change. For these robots, we recommend using the Standard Program Creation method using the calibration plate location and UTool that is generated in UFrame[9] and UTool[10] when the *i*RCalibration Master Set program is created. It is recommended that the tooling hardware be constructed to support calibration at these locations.

Six additional points are created by *i*RCalibration Master Set:

- Two additional points are created between the approach position and orientation position on each side of the plate, for a total of four additional orientation points.
- One additional point is created where the wrist changes configuration (Flip to Noflip, or vice versa). The user must ensure that no collisions occur during the change of Flip/Noflip configuration and must insert additional points if needed.
- One additional point is created where the robot changes Front to Back configuration if possible. If the robot cannot achieve this configuration, then an additional Front configuration point is created. The user must take care that no collisions occur during the change of Front/Back configuration and must insert additional points if needed.

Procedure 15-3 Creating a Custom *i*RCalibration Master Set Program

Conditions

- All required calibration hardware has been installed. Refer to Section 15.2.2 .
- The Setup status is Done.

Steps

1. Press MENUS.
2. Select UTILITIES.
3. Press F1, [TYPE].
4. Select iRCalibration. You will see a screen similar to the following:

```
UTILITIES iRCalibration
       1 TCP Shift
       2 TCP Set
       3 Frame Shift
       4 Frame Set
       5 Master Recovery
       6 Master Set
```

5. Select iRCalibration Master Set and press ENTER. You will see a screen similar to the following:

```
iRCalibration Master Set
1 Robot: Group [1]  Device: Sensor
2 Calibration Type: Master Recovery
3 Setup            (Done    )
4 Teach Points     (Not Done)
5 Create Program   (Not Done)
6 Run: RCAL_G1     (Not Done)
7 Update Master CT (Not Done)
8 View Result
```

6. Move the cursor to Teach Points and press ENTER. You will see a screen similar to the following:

```
Program Creation:
  Program Name:     RCAL_G1
  Robot Group:      1
  UTool[10]:        User Set
    X: -115.000 Y:    0.000 Z:  570.000
    W:    0.000 P:  -70.000 R:    0.000
  UFrame[9]:        User Set
    X:  176.000 Y:  600.000 Z: -335.000

  NOTE:  Calibration requires 1 plate

          CREATE          CUSTOM
```

7. Press F4, CUSTOM. You will see a screen similar to the following:

```
Program Creation

  Program Name:      RCAL_G1
  Robot Group:       1
  Calibration Sched: 1
  Detection Sched:   1
  Plate Direction:   –Y
  Plate Thickness:   3.000   mm
  Circle XY offset:  50.000  mm
  Circle Z offset:   10.000  mm
  Robot Calibration
    J1:FALSE    J2:TRUE     J3:TRUE
    J4:TRUE     J5:TRUE     J6:FALSE
    J7:FALSE    J8:FALSE    J9:FALSE
  Approach Position 1: UNINIT
  Center Position 1:   UNINIT
  Rotate Position 1:   UNINIT
  Approach Position 2: UNINIT
  Center Position 2:   UNINIT
  Rotate Position 2:   UNINIT
  Orientation Position:UNINIT
```

8. Select the plate direction. If the UFRAME number is not 0, this direction is with respect to the current user frame.

9. Enter the thickness of the touch plate.

10. Enter the circle xy offset value based on the dimension of the circle in the touch plate.

11. Enter the circle z offset value based on how deep the tool should go into the hole.

12. Select each axis you want to calibrate and set it to TRUE.

13. To clear all recorded positions, press F1, CLEAR.

14. To load the RCAL_G1 positions to the menu's position data if the teach pendant program already exists, press F2, LOAD.

15. Select each joint you want to calibrate and set it to TRUE.

16. Teach three points on one side of the touch plate: the approach position 1, center position 1, and rotate position 1, as shown in Figure 15–5 .

17. Teach four points on the other side of the touch plate: the approach position 2, the center position 2, the rotate position 2, and the orientation position (not shown in Figure 15–6). The orientation position is taught with the tool shaft extended through the circle.

18. When all positions are recorded, press PREV to return to the main menu. If all required positions are taught, the Teach Points status will be Done and the cursor will move automatically to the next step. Otherwise, the cursor will stay at Teach Points and its status will remain unchanged.

```
iRCalibration Master Set
1 Robot: Group [1]  Device: Sensor
2 Calibration Type: Master Recovery
3 Setup             (Done    )
4 Teach Points      (Done    )
5 Create Program    (Not Done)
6 Run: RCAL_G1      (Not Done)
7 Update Master CT  (Not Done)
8 View Result
```

19. With the cursor on Create Program, press ENTER or F4, SELECT to create the program using positions taught in the Teach Points menu.

If the program already exists, you will see the following message:

```
OK to Delete & Re-generate program?
 Press F2 for YES
 Press F3 for NO
```

- **To confirm, and overwrite the existing teach pendant program**, press F2, YES.
- **If you do not want to delete the existing program,** , press F3, NO.

If the program does not exist, you will see the following message:

```
OK to create a new TP program:
 Press F2 for YES
 Press F3 for NO
```

To create the teach pendant program, press F2, YES. To abort, press F3, NO,

The Create Program status on the main menu will change to Done and the cursor will move automatically to Run.

```
iRCalibration Master Set
1 Robot: Group [1]  Device: Sensor
2 Calibration Type: Master Recovery
3 Setup             (Done    )
4 Teach Points      (Done    )
5 Create Program    (Done    )
6 Run: RCAL_G1      (Not Done)
7 Update Master CT  (Not Done)
8 View Result
```

The following screen shows an example of a custom program:

```
1:  UFRAME_NUM=1
   2:  UTOOL_NUM=1
   3:  Calib Start[1]
   4:J P[1:Approach pos. 1] 10% FINE
   5:L P[2:Center 1 outside] 30deg/sec FINE
   6:  Detect Circle[4]( 30.00, 10.00)World[Y]
   7:L P[3] 30deg/sec FINE
   8:  Detect Circle[4]( 30.00, 10.00)World[Y]
   9:L P[4] 30deg/sec FINE
  10:  Detect Circle[4]( 30.00, 10.00)World[Y]
  11:L P[5:Rotate 1 outside] 30deg/sec FINE
  12:  Detect Circle[4]( 30.00, 10.00)World[Y]
  13:L P[6:Center 1 outside] 30deg/sec FINE
  14:L P[7:Approach pos. 1] 100mm/sec FINE
  15:J P[8:Approach pos. 2] 10% FINE
  16:L P[9:Center 2 outside] 100mm/sec FINE
  17:L P[10] 30deg/sec FINE
  18:  Detect Circle[4]( 30.00, 10.00)World[-Y]
  19:L P[11] 30deg/sec FINE
  20:  Detect Circle[4]( 30.00, 10.00)World[-Y]
  21:L P[12] 30deg/sec FINE
  22:  Detect Circle[4]( 30.00, 10.00)World[-Y]
  23:L P[13:Rotate 2 outside] 30deg/sec FINE
  24:  Detect Circle[4]( 30.00, 10.00)World[-Y]
  25:L P[14:Orient inside] 100mm/sec FINE
  26:  Detect Circle[4](  0.00,  0.00)World[-Y]
  27:L P[15:Rotate 2 outside] 100mm/sec FINE
  28:L P[16:Center 2 outside] 30deg/sec FINE
  29:L P[17:Approach pos. 2] 100mm/sec FINE
  30:J P[18:Approach 2 back] 10% FINE
  31:L P[19] 100mm/sec FINE
  32:  Detect Circle[4]( 30.00, 10.00)World[-Y]
  33:L P[20] 100mm/sec FINE
  34:L P[21:Approach 2 back] 100mm/sec FINE
  35:J P[22:Approach 1 flip] 10% FINE
  36:L P[23:Center 1 flip] 100mm/sec FINE
  37:  Detect Circle[4]( 30.00, 10.00)World[Y]
  38:L P[24:Center 1 flip] 100mm/sec FINE
  39:L P[25:Approach 1 flip] 100mm/sec FINE
  40:J P[26:Approach pos. 1] 10% FINE
  41:  Calib End [Calibration]
```

15.2.5 Running the *i*RCalibration Master Set Program

When all the points are taught and the program has been created, the Create Program status will be Done and the cursor will move automatically to Run.

- ***i*RCalibration Master Set** provides accurate initial calibration of the robot axes before any points have been taught. After you run the calibration program you should update the robot's mastering data by using the Update Master CT menu.

It is recommended to run *i*RCalibration Master Set at least twice to get optimal results. Especially if there were significant orientation correction to the taught points, the second execution of *i*RCalibration Master Set can result in a better calibration and lower calibration errors.

Procedure 15-4 Running the *i*RCalibration Master Set Program

Conditions

- The Create Program status is Done.

Steps

1. Press MENUS.
2. Select UTILITIES.
3. Press F1, [TYPE].
4. Select iRCalibration. You will see a screen similar to the following:

```
UTILITIES iRCalibration
       1 TCP Shift
       2 TCP Set
       3 Frame Shift
       4 Frame Set
       5 Master Recovery
       6 Master Set
```

5. Select iRCalibration Master Set and press ENTER. You will see a screen similar to the following:

```
iRCalibration Master Set
1 Robot: Group [1]  Device: Sensor
2 Calibration Type: Master Recovery
3 Setup            (Done    )
4 Teach Points     (Done    )
5 Create Program   (Done    )
6 Run: RCAL_G1     (Not Done)
7 Update Master CT (Not Done)
8 View Result
```

6. Move the cursor to Run and press ENTER. You will see a screen similar to the following:

```
OK to run calibration program?
 Press F2 for YES
 Press F3 for NO
```

7. To run the program, press F2, YES. You will be asked to confirm your selection. You will see the following message:

```
Execute Calibration Program:
 Press F2 for YES
 Press F3 for NO
```

8. Be sure that the teach pendant is ON and all faults have been cleared. Using a low override, press SHIFT+F2 to run the program for the first time to make sure all positions are reachable without colliding with obstacles. After the program runs through without collision, run the program again at 100% override to calibration the robot.

The Run status will remain “Not Done” until the teach pendant program has run successfully.

```
iRCalibration Master Set
1 Robot: Group [1]  Device: Sensor
2 Calibration Type: Master Recovery
3 Setup            (Done    )
4 Teach Points     (Done    )
5 Create Program   (Done    )
6 Run: RCAL_G1     (Done    )
7 Update Master CT (Not Done)
8 View Result
```

You should always verify the generated calibration program by stepping through it at a low override. Be careful that the robot does not collide with objects, especially for the large rotations generated between the FRONT/BACK and FLIP/NOFLIP configurations. During single-step the detection instructions are not executed, so verifying the program is relatively quick.

You can modify the generated program, but any modifications will be lost if you regenerate the program. Also, certain modifications could prevent *i*RCalibration Master Set from loading the program positions, should you want to regenerate it. It is always a good idea to keep a backup before modifying a generated *i*RCalibration Master Set program.

It is recommended to run *i*RCalibration Master Set at least twice to get optimal results. Especially if there were significant orientation correction to the taught points, the second execution of *i*RCalibration Master Set can result in a better calibration and lower calibration errors.

15.2.6 Updating the Master Count

The Update Master Count menu is not accessible unless you have completed all the steps before it.

Procedure 15-5

Conditions

- You have run the *i*RCalibration Master Set Teach Pendant program.

Steps

1. Press MENUS.
2. Select UTILITIES.
3. Press F1, [TYPE].
4. Select iRCalibration. You will see a screen similar to the following:

```
UTILITIES iRCalibration
       1 TCP Shift
       2 TCP Set
       3 Frame Shift
       4 Frame Set
       5 Master Recovery
       6 Master Set
```

5. Select iRCalibration Master Set and press ENTER. You will see a screen similar to the following:

```
iRCalibration Master Set
1 Robot: Group [1]  Device: Sensor
2 Calibration Type: New Calibration
3 Setup             (Done    )
4 Teach Points      (Done    )
5 Create Program    (Done    )
6 Run: RCAL_G1      (Done    )
7 Update Master CT (Not Done)
8 View Result
```

6. Move the cursor to Update Master CT and press ENTER. You will see a screen similar to the following.

```
Calibration Result

    J1:   0.000 J2:   0.000 J3:  0.000
    J4:  −0.005 J5:  −0.004 J6:  0.000
    J7:   0.000 J8:   0.000 J9:  0.000
  TCP[10]  Uupdate:[TRUE  ]
    X: −130.244 Y:   −0.430 Z: 529.121
    W:    0.057 P:  −45.047 R:  −0.017
  UFrame[ 9] Update:[TRUE  ]
    X:  621.159 Y:  327.239 Z:-510.843
    W:    0.000 P:    0.000 R:   0.000
Mean Error:      0.146
Max. Error:      0.275
```

7. Press F2, UPDATE, to update the master count.
8. Press PREV to return to the *i*RCalibration Master Set main menu.

15.2.7 Viewing the Calibration Result

The *i*RCalibration Master Set View menu allows you to see the results of previous calibrations. You can access the view menu at any time to review what has been done with *i*RCalibration Master Set earlier, even after a cold start. The screen you see will be based on the calibration type selected in the main menu

The calibration log is a buffer that holds previous mastering information. It has a fixed size of 20 entries. When the buffer is full, it will replace the oldest data in the buffer with the new logged data. If you want to view mastering information, you can display the Calibration Result screen. To display the calibration result screen, use Procedure 15-6 .

Procedure 15-6 Viewing the Calibration Result

Conditions

- You have installed the *i*RCalibration Master Set software option.

Steps

1. Press MENUS.
2. Select UTILITIES.
3. Press F1, [TYPE].
4. Select iRCalibration. You will see a screen similar to the following.

```
UTILITIES iRCalibration
       1 TCP Shift
       2 TCP Set
       3 Frame Shift
       4 Frame Set
       5 Master Recovery
       6 Master Set
```

5. Select iRCalibration Master Set and press ENTER. You will see a screen similar to the following.

```
iRCalibration Master Set
1 Robot: Group [1]  Device: Sensor
2 Calibration Type: Master Recovery
3 Setup            (Done    )
4 Teach Points     (Done    )
5 Create Program   (Done    )
6 Run: RCAL_G1     (Done    )
7 Update Master CT (Done    )
8 View Result
```

6. Move the cursor to View Result and press ENTER.

   You will see a screen similar to the following:

```
Calibration Result

    J1:   0.000 J2:   0.000 J3:  0.000
    J4:  −0.005 J5:  −0.004 J6:  0.000
    J7:   0.000 J8:   0.000 J9:  0.000
  TCP[ 1]  Uupdate:[TRUE  ]
    X: −130.244 Y:   −0.430 Z: 529.121
    W:    0.057 P:  −45.047 R:  −0.017
  UFrame[ 1] Update:[TRUE  ]
    X:  621.159 Y:  327.239 Z:-510.843
    W:    0.000 P:    0.000 R:   0.000
Mean Error:      0.146
Max. Error:      0.275
```

7. To view the Master Log, press F4, VIEW LOG. You will see a screen similar to the following:

```
Master Log

  Date   Time    J1(47)   J2(58)   J3(69)
14-MAY-08 10:59      0  19187  −11293
  Group Num[  1]  2766   767 0
  Extended Axes      0       0    0
14-MAY-08 11:02      0   31495  −23804
  Group Num[  1]  2881   -3197    0
  Extended Axes      0       0    0
No  J1 J2 J3 Data
No  J4 J5 J6 Data
No  J7 J8 J9 Data
```

8. To save the calibration log to a text file in the “PATH:\calb_log.dt”, press F2, SAVE.

To clear the calibration log, press F4, CLEAR.

To recover a previous master count data, press F3, APPLY.

9. Press PREV to return to the View Calibration Result page.

15.2.8 Troubleshooting *i*RCalibration Master Set

Troubleshooting *i*RCalibration Master Set

Refer to Table 15–7 for troubleshooting information.

Table 15–7. Troubleshooting

Problem	Possible Cause	Solution
The robot does not detect the Touch Plate for the I/O method.	I/O was incorrectly set up.	Verify that the I/O is set up properly.
		Verify that contact with the Touch Plate will cause the specified I/O signal to go ON. Change polarity of the I/O point if necessary.
	The Touch Plate or tool is not isolated.	Verify that either the tool is isolated (for the case of a charged tool) or that the Touch Plate is isolated (for the case of a charged plate) from ground, and that the element that is not charged is grounded electrically.
	The tool or Touch Plate surfaces are dirty.	Verify that both the tool and plate surfaces are clean and free from film.
The robot hits obstacles during rotations.	There is not enough space around the Touch Plate.	Touch up the calibration program's position.
The robot does not always touch the inside of the Touch Plate hole with a cylindrical portion of the tool.	The shaft is not far enough through the Touch Plate hole.	Touch up the calibration program's position.
When is it appropriate to use the I/O method and when is it appropriate to use Torque Observer Sensor (TOS) method?	There is delicate or compliant tooling.	Use I/O method if there is delicate or compliant tooling.
	The wrist center is in the same plane as the Touch Plate.	Use I/O method if the wrist center is to close to or in the same plane as the Touch Plate. If the tool is durable, TOS All Axes can also be used.
	The wrist center is directly above the Touch Plate.	Use I/O method if the wrist center is directly above the Touch Plate. If the tool is durable, TOS All Axes can also be used.
	There is a durable tool and a small robot.	The TOS All Axes method is recommended for small robots when the I/O method is not used.
	There is a durable tool and a large robot.	Use the TOS All Axes method if there is a durable tool.
Large Calibration Error	Incorrect UFrame or UTool setting	Make sure that the solve UFrame and solve UTool flags are TRUE unless the UFrame exactly corresponds to the calibration plate location and unless the UTool exactly corresponds to the calibration tool.
	TOS false detections	Rerun the calibration and watch to confirm the tool touches the calibration plate for every detection. If it does not touch in one or more detections, increase the TOS Tolerance or increase the learning speed and learning distance or reteach the point with a slightly different rotation.

15.3 *i*RCALIBRATION MASTER RECOVERY

15.3.1 Overview

Note *i*RCalibration Mastering is not available for all robots. Refer to Software Release Note for specifics.

*i*RCalibration Master Recovery is an optional calibration option that allows you to recover the mastering of one of more robot axes after a motor change or a crash.

*i*RCalibration Master Recovery uses a wizard-style menu. It shows the steps that need be done to complete the robot calibration process. Each step of the process has its own status display that shows where you are in the setup process. You cannot skip steps and you cannot go to the next step without completing the current step. *i*RCalibration Master Recovery also includes a menu item, View Result, that lets you review the previous calibration result and log history.

- *i*RCalibration Master Recovery uses teach pendant programs for all execution. These programs are typically generated by *i*RCalibration Master Recovery, but can also be created or modified by the user.
- *i*RCalibration Master Recovery supports schedule based sensor setup and can use a different schedule for different searches.
- The *i*RCalibration Master Recovery reference data is stored in the teach pendant program; system variables are not used to store reference data.
- *i*RCalibration Master Recovery supports all robot motion groups.
- *i*RCalibration Master Recovery supports single-axis detection and recovery

The same program used for *i*RCalibration Master Set initial setup can later be used to recover mastering for any or all robot axes, J1 through J6. If *i*RCalibration Master Set is not performed, but *i*RCalibration Master Set mastering is performed, it is recommended to both master and recover at least J2, J3, J4, and J5. It is not recommended to recover a different set of axes than were mastered, except it is OK to recover J1 and J6 as described in Table 15–9 . Also, see the special requirements regarding J2 for robots with linear J1 axis.

Hardware Requirements

The following hardware is required to use *i*RCalibration Master Recovery:

- Calibration tooling. Refer to Section 15.3.2 .
- Six-axis robot (except as indicated in the Software Release Note)
- User-supplied circuitry for using I/O sensor (for robots with delicate tooling)

If the standard tooling is used, *i*RCalibration Master Recovery can generate a program to perform mastering recovery of robot axes 2 through 5, usually without any user point teaching. Standard

tooling is available for most robots to support the default calibration program generated by *i*RCalibration Mastering.

Using *i*RCalibration Master Recovery

To use *i*RCalibration Master Recovery, you must do the following:

1. Install the *i*RCalibration Master Recovery software option and any required hardware.
2. Set up *i*RCalibration Master Recovery.
3. Create a robot calibration program.
4. Run *i*RCalibration Master Recovery.
5. Run Mastering Recovery as needed.

15.3.2 Preliminary Setup

Standard *i*RCalibration Mastering Hardware

Standard *i*RCalibration Mastering hardware is available. It includes the following components:

- Calibration Tool to attach to the faceplate of the robot. An adapter plate may be required.
- Calibration Arm with one or two calibration plates to attach to the base of the robot
- I/O cable to attach to the robot EE connector

The standard hardware available includes the following:

- EE connector cable to support the I/O detection method of the Calibration Options. Cables are available to support both 5-pin and 24-pin EE connectors.
  - — EE-5770-020-040 (4 meters).
    Cable Assembly for End Effector Touch Sensor w / 24 Pin Connector
  - — EE-5770-021-025 (2.5 meters).
    Cable Assembly for End Effector Touch Sensor w / 5 Pin Connector
- Calibration tool assembly to mount to the robot faceplate. The calibration tool assembly consists of a base to mount to the faceplate, a supporting tube extending away from the base, an attached nonconducting block, and a calibration pointer supported by the non-conducting block. The angle between the J6 axis and the calibration pointer is 70 degrees for the short tool and 45 degrees for the long tool. Multiple tools and adapters are available to support most robots.
  1. MO-1800-943 Calibration Tool Assembly – Short
  2. MO-1800-944 Calibration Tool Assembly – Long
  3. MO-1800-945-001 Calib. Tool Adapt. - M710/20L

4. MO-1800-945-002 Calib. Tool Adapt. - R2000
5. MO-1800-945-003 Calib. Tool Adapt. - M900
6. MO-1800-945-004 Calib. Tool Adapt. - Arc Solution Arm
7. MO-1800-945-005 Calib. Tool Adapt. - Arc Tregasis
8. MO-1800-945-006 Calib. Tool Adapt. - Arc Binzel

- Calibration arm to mount to the robot base. The calibration arm assembly includes two calibration plates and a mounting bracket. *i*RCalibration Mastering, depending on robot model and configuration, uses one or both of the plates. Multiple calibration arms are available to support most robots.
  - — MO-1800-940 Calibration Arm Assembly –Standard
  - — MO-1800-941 Calibration Arm Assembly – Short

Note The standard calibration arm, MO-1800-940, can be used on any robot which is supported by the short calibration arm, MO-1800-941.

Table 15–8. Calibration Hardware

Robot Series	Calibration Arm	Calibration Tool	Adapter(s)
M6 (AM100iB)	MO-1800-941	MO-1800-943	
AM100iB - SA			MO-1800-945-004
M6-6S (AM100iB/6S)			
M16 (AM120iB)			
AM120iB - SA			MO-1800-945-004
M16/10L (AM120iB/10L)			
AM120iB/10L - SA			MO-1800-945-004
M710iB			MO-1800-945-002
M710iB/20L	MO-1800-940		
M710iB/20L –SA*			
R-2000 (R & P)		MO-1800-944	MO-1800-945-002
R-2000		MO-1800-943	
R-2000 - SA		MO-1800-944	
M-900 (350)			MO-1800-945-003
M-900 (600)			

*The M710iB/20L-SA does not currently have a Standard Calibration Tool assembly available.

Some robots, for example the top loader series of robots, do not have an available standard calibration arm. For these cases *i*RCalibration Master Recovery can still generate the standard *i*RCalibration Mastering program if the user installs the standard calibration tool to the robot faceplate and installs a calibration plate at the proper location for the generated calibration program.

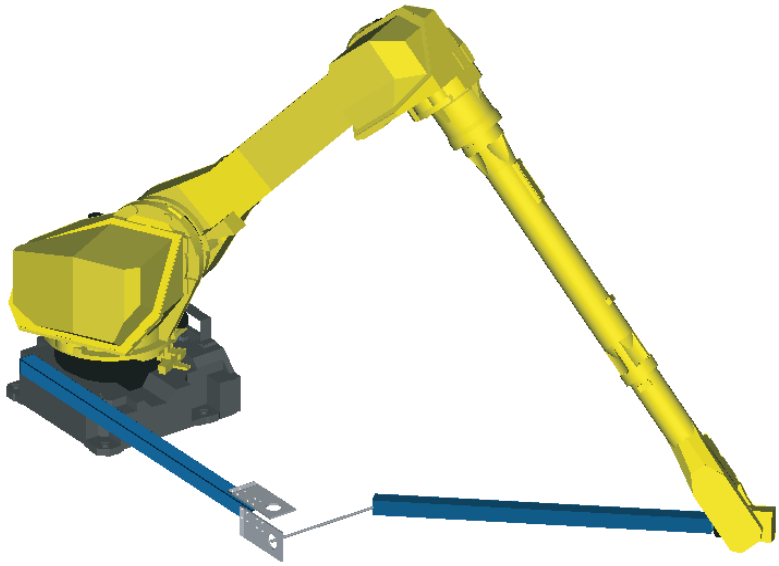
Robots that cannot do FRONT/BACK configuration change or FLIP/NOFLIP configuration change require two calibration plates. **For all robots**, the first calibration plate must be located parallel to the robot World Y-Z plane with the hole centered at the Uframe coordinates.

If the robot requires two plates, the second plate must be located in the X-Y plane with the hole centered at these values: (Uframe.x – 56, Uframe.y-82, Uframe.z+57). The two-plate method of initial calibration is more accurate for robots that cannot change configuration than the single plate method. The two-plate method generates 11 calibration points on the first plate and 6 on the second plate.

The Standard *i*RCalibration Mastering calibration plate(s) are attached (by a calibration arm) to the mastering holes located on front of the robot base. The calibration arm is located along the +Y direction of robot World frame. The calibration plates are not precisely located; *i*RCalibration Mastering will fine tune the actual location of the calibration plates.

Refer to Procedure 15-13 for information on installing and aligning the Touch Plate and for setting up the Touch Plate for use with *i*RCalibration Mastering.

Figure 15–4. Standard *i*RCalibration Mastering Calibration Plates



15.3.3 *i*RCalibration Master Recovery Setup

The first time you run the recovery program, *i*RCalibration Master Recovery *masters* the recovery program by storing *golden data* into the teach pendant program. For mastering, no update is required. After the teach pendant program has been mastered, you can recover mastering data by running the *i*RCalibration Mastering program again and using the Update menu to update the robot's mastering data.

*i*RCalibration Master Recovery can recover mastering data by running the mastered teach pendant program after a motor has been replaced. The *i*RCalibration Master Recovery computes the offset between the new contact positions with the golden data and convert the offset to pulse counts to adjust the zero position for that motor. After the mastering data of that motor has been updated, the robot is re-mastered and no touch up is required for production teach pendant programs.

Setting up and running *i*RCalibration Master Recovery includes the following:

- Setting up the Mastering Schedule. Refer to Table 15–9 for descriptions of the items on the Mastering Schedule screen.
- Setting up the Detection Schedule. Refer to Table 15–10 for descriptions of the items on the Detection Schedule screen.
- Creating the robot calibration program. Refer to Section 15.3.4 .
- Running the calibration program. Refer to Section 15.3.5 .

- Updating the master count (Master CT) with the calibration result obtained from running the robot calibration program. Refer to Section 15.3.7 .

Note There are 10 Mastering Schedules.

Table 15–9. *i*RCalibration Mastering Schedule Screen Description

ITEM	DESCRIPTION
Robot J1 - J9	The calibration for each axis can be enabled or disabled individually. The default values are good for normal operation where initial mastering is performed and for mastering *i*RCalibration Mastering from the recovery menu. For robots with all rotary joints, it is recommended that J2, J3, J4, and J5 all be TRUE. As described in the Calibrate TCP and Calibrate UFrame sections, J1 (and J2 for linear J1 robots), and J6 have special considerations. The normal case for mastering recovery is to use the same settings as for initial mastering. If you need to recover J1 and/or J6 (and/or J2 for linear J1 robots) then you must follow the instructions regarding TCP and UFrame.
Calibration Group Range 1 - 8 Default: 1	This item defines the robot group to be calibrated.
Calibrate TCP True/False Default: True	This item allows calibration software to solve for the TCP value. Normally Calibrate TCP is TRUE unless you are using a previously accurately calibrated TCP. If Calibrate TCP is TRUE, then the calibration field corresponding to J6 must be false. You can never calibrate or recover both the TCP and J6. For mastering recovery, if UTool has been maintained and is still valid at the time of mastering recovery, then Calibrate UTool can be FALSE and J6 can be TRUE.

Table 15–9. *i*RCalibration Mastering Schedule Screen Description (Cont'd)

ITEM	DESCRIPTION
Calibrate UFRAME True/False Default: True	This item allows calibration software to solve for the user frame. Note that this will make UFRAME be the center of the circle in the calibration plate and will only set uframe XYZ value. Normally, Calibrate UFrame is TRUE unless you are using a previously accurately calibrated calibration plate. If Calibrate UFrame is TRUE, then the calibration field corresponding to J1 must be false. For TopLoaders or other robots with a linear J1 axis then J2 must be false whenever Calibrate UFrame is TRUE You can never calibrate or recover both UFrame and J1. Additionally for robots with a linear J1 axis you can never calibrate or recover both UFrame and J2. Any J1 (and J2 for linear J1 robots) calibration error can be fully corrected by UFrame. If you use CellCal to maintain UFrame following *i*RCalibration Mastering mastering recovery then it is not necessary to recover J1 (and J2 for linear J1 robots). However, if programming causes these axis to move close to their limits then a run-time limit error may occur if these axes are not recovered. If you want to recover J1 (and J2 ror linear J1 robots) then you MUST mount the calibration plate in a fixed location or in a location that can be exactly duplicated at the time of recovery AND the exact UFrame found during initial mastering must be used at the time of recovery. Normally this is done by dedicating a specific UFrame number for mastering and mastering recovery.
Allowable mean error Units: mm Range: 0.01 - 100 Default:	This item is the maximum allowable mean error for the calibration computation.
Mean Error Register Range: 0 - 999 Default: 0	If the number is greater than 0, the calibration software will write the calibration mean error to the register.

Table 15–9. *i*RCalibration Mastering Schedule Screen Description (Cont'd)

ITEM	DESCRIPTION
Max. Error Register Range: 0 - 999 Default: 0	If this number is greater than 0, the calibration software will update the calibration maximum error to the register.
Use Current TCP True/False Default : FALSE	This item is used for calibration recovery. • **If this value is TRUE, and Calibrate TCP is FALSE**, then the current active UTool will be used for determining the calibration parameters. • **If this value is FALSE and Calibrate TCP is FALSE**, then the current UTool must be the same as was used during calibration mastering. • **If this value is TRUE, and the physical tool has changed by maintenance or other reasons** since *i*RCalibration Mastering was mastered, then TCPCal or some other method should be used to maintain the correspondence between the UTool value and the physical tool.

Detection Schedules

Note The 10 detection schedules are shared among calibration options *i*RCalibration TCP Set, *i*RCalibration Frame Set, and *i*RCalibration Mastering. Therefore, a change in detection schedules affects all the *i*RCalibration options you have installed on your controller.

When you set up a detection schedule, you

- determine the sensor
- determine the detection motion
- determine TOS

Note Default values are usually acceptable.

Table 15–10. *i*RCalibration Mastering Detection Schedule Description

ITEM	DESCRIPTION
Sensor Type Values: TOS Wrist, I/O, TOS ALL AXES Default: TOS ALL AXES	This item indicates the kind of sensor used for contact detection. The four modes are: • **TOS Wrist** (Torque Observer Sensor on Wrist) - the contact torque will be monitored, and only the observed torque on the wrist will be used to determine the contact position. When TOS Wrist is selected, no I/O connection is required for contact detection. TOS Wrist is very sensitive and can be more prone to false detections in certain cases than the more robust TOS All Axes method. TOS Wrist is not recommended except for special cases where it can verified that false detections will not occur. • **I/O** - the specified I/O will be monitored and only the state of this input will be used to determine the contact position. Use the detection port for contact detection with I/O. Refer to Detection Port item in this table for more information. • **TOS ALL AXES** (Torque Observer Sensor, All Axes) - The contact torque will be monitored on all robot axes, and only the observed torque will be used to determine the contact position. This is the default value. When TOS ALL AXES is selected, no I/O connection is required for contact detection.
Detection Port Values: RDI, DI, WDI, WSI	This item indicates the desired input port type. The port types are: • **RDI** - Robot Digital Input • **DI** - normal Digital Input • **WDI** - Weld Digital Input • **WSI** - Wire Stick circuit Input. The port type that you specify is followed by the desired port number that will go from OFF to ON when contact is detected. The WDI and WSI ports are typically used for arc welding. You will set this item up if you selected I/O as the Sensor Type. You do not need to set this up if you selected TOS Wrist or TOS All Axes as the Sensor Type.

Table 15–10. *i*RCalibration Mastering Detection Schedule Description (Cont'd)

ITEM	DESCRIPTION
Sensor Enable Values: RDO, DOUT, WDO, or WSO Default: RDO	This item indicates the I/O sensor enable digital output port type. The port types are: • **RDO** - Robot Digital Output • **DOUT** - normal Digital Output • **WDO** - Weld Digital Output • **WSO** - Wire Stick circuit Output. The port type that you specify is followed by the desired port number that will be used to enable the I/O detection circuit. The WDO and WSO ports are typically used for arc welding. You will set this item up if you selected I/O as the Sensor Type. Set the port number to 0 (zero) if you are not using this feature.
Detection Speed Units: mm/sec Range: 1.0 - 100 Default: 10	This item is the search speed for contact detection. This speed is used for both TOS and I/O searches. The value can be increased if cycle time is of great concern (although increasing the speed will also affect the resultant accuracy and repeatability) , or reduced if vibration or other disturbances cause *i*RCalibration Mastering to retry excessively.
Detection Distance Units: mm Range: 1 - 1000 Default: 50	This item is the maximum distance that the robot will move while looking for the signal. If no contact is detected within this distance during a search, an error message will be posted.
Return Speed Units: mm/sec Range: 5 - 5000 Default: 50	This item is the search return speed. This speed is used for both TOS and I/O searches.
Return to Start Default: Yes	When this item is set to YES, the robot will move back to the position before it starts the search motion. This item applies to DETECT POINT instructions only. When this item is set to NO, the robot will move back to a distance specified by the next item (Return distance)
Return Distance Units: mm Range: 1 - 1000 Default: 50	When return to start is set to FALSE, the robot will move back to the distance specified.
Return term type Default: FINE	This item is the termination type for return motion.
Air Cut Speed Units: mm/sec Range: 1 - 1000 Default: 100	This item is the non-search motion speed.

Table 15–10. *i*RCalibration Mastering Detection Schedule Description (Cont'd)

ITEM	DESCRIPTION
Learn Distance Units: mm Range: 0.5– 20.0 Default: 5.0	For "TOS All Axes" method only, there is a short motion before each search to identify dynamic torque values. Learn Distance is the distance in millimeters of this short motion, the direction is determined at the time of the motion.
Learn Speed Units: mm/sec Range: 0.5 – 50.0 Default: 10.0	For "TOS All Axes" method only, this is the speed of the learning motion before each search.
Sensitivity Tolerance Units: max. motor torque/sec. Range: 1 - 999 Default: 10	This item is the primary sensitivity tolerance of TOS Wrist and is not used for TOS All-Axes. It is the rate of change of the disturbance torque. The actual units vary with motor model, gear ratio, and amplifier current, and are affected by tooling length. However, the effect for calibration is largely independent of these factors and a value of 10 is considered normal operation for typical installations. You should not use a value of less than 10 or misdetections might occur. This value can be increased if false detections occur frequently. **Warning** **Be careful when you change this number. Changing the sensitivity tolerance might cause a false detection. This could injure personnel or damage equipment.**
Torque Threshold Units: max. motor torque Range: 1 to 999 Default: 100	For TOS All Axes, this item is the only measure of contact detection. During the learning motion a torque level is memorized and that level plus Torque Threshold must be exceeded for contact to be determined. If false detections are being experienced then this value should be increased. For TOS Wrist, this item is the level of disturbance torque that is used as another means for contact detection. The default value of 100 is considered normal operation for typical installations. If increasing the value of **Sensitivity Tolerance** does not make false detections disappear, then this value should be increased until the false detections disappear; then, reset the value of **Sensitivity Tolerance** .
Fit Tolerance Units: mm Range: 0.1 - 100 Default: 0.5	For Detect Circle instruction: The robot detects 4 points inside the circle and the calibration software will fit a circle to the 4 detected positions. The fit error is defined as the maximum distance from the 4 points to the circle. If the fitting error exceeds the tolerance, the calibration software will post an error.
Radius Tolerance Units: mm Range: 0.1 — 100 Default: 0.5	The Detect Circle instruction stores the circle's radius during mastering. At run time, if the computed circle radius differs from the stored radius value and difference exceeds the tolerance, the calibration software will post an error.

Table 15–10. *i*RCalibration Mastering Detection Schedule Description (Cont'd)

ITEM	DESCRIPTION
Number of Retry Range: 0 - 9 Default:	If the number of retry is >0, the calibration software will re-run the detect circle instruction if the current circle detection has either a fit error or a radius error.
Contact Position Register	If the position register number is any number other than 0, *i*RCalibration Mastering will copy the contact position to the position register.

15.3.3.1 Setting Up *i*RCalibration Master Recovery

*i*RCalibration Master Recovery uses a wizard-style menu. It shows the steps that need be done to complete the robot master recovery process. Each step of the process has its own status display that shows where you are in the setup process. You cannot skip steps and you cannot go to the next step without completing the current step.

*i*RCalibration Master Recovery also includes a menu item, View Result, that lets you review the previous calibration result and log history.

For a robot that has all its program taught, *i*RCalibration Master Recovery allows you to recover the robot's mastering data when necessary. You run the recovery program initially to "master" the data in the program. No update is necessary at this time. Then, when you need to recover the mastering data, you run the program again and use the Update menu to update the mastering data .

Procedure 15-7 Setting Up *i*RCalibration Mastering

Conditions

- The *i*RCalibration Master Recovery software option is loaded.
- Preliminary setup is complete. Refer to Section 15.3.2 .

Steps

1. Press MENUS.
2. Select UTILITIES.
3. Press F1, [TYPE].
4. Select iRCalibration. You will see a screen similar to the following:

```
UTILITIES iRCalibration
       1 TCP Shift
       2 TCP Set
       3 Frame Shift
       4 Frame Set
       5 Master Recovery
       6 Master Set
```

5. Select iRCalibration Master Recovery and press ENTER. You will see a screen similar to the following:

```
 iRCalibration Master Recovery
1 Robot: Group [1]  Device: Sensor
  Calibration Type: Master Recovery
2 Setup            (Not Done)
3 Teach Points     (Not Done)
4 Create Program   (Not Done)
5 Run: RCAL_G1     (Not Done)
6 Update Master CT (Not Done)
7 View Result
```

6. Move the cursor to Robot: Group [1], type the group number of the motion group you want to calibrate, and press ENTER.

Note Device is always Sensor and the calibration type is always Master recovery. You cannot change them.

7. **Specify the Calibration Program Name**. The *i*RCalibration Master Recovery Setup menu allows you to specify the robot calibration teach pendant program name, select and set the *i*RCalibration Mastering schedule and detection schedule.

Note The Group number is provided for reference. You cannot change it in this menu.

a. Move the cursor to Setup and press ENTER. You will see a screen similar to the following:

```
Mastering Setup
      1 Program name: RCAL_G1      Group[1]
      2 Mastering Schedule:  1
      3 Detect Schedule:    1
```

b. The default program name is RCAL_Gx, where x is the group number. If you want to specify a different program name, press ENTER and use the function keys.

8. **Select the Mastering Schedule and set schedule detail items.** Move the cursor to Mastering Schedule. Refer to Table 15–9 for information on the *i*RCalibration Mastering Schedule items.

a. To change the number of the Mastering Schedule, move the cursor to the schedule number, type the number of the schedule you want, and press ENTER.

b. To see the detail screen for the selected schedule, press F4, SELECT, or press ENTER. You will see a screen similar to the following:

```
 Mastering Schedule
                                  1/10
  Schedule Number [ 1]
  Robot
    J1:FALSE   J2:TRUE   J3:TRUE
    J4:TRUE    J5:TRUE   J6:FALSE
    J7:FALSE   J8:FALSE  J9:FALSE
Calibration Group:     1
Calibrate TCP:         TRUE
Calibrate UFrame:      TRUE
Allowed Mean Error:    30.00  mm
Mean Error Register    0
Max Error Register    0
Use Current TCP:        TRUE
```

To calibrate a specific joint angle of the robot, select the axis and set it to TRUE:

- Move the cursor to select each item and set it as desired.
- Press F4, CHOICE, to choose from more than one setting for a particular item.

Refer to Table 15–9 for information on the *i*RCalibration Mastering Schedule items.

When you have finished setting up the *i*RCalibration Mastering schedule, press PREV to return to the *i*RCalibration Mastering Setup Menu:

```
Mastering Setup
       1 Program name: RCAL_G1      Group[1]
       2 Mastering Schedule:  1
       3 Detect Schedule:    1
```

9. Select the Detection Schedule and Set Detection Schedule detail items. Move the cursor to Detect Schedule.

a. To change the number of the Detect Schedule, move the cursor to the schedule number, type the number of the schedule you want, and press ENTER.

b. To set up the detection schedule detail items, press F4, SELECT. You will see a screen similar to the following:

```
Detection Schedule
                                    1/17
  Detection Schedule [ 1]
  Sensor Type:              I/O
  I/O Setup
    Detection Port:         DI   [  1]
    Sensor Enable:          DOUT[  1]
  Motion Setup
    Detection Speed:        10.000  mm/sec
    Detection Distance:    100.000  mm
    Return Speed:           50.00   mm/sec
    Return to Start:       TRUE
    Return Distance:         0.000  mm
    Return Term Type:       Fine
    Learn Speed:            10.000  mm/sec
    Learn distance:          5.000  mm
    Air Cut Speed:         100.000  mm
  TOS Setup
    Sensitivity:            100.000
    Torque Threshold:       100
  Circle Setup
    Fit Tolerance:          1.000
    Radius Tolerance:       .500
    Number of Retry:        2
  Contact Position Register:   0
```

Note The default settings are good for most applications.

Set each item as desired. Move the cursor to the item you want to change and type the value.

Refer to Table 15–10 for information about detection schedule setup detail items.

10. When setup is complete, press F1, DONE. You will see the *i*RCalibration Master Recovery main menu with Setup (Done) in its status display.

```
iRCalibration Master Recovery
1 Robot: Group [1]  Device: Sensor
  Calibration Type: Master Recovery
2 Setup             (Done    )
3 Teach Points      (Not Done)
4 Create Program    (Not Done)
5 Run: RCAL_G1      (Not Done)
6 Update Master CT  (Not Done)
7 View Result
```

Note You must press DONE to change the Setup status. If you press PREV to return to the TCPCal main menu, the Setup status display will not change.

If the teach pendant program already exists when you press DONE, you will see the following screen:

```
Program exists. Skip Teach/Create?
  Press F2 for YES
  Press F3 for NO
```

If the teach pendant program already exists and has been taught, and you want to skip the teach/create step, press F2, YES, and proceed directly to Run. Refer to Section 15.3.5 to run the program.

If the TP program exists but you want to teach it, change it, or use another program, press F3, NO, and continue to the Teach Points menu item.

15.3.4 Creating a Robot Calibration Teach Pendant Program

In order to master a robot with *i*RCalibration Master Recovery you must first create a robot mastering teach pendant program. You can create the program manually or you can have *i*RCalibration Mastering create the program for you automatically. When *i*RCalibration Master Recovery setup is complete, the *i*RCalibration Master Recovery main menu will be displayed with the cursor at Teach Points.

For most six-axis robots

- you can have *i*RCalibration Master Recovery create a standard *i*RCalibration Mastering Calibration program automatically,
- you can customize an automatically generated program, or
- you can create your own custom program.

If yours is not a six-axis robot, you can create a custom program, but *i*RCalibration Master Recovery cannot create a standard *i*RCalibration Master Recovery program automatically.

If a robot is not supported for the Standard *i*RCalibration Master Recovery program creation, the menu goes directly to the CUSTOM calibration menu. You can also go to this page by pressing F4, CUSTOM, at the Program Creation menu.

If you cannot use the Standard *i*RCalibration Master Recovery generated program because of robot limits, workspace restrictions, or other reasons, you should use the CUSTOM program creation method. You can touch up generated points from either the Standard or Custom generated programs to meet the specific restrictions of the workcell. However, the final program must contain significant rotations of all axes in order to get the best results.

To disable the Standard *i*RCalibration Master Recovery program creation menu, set the system variable $CB_VARS.$STDCAL_ENB = FALSE. If the Standard *i*RCalibration Mastering program

creation menu is disabled, when you select “Teach Points” from the main menu you will go directly to the custom *i*RCalibration Master Recovery program creation menu.

Teaching Points

Creating a standard *i*RCalibration Master Recovery program automatically can usually be done without teaching any points. If you are creating a custom program to recover mastering with *i*RCalibration Mastering, you must first teach several points to be used in your robot mastering teach pendant program. Refer to Section 15.3.4.2 .

15.3.4.1 Creating a Standard *i*RCalibration Master Recovery Program Automatically

Table 15–11. Program Creation Screen Items for Standard *i*RCalibration Master Recovery Programs

Item	Description
Program Name	The program name for the *i*RCalibration Master Recovery program creation cannot be changed. It is of the form “RCAL_G” followed by the group number of the robot being calibrated. The current active jog group is used in determining which group will be used.
UTool[10]: Uframe[9]:	*i*RCalibration Mastering program creation always uses UTool[10], UFrame[9]. **Program Defined** will appear if the corresponding frame was not previously set; that is, the frame xyzwpr components were all zero. For this case, when you press F2, CREATE, *i*RCalibration Master Recovery will set the frames to the identified values. The Uframe and UTool may be different for different robot models. You cannot change the UTool, Uframe, Detection Schedule, or Calibration Schedule number used by Standard *i*RCalibration Mastering. However, you can edit the generated program to use different Detection or Calibration schedules. You can change the Uframe or UTool number after the program has been created by using the Tool offset or Frame offset utilities.
	User Set will appear if *i*RCalibration Master Recovery had previously generated a program for the current group or if the Uframe or UTool were changed by the user. For this case, *i*RCalibration Master Recovery will use the current Uframe or UTool and will not change it when the program is generated.
Calibration Schedule[10] Detection Schedule[10]	The *i*RCalibration Master Recovery program creation always uses Calibration Schedule[10] and Detection Schedule[10].

Procedure 15-8 Creating a Standard *i*RCalibration Master Recovery Program Automatically

Conditions

- The *i*RCalibration Setup status is Done.
- All required calibration hardware has been installed. Refer to Section 15.3.2 .

- You are using a supported six-axis robot

Steps

1. Press MENUS.
2. Select UTILITIES.
3. Press F1, [TYPE].
4. Select iRCalibration. You will see a screen similar to the following:

```
UTILITIES iRCalibration
    1 TCP Shift
    2 TCP Set
    3 Frame Shift
    4 Frame Set
    5 Master Recovery
    6 Master Set
```

5. Select iRCalibration Master Recovery and press ENTER. You will see a screen similar to the following:

```
iRCalibration Master Recovery
1 Robot: Group [1]  Device: Sensor
  Calibration Type: Master Recovery
2 Setup            (    Done)
3 Teach Points     (Not Done)
4 Create Program   (Not Done)
5 Run: RCAL_G1     (Not Done)
6 Update Master CT (Not Done)
7 View Result
```

6. Move the cursor to Teach Points and press ENTER. You will see a screen similar to the following:

```
Program Creation:
  Program Name:     RCAL_G1
  Robot Group:      1
  UTool[10]:        User Set
    X: -115.000 Y:    0.000 Z:  570.000
    W:    0.000 P:  -70.000 R:    0.000
  UFrame[9]:        User Set
    X:  176.000 Y:  600.000 Z: -335.000

  NOTE:  Calibration requires 1 plate

          DONE          CUSTOM
```

Note **If your robot is not supported**, you will see a screen similar to the following. Refer to Procedure 15-9 to create a custom program.

```
Program Creation

  Program Name:      RCAL_G1
  Robot Group:       1
  Calibration Sched: 1
  Detection Sched:   1
  Plate Direction:   –Y
  Plate Thickness:   3.000   mm
  Circle XY offset:  50.000  mm
  Circle Z offset:   10.000  mm
  Robot Calibration
    J1:FALSE    J2:TRUE     J3:TRUE
```

7. To use the automatically generated standard program, press DONE. To customize the standard program, press CUSTOM. Refer to Section 15.3.4.2 .

If all required positions are taught, the Teach Points status will be Done and the cursor will automatically move to the next step. Otherwise, the cursor will stay at Teach Points and its status will remain unchanged.

```
iRCalibration Master Recovery
1 Robot: Group [1]  Device: Sensor
2 Calibration Type: Master Recovery
3 Setup            (Done    )
4 Teach Points     (Done    )
5 Create Program   (Not Done)
6 Run: RCAL_G1     (Not Done)
7 Update Master CT (Not Done)
8 View Result
```

8. With the cursor on Create Program, press ENTER or F4, SELECT to create the program using positions taught in the Teach Points menu.

If the program already exists, you will see the following message:

```
OK to Delete & Re-generate program?
 Press F2 for YES
 Press F3 for NO
```

- **To confirm, and overwrite the existing TP program**, press F2, YES.
- **If you do not want to delete the existing program,** , press F3, NO.

If the program does not exist, you will see the following message:

```
OK to create a new TP program:
 Press F2 for YES
 Press F3 for NO
```

To create the TP program, press F2, YES. To abort, press F3, NO,

The Create Program status on the main menu will change to Done and the cursor will move automatically to Run.

```
iRCalibration Master Recovery
1 Robot: Group [1]  Device: Sensor
  Calibration Type: Master Recovery
2 Setup            (Done    )
3 Teach Points     (Done    )
4 Create Program   (Done    )
5 Run: RCAL_G1     (Not Done)
6 Update Master CT (Not Done)
7 View Result
```

15.3.4.2 Custom Program Creation

The custom program menu will generate programs supporting only the first, vertically oriented, calibration plate. The custom program creation menu does not support generating programs using two calibration plates.

Guidelines for Creating a Custom Robot Calibration Program

Review the following guidelines before you begin to teach points.

- If the starting TCP is all zero, an estimated TCP will be created and applied when the calibration program is created. The seven taught points are also modified to reflect the new estimated TCP. The user must take care that the rotations during program motion do not cause collisions to occur and must insert additional points if needed.
- If the starting TCP is not all zero, then no estimated TCP is used and all program rotations will occur with respect to the starting TCP. The user must take care that the rotations during program motion do not cause collisions to occur and must insert additional points if needed.
- The calibration of Axis 1 and Axis 6 require that both the UFrame and UTool be known. If fixturing with known dimensions is used then the user can set UFrame and UTool to be the correct values and disable solving for UFrame and UTool and enable calibration of J1 and J6.
- If the robot cannot reach a BACK configuration, then the calibration tool dimensions and the location of the touch plate are important. For this case, the calibration tool must be large enough and the touch plate located such that all the axes have significant range of motion (greater than 90 degrees) during the calibration.
- Careful teaching of the calibration points such that they are centered above the touch plate holes and the with the tool shaft oriented perpendicular to the touch plate surface will improve calibration.

There are seven taught points. They are described in Table 15–12 .

Refer to Figure 15–5 and Figure 15–6 for illustrations of the first six taught points.

Table 15–12. Robot Calibration Program Taught Points

Taught Point	Description
Approach Position 1	This point must be located such that the robot can move to both Center Position 1 and Approach position 2 without collision. Also, the wrist must be able to rotate freely at this location without collision.
Center Position 1	This point must be located on the side of the touch plate that is consistent with the “Plate Direction” setup value. This position should be rotated toward one extreme of orientation about the plate axis. This position should be taught centered over the touch plate circle and about one inch above the plate.

Table 15–12. Robot Calibration Program Taught Points (Cont'd)

Taught Point	Description
Rotate Position 1	This point must be located on the same side of the touch plate as Center Position 1. This position should be rotated toward the other extreme of orientation about the plate axis, but the total rotation must be less than 180 degrees. This position should be taught centered over the touch plate circle and about one inch above the plate.
Approach Position 2	This point must be located such that the robot can move to both Center Position 2 and Approach position 1 without collision. Also, the wrist must be able to rotate freely at this location without collision.
Center Position 2	This point must be taught on the opposite side of the plate as Center Position 1. This position should be rotated toward one extreme of orientation about the plate axis. This position should be taught centered over the touch plate circle and about one inch above the plate.
Rotate Position 2	This point must be located on the same side of the touch plate as Center Position 2. This position should be rotated toward the other extreme of orientation about the plate axis, but the total rotation must be less than 180 degrees. This position should be taught centered over the touch plate circle and about one inch above the plate.
Orientation position	This point must be taught with the same orientation as Rotate Position 2 but with the tool shaft extending through the touch plate hole. This position defines the orientation of the tool and +Z will extend out from the tool (such as used by waterjet cutting) if the system variable$CB_VARS.$Z_OUT = TRUE, otherwise +Z will go through the tool (such as used by arc welding) if $CB_VARS.$Z_OUT = FALSE. If the Orientation position is the same as Rotate Position 2 or does not extend through the plate by at least 20mm, then the plate vector is used for determining TOOL Z.

Figure 15–5. Example of First Three Positions

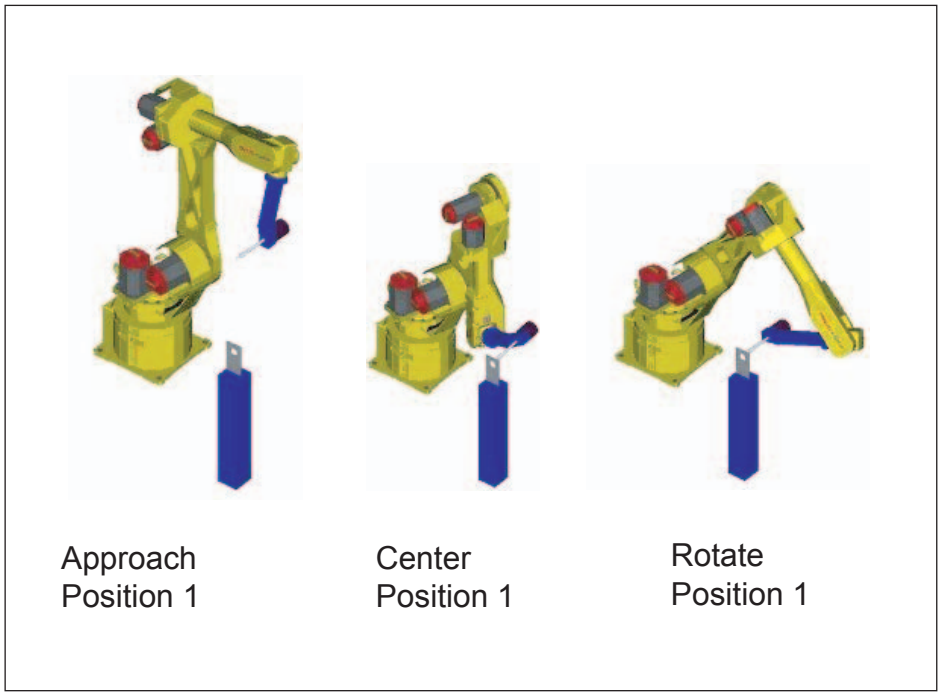
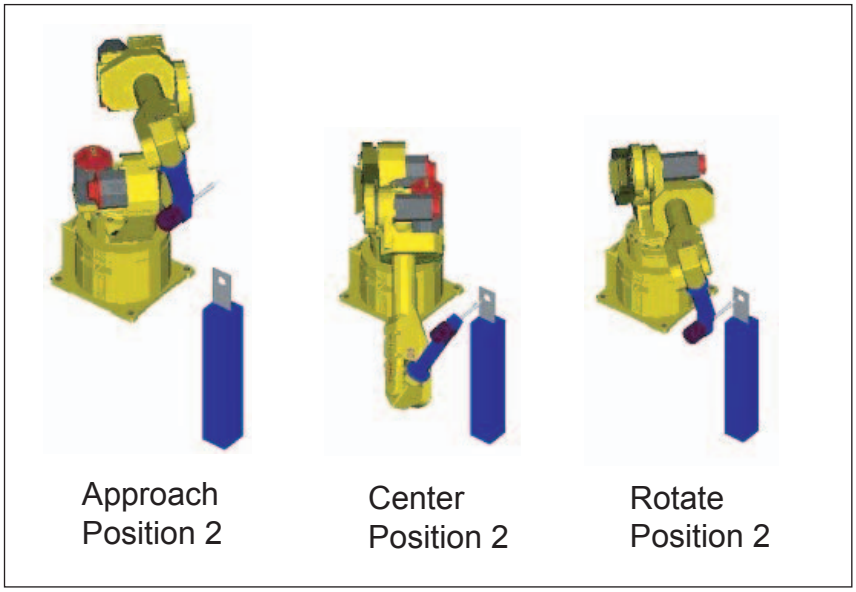


Figure 15–6. Example of Second Three Positions



Note These examples show an M-16 series robot which is capable of the FRONT/BACK and FLIP/NOFLIP configuration change. When the robot can achieve these configuration changes there is a great deal of latitude in the placement of the calibration plate.

The placement shown is not acceptable for robots such as the M-6 series which do not support the FRONT/BACK configuration change. For these robots, we recommend using the Standard Program Creation method using the calibration plate location and UTool that is generated in UFrame[9] and UTool[10] when the *i*RCalibration Mastering program is created. It is recommended that the tooling hardware be constructed to support calibration at these locations.

Six additional points are created by *i*RCalibration Master Recovery:

- Two additional points are created between the approach position and orientation position on each side of the plate, for a total of four additional orientation points.
- One additional point is created where the wrist changes configuration (Flip to Noflip, or vice versa). The user must ensure that no collisions occur during the change of Flip/Noflip configuration and must insert additional points if needed.
- One additional point is created where the robot changes Front to Back configuration if possible. If the robot cannot achieve this configuration, then an additional Front configuration point is created. The user must take care that no collisions occur during the change of Front/Back configuration and must insert additional points if needed.

Procedure 15-9 Creating a Custom *i*RCalibration Master Recovery Program

Conditions

- All required calibration hardware has been installed. Refer to Section 15.3.2 .
- The Setup status is Done.

Steps

1. Press MENUS.
2. Select UTILITIES.
3. Press F1, [TYPE].
4. Select iRCalibration. You will see a screen similar to the following:

```
UTILITIES iRCalibration
       1 TCP Shift
       2 TCP Set
       3 Frame Shift
       4 Frame Set
       5 Master Recovery
       6 Master Set
```

5. Select iRCalibration Master Recovery and press ENTER. You will see a screen similar to the following:

```
iRCalibration Master Recovery
1 Robot: Group [1]  Device: Sensor
  Calibration Type: Master Recovery
2 Setup            (Done    )
3 Teach Points     (Not Done)
4 Create Program   (Not Done)
5 Run: RCAL_G1     (Not Done)
6 Update Master CT (Not Done)
7 View Result
```

6. Move the cursor to Teach Points and press ENTER. You will see a screen similar to the following:

```
Program Creation:
  Program Name:     RCAL_G1
  Robot Group:      1
  UTool[10]:        User Set
    X: -115.000 Y:    0.000 Z:  570.000
    W:    0.000 P:  -70.000 R:    0.000
  UFrame[9]:        User Set
    X:  176.000 Y:  600.000 Z: -335.000

  NOTE:  Calibration requires 1 plate

          CREATE          CUSTOM
```

7. Press F4, CUSTOM. You will see a screen similar to the following:

```
Program Creation

  Program Name:      RCAL_G1
  Robot Group:       1
  Calibration Sched: 1
  Detection Sched:   1
  Plate Direction:   -Y
  Plate Thickness:   3.000   mm
  Circle XY offset:  50.000  mm
  Circle Z offset:   10.000  mm
  Robot Calibration
    J1:FALSE    J2:TRUE     J3:TRUE
    J4:TRUE     J5:TRUE     J6:FALSE
    J7:FALSE    J8:FALSE    J9:FALSE
  Approach Position 1: UNINIT
  Center Position 1:   UNINIT
  Rotate Position 1:   UNINIT
  Approach Position 2: UNINIT
  Center Position 2:   UNINIT
  Rotate Position 2:   UNINIT
  Orientation Position:UNINIT
```

8. Select the plate direction. If the UFRAME number is not 0, this direction is with respect to the current user frame.
9. Enter the thickness of the touch plate.
10. Enter the circle xy offset value based on the dimension of the circle in the touch plate.
11. Enter the circle z offset value based on how deep the tool should go into the hole.
12. Select each axis you want to calibrate and set it to TRUE.
13. To clear all recorded positions, press F1, CLEAR.
14. To load the RCAL_G1 positions to the menu's position data if the TP program already exists, press F2, LOAD.
15. Select each joint you want to calibrate and set it to TRUE.
16. Teach three points on one side of the touch plate: the approach position 1, center position 1, and rotate position 1, as shown in Figure 15–5 .
17. Teach four points on the other side of the touch plate: the approach position 2, the center position 2, the rotate position 2, and the orientation position (not shown in Figure 15–6). The orientation position is taught with the tool shaft extended through the circle.
18. When all positions are recorded, press PREV to return to the main menu. If all required positions are taught, the Teach Points status will be Done and the cursor will move automatically to the next step. Otherwise, the cursor will stay at Teach Points and its status will remain unchanged.

```
iRCalibration Master Recovery
1 Robot: Group [1]  Device: Sensor
  Calibration Type: Master Recovery
2 Setup            (Done    )
3 Teach Points     (Done    )
4 Create Program   (Not Done)
5 Run: RCAL_G1     (Not Done)
6 Update Master CT (Not Done)
7 View Result
```

19. With the cursor on Create Program, press ENTER or F4, SELECT to create the program using positions taught in the Teach Points menu.

If the program already exists, you will see the following message:

```
OK to Delete & Re-generate program?
 Press F2 for YES
 Press F3 for NO
```

- **To confirm, and overwrite the existing TP program**, press F2, YES.
- **If you do not want to delete the existing program,** , press F3, NO.

If the program does not exist, you will see the following message:

```
OK to create a new TP program:
 Press F2 for YES
 Press F3 for NO
```

To create the TP program, press F2, YES. To abort, press F3, NO,

The Create Program status on the main menu will change to Done and the cursor will move automatically to Run.

```
iRCalibration Master Recovery
1 Robot: Group [1]  Device: Sensor
  Calibration Type: Master Recovery
2 Setup            (Done    )
3 Teach Points     (Done    )
4 Create Program   (Done    )
5 Run: RCAL_G1     (Not Done)
6 Update Master CT (Not Done)
7 View Result
```

The following screen shows an example of a custom program:

```
1:  UFRAME_NUM=1
   2:  UTOOL_NUM=1
   3:  Calib Start[1]
   4:J P[1:Approach pos. 1] 10% FINE
   5:L P[2:Center 1 outside] 30deg/sec FINE
   6:  Detect Circle[4]( 30.00, 10.00)World[Y]
   7:L P[3] 30deg/sec FINE
   8:  Detect Circle[4]( 30.00, 10.00)World[Y]
   9:L P[4] 30deg/sec FINE
  10:  Detect Circle[4]( 30.00, 10.00)World[Y]
  11:L P[5:Rotate 1 outside] 30deg/sec FINE
  12:  Detect Circle[4]( 30.00, 10.00)World[Y]
  13:L P[6:Center 1 outside] 30deg/sec FINE
  14:L P[7:Approach pos. 1] 100mm/sec FINE
  15:J P[8:Approach pos. 2] 10% FINE
  16:L P[9:Center 2 outside] 100mm/sec FINE
  17:L P[10] 30deg/sec FINE
  18:  Detect Circle[4]( 30.00, 10.00)World[-Y]
  19:L P[11] 30deg/sec FINE
  20:  Detect Circle[4]( 30.00, 10.00)World[-Y]
  21:L P[12] 30deg/sec FINE
  22:  Detect Circle[4]( 30.00, 10.00)World[-Y]
  23:L P[13:Rotate 2 outside] 30deg/sec FINE
  24:  Detect Circle[4]( 30.00, 10.00)World[-Y]
  25:L P[14:Orient inside] 100mm/sec FINE
  26:  Detect Circle[4](  0.00,  0.00)World[-Y]
  27:L P[15:Rotate 2 outside] 100mm/sec FINE
  28:L P[16:Center 2 outside] 30deg/sec FINE
  29:L P[17:Approach pos. 2] 100mm/sec FINE
  30:J P[18:Approach 2 back] 10% FINE
  31:L P[19] 100mm/sec FINE
  32:  Detect Circle[4]( 30.00, 10.00)World[-Y]
  33:L P[20] 100mm/sec FINE
  34:L P[21:Approach 2 back] 100mm/sec FINE
  35:J P[22:Approach 1 flip] 10% FINE
  36:L P[23:Center 1 flip] 100mm/sec FINE
  37:  Detect Circle[4]( 30.00, 10.00)World[Y]
  38:L P[24:Center 1 flip] 100mm/sec FINE
  39:L P[25:Approach 1 flip] 100mm/sec FINE
  40:J P[26:Approach pos. 1] 10% FINE
  41:  Calib END [Recovery]
```

15.3.5 Running the *i*RCalibration Master Recovery Program

When all the points are taught and the program has been created, the Create Program status will be Done and the cursor will move automatically to Run.

*i*RCalibration Master Recovery

- Allows you to recover mastering after an event such as a motor change or collision. The first time you run the recovery program, *i*RCalibration Mastering does the "Mastering" of the teach pendant Calibration program. No update is required.

  After the teach pendant program has been mastered, you can recover mastering data by running the calibration program again and using the Update Master CT menu to update the mastering data.

  *i*RCalibration Master Recovery can recover mastering in one of two ways

  — **Multiple axis recovery**. This requires a program either generated by *i*RCalibration Mastering or written and/or modified by the user. All the detection instructions must be DETECT CIRCLE instructions and a calibration plate is required. It is recommended that the program creation menu be used to generate this *i*RCalibration Mastering program. There must be significant (at least 60 degrees) motion on all axes except perhaps J4. It is highly recommended for robots capable of FRONT/BACK configuration change that the calibration program contain both configurations as well as a FLIP/NOFLIP configuration.

  These constraints will guarantee sufficient motion on each axis. If the robot does not support such configuration changes, either by robot joint limits or by dressout limitations, then special care must be taken in choosing a calibration tool and locating the calibration plate. Special calibration tooling may be required.

  — **Single axis recovery**. This requires a program written by the user to detect contact by moving a single axis. The axis that moves is the axis that can be recovered. This program must be mastered with Master Recovery. This method will work with either robot axes, positioner axes, or aux axes. There must be exactly one DETECT JOINT instruction in the user-written *i*RCalibration Mastering program. Refer to Section 15.3.6 .

Procedure 15-10 Running the *i*RCalibration Master Recovery Program

Conditions

- The Create Program status is Done.

Steps

1. Press MENUS.
2. Select UTILITIES.
3. Press F1, [TYPE].
4. Select *i*RCalibration. You will see a screen similar to the following:

```
UTILITIES iRCalibration
       1 TCP Shift
       2 TCP Set
       3 Frame Shift
       4 Frame Set
       5 Master Recovery
       6 Master Set
```

5. Select iRCalibration Mastering and press ENTER. You will see a screen similar to the following:

```
iRCalibration Master Recovery
1 Robot: Group [1]  Device: Sensor
  Calibration Type: Master Recovery
2 Setup            (Done    )
3 Teach Points     (Done    )
4 Create Program   (Done    )
5 Run: RCAL_G1     (Not Done)
6 Update Master CT (Not Done)
7 View Result
```

6. Move the cursor to Run and press ENTER. You will see a screen similar to the following:

```
OK to run calibration program?
 Press F2 for YES
 Press F3 for NO
```

7. To run the program, press F2, YES. You will be asked to confirm your selection. You will see the following message:

```
Execute Calibration Program:
 Press F2 for YES
 Press F3 for NO
```

8. Be sure that the teach pendant is ON and all faults have been cleared. Using a low override, press SHIFT+F2 to run the program for the first time to make sure all positions are reachable without colliding with obstacles. After the program runs through without collision, run the program again at 100% override to calibration the robot.

The Run status will remain “Not Done” until the TP program has run successfully.

```
iRCalibration Master Recovery
1 Robot: Group [1]  Device: Sensor
  Calibration Type: Master Recovery
2 Setup            (Done    )
3 Teach Points     (Done    )
4 Create Program   (Done    )
5 Run: RCAL_G1     (Done    )
6 Update Master CT (Not Done)
7 View Result
```

You should always verify the generated calibration program by stepping through it at a low override. Be careful that the robot does not collide with objects, especially for the large rotations generated between the FRONT/BACK and FLIP/NOFLIP configurations. During single-step the detection instructions are not executed, so verifying the program is relatively quick.

You can modify the generated program, but any modifications will be lost if you regenerate the program. Also, certain modifications could prevent *i*RCalibration Master Recovery from loading the program positions, should you want to regenerate it. It is always a good idea to keep a backup before modifying a generated *i*RCalibration Master Recovery program.

It is recommended to run *i*RCalibration Master Recovery at least twice to get optimal results. Especially if there were significant orientation correction to the taught points, the second execution of *i*RCalibration Mastering can result in a better calibration and lower calibration errors.

15.3.6 Single Axis Mastering Recovery

*i*RCalibration Master Recovery can recover mastering by a very simple program with only a single axis to move and cause detection. You must write a program for each axis to be calibrated; *i*RCalibration Master Recovery does not generate programs for single axis recovery.

The following is an example program for single axis recovery:

```
1:J P[1:Approach Point] 5% FINE
2:L P[2:Detect Point] 100mm/sec FINE
3:  Calib Start[1]
4:  Detect Joint[1] Axis(1,+)
5:  Calib END[Recovery]
6:L P[1:Approach Point] 100mm/sec FINE
```

You can write a similar program to recover mastering any axis, including robot axes, aux axes, and positioners. The parameter of Detect Joint[] is the detection schedule number. The first parameter of Axis() is the axis number. The second parameter is the direction of motion of that axis, either plus or minus for the detection motion.

If more than one axis is to be recovered (one at a time), it is recommended that care be used in determining the locations of detections. You should not have two detections on the same mechanical unit to search in approximately the same direction and use the same contact point. You can use any point on the unit to make contact, it does not need to be near the TCP. If using this method on a multiple axis robot or positioner, it is beneficial to have the contact on the link immediately next to the axis being calibrated. Then by recovering from the base axis out each axis has the benefit of the previous axes being already calibrated.

15.3.7 Updating the Master Count

The Update Master Count menu is not accessible unless you have completed all the steps before it. It has two different menus, based on the Calibration Type.

Procedure 15-11

Conditions

- You have run the *i*RCalibration program for New Calibration or Master Recovery.

Steps

1. Press MENUS.
2. Select UTILITIES.
3. Press F1, [TYPE].
4. Select iRCalibration. You will see a screen similar to the following:

```
UTILITIES iRCalibration
       1 TCP Shift
       2 TCP Set
       3 Frame Shift
       4 Frame Set
       5 Master Recovery
       6 Master Set
```

5. Select iRCalibration Master Recovery and press ENTER. You will see a screen similar to the following:

```
iRCalibration Master Recovery
1 Robot: Group [1]  Device: Sensor
  Calibration Type: New Calibration
2 Setup            (Done    )
3 Teach Points     (Done    )
4 Create Program   (Done    )
5 Run: RCAL_G1     (Done    )
6 Update Master CT (Not Done)
7 View Result
```

6. Move the cursor to Update Master CT and press ENTER. You will see a screen similar to the following.

```
Calibration Result

  Robot Setting [G:1]
    J1:FALSE   J2:TRUE   J3:TRUE
    J4:TRUE    J5:TRUE   J6:FALSE
    J7:FALSE   J8:FALSE  J9:FALSE
  Recovery Result:
    J1:  0.000 J2:  0.000 J3:0.000
    J4:  0.000 52:  0.000 J6:0.000
    J7:  0.000 J8:  0.000 J9:0.000
  Mean Error:  0.000 mm
  Max. Error:  0.000 mm
```

7. Press F2, UPDATE, to update the master count.

8. Press PREV to return to the *i*RCalibration Master Recovery main menu.

15.3.8 Viewing the Calibration Result

The *i*RCalibration Master Recovery View menu allows you to see the results of previous calibrations. You can access the view menu at any time to review what has been done with *i*RCalibration Master Recovery earlier, even after a cold start. The screen you see will be based on the calibration type selected in the main menu

The calibration log is a buffer that holds previous mastering information. It has a fixed size of 20 entries. When the buffer is full, it will replace the oldest data in the buffer with the new logged data. If you want to view mastering information, you can display the Calibration Result screen. To display the calibration result screen, use Procedure 15-12 .

Procedure 15-12 Viewing the Calibration Result

Conditions

- You have installed the *i*RCalibration Master Recovery software option.

Steps

1. Press MENUS.
2. Select UTILITIES.
3. Press F1, [TYPE].
4. Select iRCalibration. You will see a screen similar to the following: l and press ENTER. You will see a screen similar to the following.

```
UTILITIES iRCalibration
       1 TCP Shift
       2 TCP Set
       3 Frame Shift
       4 Frame Set
       5 Master Recovery
       6 Master Set
```

5. Select iRCalibration Master Recovery and press ENTER. You will see a screen similar to the following.

```
iRCalibration Master Recovery
1 Robot: Group [1]  Device: Sensor
  Calibration Type: Master Recovery
2 Setup            (Done    )
3 Teach Points     (Done    )
4 Create Program   (Done    )
5 Run: RCAL_G1     (Done    )
6 Update Master CT (Done    )
78 View Result
```

6. Move the cursor to View Result and press ENTER. You will see a screen similar to the following.

```
Calibration Result

  Robot Setting [G:1]
    J1:FALSE   J2:TRUE   J3:TRUE
    J4:TRUE    J5:TRUE   J6:FALSE
    J7:FALSE   J8:FALSE  J9:FALSE
  Recovery Result:
    J1:  0.000 J2:  0.000 J3:0.000
    J4:  0.000 52:  0.000 J6:0.000
    J7:  0.000 J8:  0.000 J9:0.000
  Mean Error:  0.000 mm
  Max. Error:  0.000 mm
```

7. To view the Master Log, press F4, VIEW LOG. You will see a screen similar to the following:

```
Master Log

  Date   Time    J1(47)   J2(58)   J3(69)
14-MAY-08 10:59      0  19187  -11293
  Group Num[  1]  2766   767 0
  Extended Axes      0       0    0
14-MAY-08 11:02      0   31495  -23804
  Group Num[  1]  2881   -3197    0
  Extended Axes      0       0    0
No  J1 J2 J3 Data
No  J4 J5 J6 Data
No  J7 J8 J9 Data
```

8. To save the calibration log to a text file in the "PATH:\calb_log.dt", press F2, SAVE.

To clear the calibration log, press F4, CLEAR.

To recover a previous master count data, press F3, APPLY.

9. Press PREV to return to the View Calibration Result page.

15.3.9 Troubleshooting *i*RCalibration Mastering

Troubleshooting *i*RCalibration Mastering

Refer to Table 15–13 for troubleshooting information.

Table 15–13. Troubleshooting

Problem	Possible Cause	Solution
The robot does not detect the Touch Plate for the I/O method.	I/O was incorrectly set up.	Verify that the I/O is set up properly.
		Verify that contact with the Touch Plate will cause the specified I/O signal to go ON. Change polarity of the I/O point if necessary.
	The Touch Plate or tool is not isolated.	Verify that either the tool is isolated (for the case of a charged tool) or that the Touch Plate is isolated (for the case of a charged plate) from ground, and that the element that is not charged is grounded electrically.
	The tool or Touch Plate surfaces are dirty.	Verify that both the tool and plate surfaces are clean and free from film.
The robot hits obstacles during rotations.	There is not enough space around the Touch Plate.	Touch up the calibration program's position.
The robot does not always touch the inside of the Touch Plate hole with a cylindrical portion of the tool.	The shaft is not far enough through the Touch Plate hole.	Touch up the calibration program's position.
When is it appropriate to use the I/O method and when is it appropriate to use Torque Observer Sensor (TOS) method?	There is delicate or compliant tooling.	Use I/O method if there is delicate or compliant tooling.
	The wrist center is in the same plane as the Touch Plate.	Use I/O method if the wrist center is to close to or in the same plane as the Touch Plate. If the tool is durable, TOS All Axes can also be used.
	The wrist center is directly above the Touch Plate.	Use I/O method if the wrist center is directly above the Touch Plate. If the tool is durable, TOS All Axes can also be used.
	There is a durable tool and a small robot.	The TOS All Axes method is recommended for small robots when the I/O method is not used.
	There is a durable tool and a large robot.	Use the TOS All Axes method if there is a durable tool.
Large Calibration Error	Incorrect UFrame or UTool setting	Make sure that the solve UFrame and solve UTool flags are TRUE unless the UFrame exactly corresponds to the calibration plate location and unless the UTool exactly corresponds to the calibration tool.
	TOS false detections	Rerun the calibration and watch to confirm the tool touches the calibration plate for every detection. If it does not touch in one or more detections, increase the TOS Tolerance or increase the learning speed and learning distance or reteach the point with a slightly different rotation.

15.4 iRCALIBRATION TCP SET

15.4.1 Overview

Note *i*RCalibration TCP Set is not available for all software application Tools.

The *i*RCalibration TCP Set option, formerly TCPCal, provides a cost effective, easy-to-use solution to set the Tool Center Point (TCP) automatically.

When you use *i*RCalibration TCP Set you start with a fully functional system and an unknown or inaccurate TCP. You then set up *i*RCalibration TCP Set to generate an accurate TCP.

*i*RCalibration TCP Set works with any robot and tool where the TCP is at the end of cylindrically-shaped tooling. If the normal tooling is not cylindrical, or if the TCP is away from the tooling, a teach tip can be installed.

To use *i*RCalibration TCP Set, you must do the following:

1. Install the *i*RCalibration TCP Set software option and any required hardware.
2. Set up *i*RCalibration TCP Set.
3. Create a TCP calibration TP program.
4. Run *i*RCalibration TCP Set to set the initial TCP value.

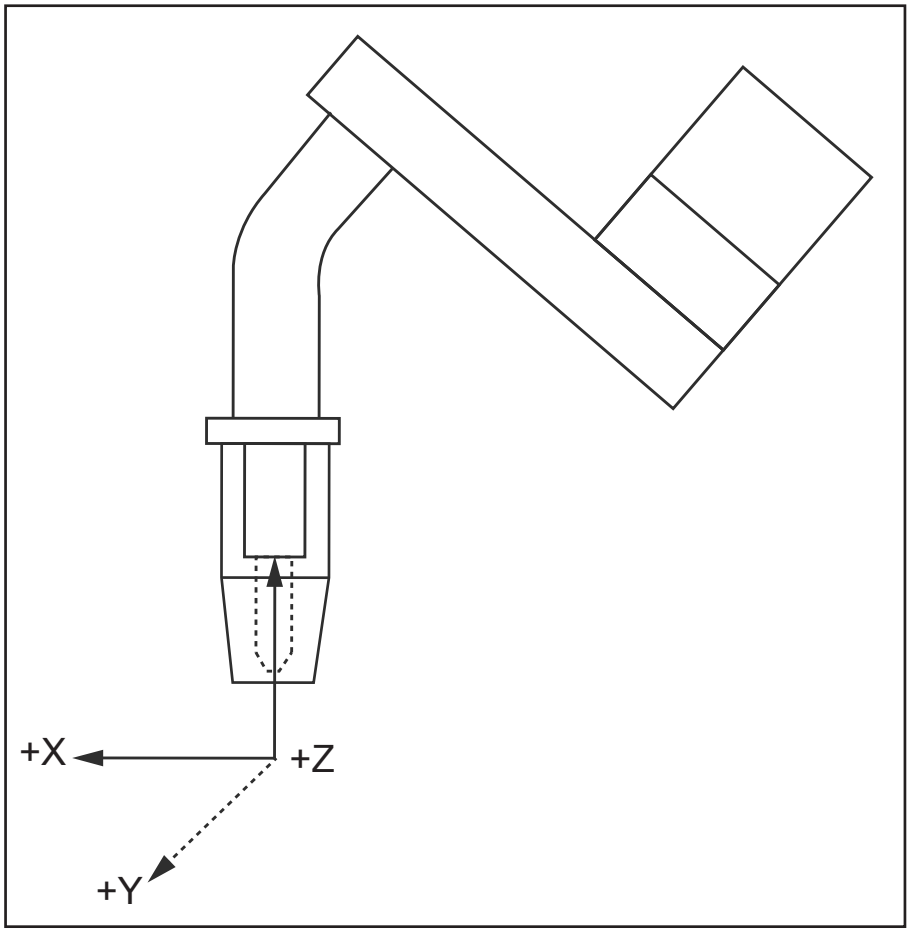
15.4.2 Preliminary Setup

Before you can set up and use *i*RCalibration TCP Set you must do the following:

- install the *i*RCalibration TCP Set software option. Refer to the *Software Installation Manual* for more information about installing software options.
- Install the touch plate on a stationary object in the robot workcell. Refer to Section 15.4.2.1

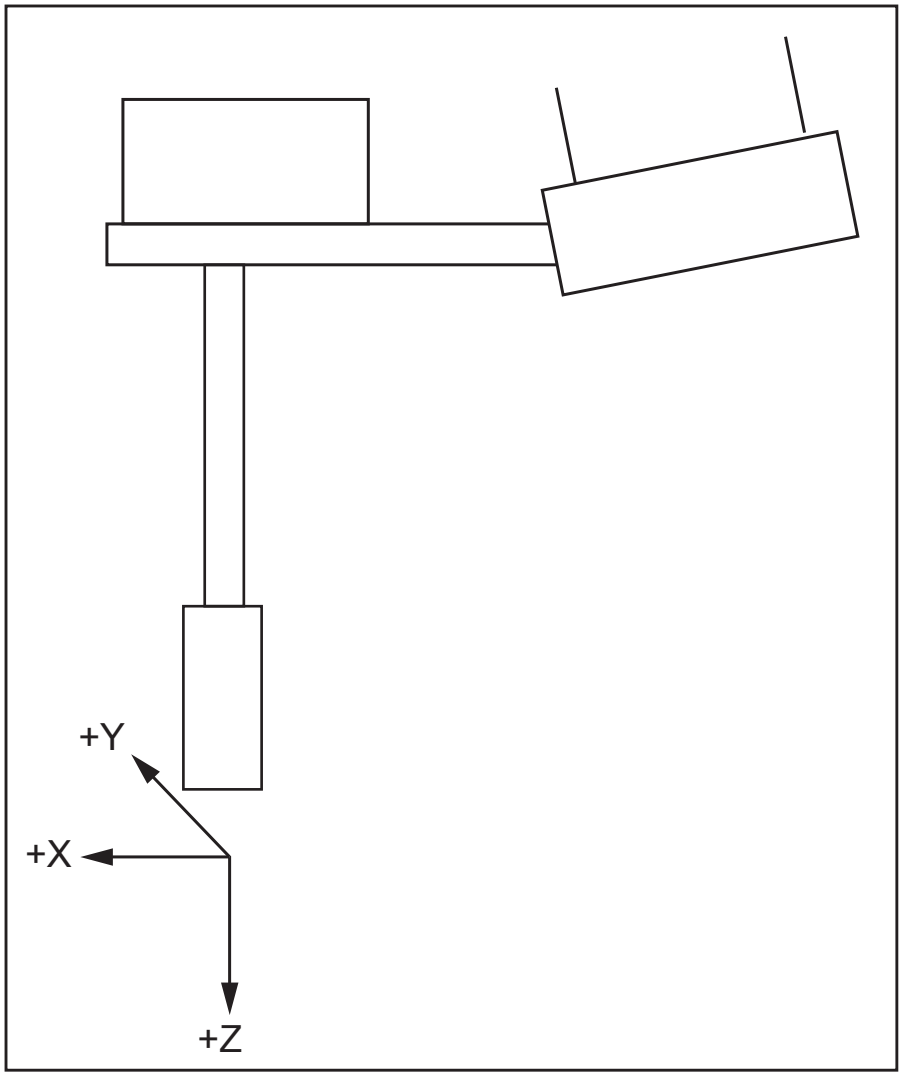
You will then use the *i*RCalibration TCP Set software to set up the initial tool center point. The following figures show typical arc welding and waterjet tool definition.

Figure 15–7. Typical Arc Welding tool definition



Typical Arc Welding tool definition $CB_VARS.$Z_OUT = FALSE, set automatically on initial start for ArcTool.

Figure 15–8. Typical Waterjet Tool Definition

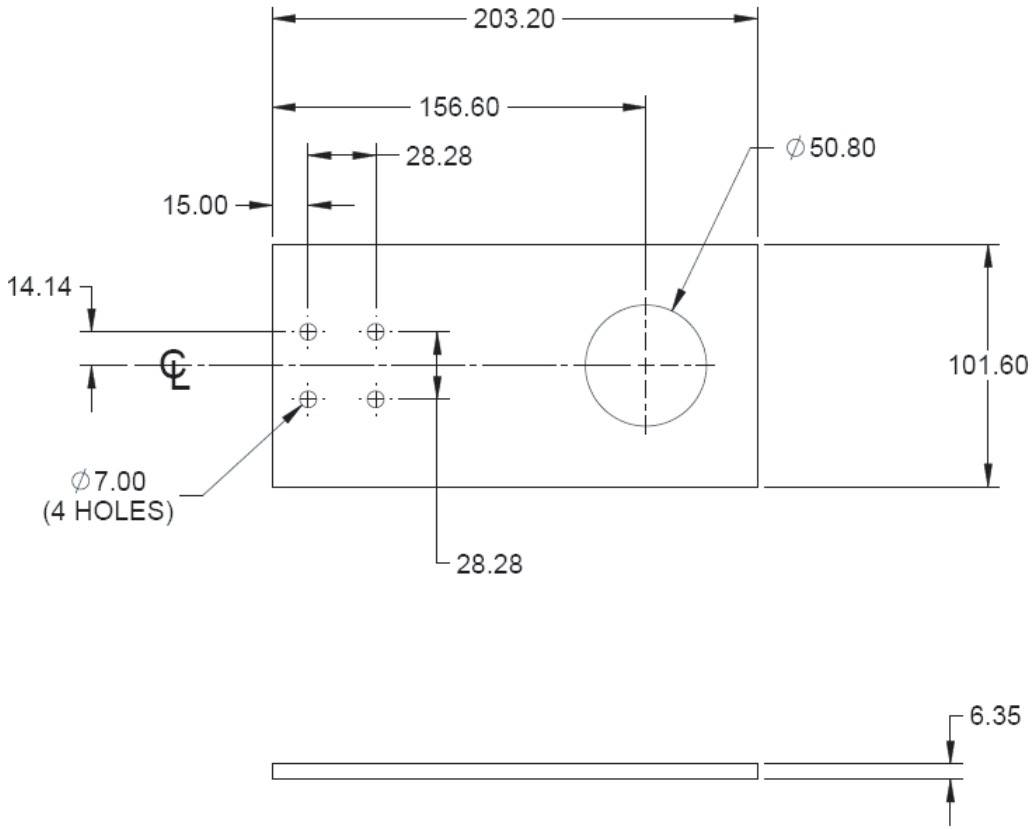


Typical Waterjet Tool Definition, $CB_VARS.$Z_OUT = TRUE, default for all applications except ArcTool

15.4.2.1 Touch Plate Installation

Before you can set up and run *i*RCalibration TCP Set, you must install a touch plate on a stationary object for pointer type tools, within the robot workcell. See Figure 15–9 for an illustration of a recommended touch plate and pointer.

Figure 15–9. Recommended *i*RCalibration TCP Set Touch Plate and Pointer Specifications



Use Procedure 15-13 to install and align the touch plate.

Procedure 15-13 Installing and Aligning the Touch Plate

Conditions

- The TOOL frame has been set up correctly.

  Refer to “Frames Setup” in the application-specific *Setup and Operations Manual* for more information.

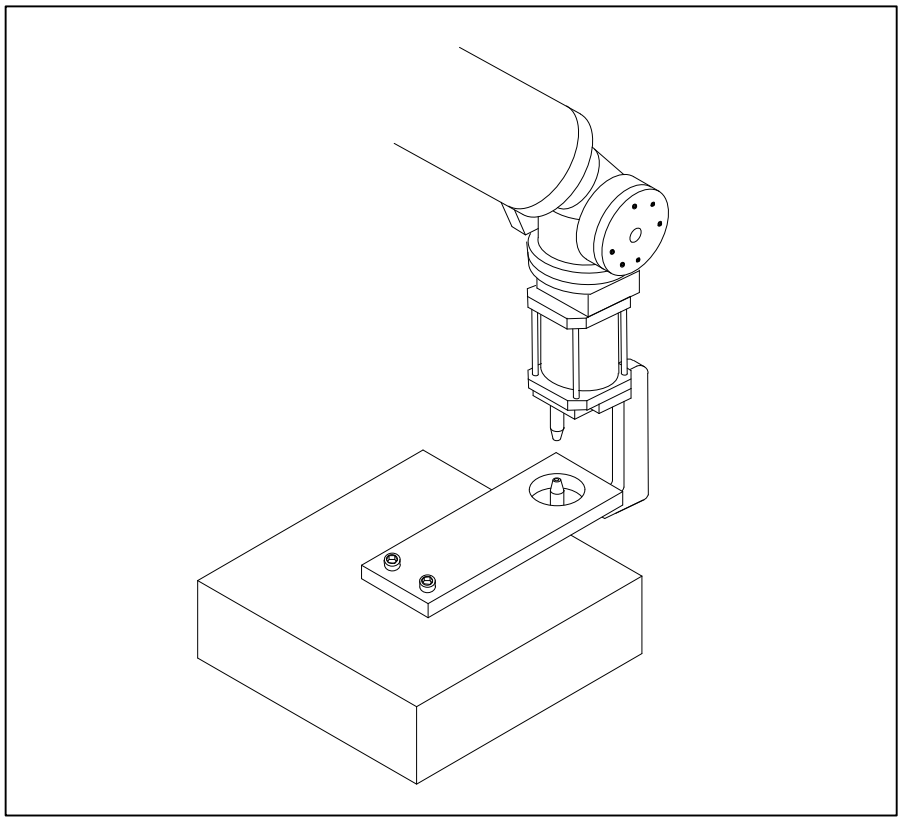
Steps

1. Install the touch plate (touch pointer for Cal-Plate calibration) on a flat level surface well within the workcell. See Figure 15–10 .

   The touch plate or touch pointer should be securely fastened to a firm surface. If the touch plate position changes due to collisions or other events, the recovered TCP will be incorrect.

There should be ample clearance below the plate so that the tooling can be inserted through the circular opening in the touch plate.

Figure 15–10. Installing the Touch Plate



2. For *i*RCalibration TCP Set, *i*RCalibration Master Set, and *i*RCalibration Master Recovery, the touch plate is typically oriented in a vertical plane as shown in Figure 15–11 and Figure 15–12 . This vertical orientation (with the plate typically parallel to the robot WORLD x-z or y-z plane) makes it easier to orient the tool so it can touch both sides of the plate with the proper orientation and without encountering axis limits.

   In most applications, the typical TCP definition is that the TOOL z axis is along the shaft used by *i*RCalibration TCP Set. For arc welding applications, the TOOL +z direction extends from the TCP location into the torch body. For other applications, the TOOL +z direction extends away from the shaft at the TCP location. *i*RCalibration TCP Set works with any TOOL frame definition.

   If a touch pointer is used for Cal-Plate calibration, align the pointer so it is pointing in one of the world X, -X, Y, -Y, Z or -Z directions.

   Installation and alignment of the touch plate is now complete.

Note If you are using ArcTool and require compatibility with TorchMate, refer to the "TorchMate" chapter in the *ArcTool Setup and Operations Manual* for further information. *i*RCalibration TCP Set can coexist with TorchMate.

15.4.3 *i*RCalibration TCP Set Setup

Before you can use *i*RCalibration TCP Shift to set initial TCP, you must perform some setup procedures. Setup involves the initial setting of menu options within *i*RCalibration TCP Set software.

Setting up and running *i*RCalibration TCP Set includes the following processes:

- Selecting the motion group you want to calibrate
- Selecting the teach pendant program, the UTool you want to calibrate, and the calibration mode: New XYZWPR, New XYZ or Cal_Plate. Refer to Table 15–14 for details of the calibration modes.
- Setting up the UTool schedule. Refer to Table 15–15 for a description of the items on the UTool Schedule Detail Screen.
- Setting up the detection schedule. Refer to Table 15–16 for a description of the items on the *i*RCalibration TCP Set Detection Schedule Detail Screen.
- Teaching points for the teach pendant calibration program. Refer to Procedure 15-15 .
- Creating the teach pendant calibration program from the taught points and running the calibration program. You can have *i*RCalibration TCP Set create the program for you automatically (refer to Procedure 15-15).
- Updating the UTool with the calibration results obtained from running the calibration program. Refer to Procedure 15-17 .

Table 15–14. Calibration Mode Descriptions

Calibration Mode	Description
New XYZWPR	Sets initial TCP xyzwpr
New XYZ	Sets initial TCP xyz
Cal-Plate	Sets initial TCP XYZWPR location of a calibration plate mounted on the faceplate.

Table 15–15. *i*RCalibration TCP Set UTOOL Schedule Item Description

ITEM	DESCRIPTION
Auto Update Default: TRUE	This item indicates whether *i*RCalibration TCP Set's UTOOL END instruction will update the corresponding $mnutool or not. If you set Auto Update to TRUE and run the TCP calibration program, the $mnutool will be updated based on the offset.
Store Offset Default: FALSE	This item indicates whether or not *i*RCalibration TCP Set's UTOOL END instruction will update the position register with the TCP offset.
Tool Z Offset Units: mm Range: – 500.000– 500.000 Default: 0	This item is the distance from the point of the tooling that contacts the plate to the actual TCP. This is used when XYZWPR mode is chosen to adjust the XYZ correction at the TCP properly when there is a tool Z change. For spot welding and other applications where the contact point is the actual TCP location, this value should be zero. For Waterjet, arc welding, and other applications where the contact point is above the TCP at the time of contact, this value should be set to the distance from the contact point to the actual TCP.
Max. XYZ Offset Units: mm Range: 0.1 — 1000 Default:	This item is the distance in mm from the previous TCP to the new TCP for a single correction. If a single correction is larger than the max. XYZ offset, an error message will be posted. If the out DOUT port is not zero, *i*RCalibration TCP Set will also set the port to ON.
Max. Orent Offset Units: deg Range: 0.01 — –20 Default:	This item is the orientation change in degrees from the previous TCP to the new TCP for a single correction. If a single correction is larger than the max. orientation offset, an error will be posted.
Max. Alignment Offset Units: mm Range: 0 — 500 Default: 10	This item is the maximum alignment change allowed in mm from the previous TCP to the new TCP. If the alignment change is greater than the alignment offset, an error will be posted. This item is in effect when the XYZ-Align calibration mode is selected.
Deviation Register Range: 0 — 999 Default: 0	This item indicates the register number where *i*RCalibration TCP Set will update with the change between the previous TCP and the new TCP. If this item is 0, no update is performed.
Accum. Register Range: 0 — 999 Default:	This item is the change between the new TCP and the original mastered TCP. If this item is not 0, *i*RCalibration TCP Set will update the register with the accumulated deviation value.
Accumulated Error Enable Default: FALSE	If this item is TRUE, *i*RCalibration TCP Set will check the accumulated deviation against the accumulated error threshold. If the deviation is greater than 0, an error will be posted.
Accumulated Offset Threshold Default: 25 Min = 0 Max = 500mm	When the Accumulated Offset Threshold is exceeded and the function is enabled, *i*RCalibration TCP Set will post and ACAL-060, Excessive Accumulated Offset warning.

When you set up a detection schedule, you

- determine the sensor
- determine the detection motion
- determine TOS

Note Default values are usually acceptable.

Table 15–16. *i*RCalibration TCP Set Detection Schedule Description

ITEM	DESCRIPTION
Sensor Type Values: TOS Wrist, I/O, TOS ALL AXES Default: TOS ALL AXES	This item indicates the kind of sensor used for contact detection. The three modes are: • **TOS Wrist** (Torque Observer Sensor on Wrist) - the contact torque will be monitored, and only the observed torque on the wrist will be used to determine the contact position. When TOS Wrist is selected, no I/O connection is required for contact detection. TOS Wrist is very sensitive and can be more prone to false detections in certain cases than the more robust TOS All Axes method. TOS Wrist is not recommended except for special cases where it can verified that false detections will not occur. • **I/O** - the specified I/O will be monitored and only the state of this input will be used to determine the contact position. Use the detection port for contact detection with I/O. Refer to Detection Port item in this table for more information. • **TOS ALL AXES** (Torque Observer Sensor, All Axes) - The contact torque will be monitored on all robot axes, and only the observed torque will be used to determine the contact position. This is the default value. When TOS ALL AXES is selected, no I/O connection is required for contact detection.

Table 15–16. *i*RCalibration TCP Set Detection Schedule Description (Cont'd)

Detection Port Values: RDI, DI, WDI, WSI Note: AIN is not supported in *i*RCalibration TCP Set.	This item indicates the desired input port type. The port types are: • **RDI** - Robot Digital Input • **DI** - normal Digital Input • **WDI** - Weld Digital Input • **WSI** - Wire Stick circuit Input. The port type that you specify is followed by the desired port number that will go from OFF to ON when contact is detected. The WDI and WSI ports are typically used for arc welding. You will set this item up if you selected I/O as the Sensor Type. You do not need to set this up if you selected TOS Wrist or TOS All Axes as the Sensor Type.
Sensor Enable Values: RDO, DOUT, WDO, or WSO Default: RDO	This item indicates the I/O sensor enable digital output port type. The port types are: • **RDO** - Robot Digital Output • **DOUT** - normal Digital Output • **WDO** - Weld Digital Output • **WSO** - Wire Stick circuit Output. The port type that you specify is followed by the desired port number that will be used to enable the I/O detection circuit. The WDO and WSO ports are typically used for arc welding. You will set this item up if you selected I/O as the Sensor Type. Set the port number to 0 (zero) if you are not using this feature.
Detection Speed Units: mm/sec Range: 1.0 - 100 Default: 10	This item is the search speed for contact detection. This speed is used for both TOS and I/O searches. The value can be increased if cycle time is of great concern (although increasing the speed will also affect the resultant accuracy and repeatability) , or reduced if vibration or other disturbances cause *i*RCalibration TCP Set to retry excessively.
Detection Distance Units: mm Range: 1 - 1000 Default: 50	This item is the maximum distance that the robot will move while looking for the signal.
Return Speed Units: mm/sec Range: 5 - 5000 Default: 50	This item is the search return speed. This speed is used for both TOS and I/O searches.

Table 15–16. *i*RCalibration TCP Set Detection Schedule Description (Cont'd)

Return to Start Default: Yes	When this item is set to YES, the robot will move back to the position before it starts the search motion. When this item is set to NO, the robot will move back to a distance specified by the next item (Return distance)
Return Distance Units: mm Range: 1 - 1000 Default: 50	When return to start is set to FALSE, the robot will move back to the distance specified. This function applies to DETECT POINT instructions only.
Return term type Default: FINE	This item is the termination type for return motion.
Air Cut Speed Units: mm/sec Range: 1 - 1000 Default: 100	This item is the non-search motion speed.
Learn Distance Units: mm Range: 0.5– 20.0 Default: 5.0	For "TOS All Axes" method only, there is a short motion before each search to identify dynamic torque values. Learn Distance is the distance in millimeters of this short motion, the direction is determined at the time of the motion.
Learn Speed Units: mm/sec Range: 0.5 – 50.0 Default: 10.0	For "TOS All Axes" method only, this is the speed of the learning motion before each search.
Sensitivity Tolerance Units: max. motor torque/sec. Range: 1 - 999 Default: 10	This item is the primary sensitivity tolerance of TOS Wrist and is not used for TOS All-Axes. It is the rate of change of the disturbance torque. The actual units vary with motor model, gear ratio, and amplifier current, and are affected by tooling length. However, the effect for calibration is largely independent of these factors and a value of 10 is considered normal operation for typical installations. You should not use a value of less than 10 or misdetections might occur. This value can be increased if false detections occur frequently.  **Warning** **Be careful when you change this number. Changing the sensitivity tolerance might cause a false detection. This could injure personnel or damage equipment.**

Table 15–16. *i*RCalibration TCP Set Detection Schedule Description (Cont'd)

Torque Threshold Units: max. motor torque Range: 1 to 999 Default: 100	For **TOS All Axes** , this item is the only measure of contact detection. During the learning motion a torque level is memorized and that level plus **Torque Threshold** must be exceeded for contact to be determined. If false detections are being experienced then this value should be increased. **For TOS Wrist** , this item is the level of disturbance torque that is used as another means for contact detection. The default value of 100 is considered normal operation for typical installations. If increasing the value of **Sensitivity Tolerance** does not make false detections disappear, then this value should be increased until the false detections disappear; then, reset the value of **Sensitivity Tolerance**.
Fit Tolerance Units: mm Range: 0.1 - 100 Default: 0.5	For Detect Circle instruction: The robot detects 4 points inside the circle and the calibration software will fit a circle to the 4 detected positions. The fit error is defined as the maximum distance from the 4 points to the circle. If the fitting error exceeds the tolerance, the calibration software will post an error.
Radius Tolerance Units: mm Range: 0.1— 100 Default: 0.5	The Detect Circle instruction stores the circle's radius during mastering. At run time, if the computed circle radius differs from the stored radius value and difference exceeds the tolerance, the calibration software will post an error.
Number of Retry Range: 0 - 9 Default: 0	If the number of retry is >0, the calibration software will re-run the detect circle instruction if the current circle detection has either a fit error or a radius error.
Contact Position Register Range 0-100 Default 0	If the number is non-zero. *i*RCalibration TCP Set records the contact position to the specified position register.

15.4.3.1 Setting Up *i*RCalibration TCP Set

*i*RCalibration TCP Set uses a wizard-style menu. It shows the steps that need be done to complete the TCP calibration process. Each step of the process has its own status display that shows where you are in the setup process. You cannot skip steps and you cannot go to the next step without completing the current step.

*i*RCalibration TCP Set also includes a menu item, View Result, that lets you review the previous calibration result and log history.

Procedure 15-14 Setting Up *i*RCalibration TCP Set

Conditions

- The *i*RCalibration TCP Set software option is loaded.

- Preliminary setup is complete. Refer to Section 15.4.2 .

Steps

1. Press MENUS.
2. Select UTILITY.
3. Press F1, [TYPE].
4. Select *i*RCalibration. You will see a screen similar to the following:

```
UTILITIES iRCalibration
   1 TCP Shift
   2 TCP Set
   3 Frame Shift
   4 Frame Set
   5 Master Recovery
   6 Master Set
```

5. Select *i*RCalibration TCP Set and press ENTER. You will see a screen similar to the following:

```
iRCalibration TCP Set
        1 Robot: Group [1]  Device: Sensor
          Calibration Type: New TCP
        2 Setup            (Not Done)
        3 Teach Points     (Not Done)
        4 Create Program   (Not Done)
        5 Run: TCAL_G1     (Not Done)
        6 Update UTool[ 1](Not Done)
        7 View Result
```

6. Move the cursor to Robot: Group [1], type the group number of the motion group you want to calibrate, and press ENTER.

   Note Device is always Sensor. You cannot change it.

7. **Specify the Calibration Program Name**. The *i*RCalibration TCP Set Setup menu allows you to specify the TCP calibration TP program name, the UTool number, and the calibration mode, and select the *i*RCalibration TCP Set calibration schedule and the detection schedule.

   Note The Group number is provided for reference. You cannot change it in this menu.

   a. Move the cursor to Setup and press ENTER or F4, SELECT. You will see a screen similar to the following:

```
New TCP Setup
        1 Program name: TCAL_G1    Group[1]
        2 UTool Number:     1
        3 Calibration Mode: New XYZWPR
        4 UTool Schedule:     1
        5 Detect Schedule:  1
```

b. The default program name is TCAL_Gx, where x is the group number. If you want to specify a different program name, press ENTER and use the function keys.

8. Select the UTool. Move the cursor to UTool Number:, type the number of the Tool Frame you want to calibrate, and press ENTER.

9. Select the Calibration Mode. Move the cursor to Calibration Mode. Press F4, [CHOICE] and make your selection.

- The calibration mode choices are New XYZWPR, New XYZ, and Cal_Plate.

Note The TOOL frame is relative to the touch plate, and the TOOL frame z axis corresponds to the touch plate z axis. Because the touch plate is always aligned with its own x-y axis, regardless of the physical mounting of the plate in WORLD space, the WORLD y-z axis and the touch plate z axis both align with the WORLD x axis. It might be helpful to think in terms of correcting for the TOOL axes that correspond to PLATE xy, PLATE xyz, PLATE xyzwpr, and PLATE

Refer to Table 15–14 for more information.

10. Select the UTool Schedule and Set UTool Schedule detail items. Move the cursor to UTool Schedule. Refer to Table 15–15 for information on the UTool Schedule items.

1. To change the number of the TCP, move the cursor to the schedule number, type the number of the UTool Schedule you want, and press ENTER.
2. To see the detail screen for the selected UTool, press ENTER. You will see a screen similar to the following:

```
UTool Schedule                 1/12
   Schedule Number[  1]
   1 Auto Upate:            TRUE
   2 Store Offset: FALSE
   3               In PR[0  ]
   4 Tool Z Offset:         0.000 mm
   5 Max. XYZ Offset:      30.000 mm
   6      Report on DOUT[0  ]
   7 Max. Orient Offset:   5.000  deg
   8 Max. Alignment Offset:10.000 mm
   9 Deviation Register:   0
  10 Accum. Offset Register: 0
  11 Accum. Offset Enabled:  FALSE
  12              Threshold: 25.000 mm
```

Set each item as desired. For information on the UTool Schedule Setup items, refer to Table 15–15 .

Note The default settings are good for most applications.

When you have finished setting up the UTool schedule, press PREV to return to the *i*RCalibration TCP Set Setup Menu:

```
New TCP  Setup
        1 Program name: TCAL_G1    Group[1]
        2 UTool Number:     1
        3 Calibration Mode: New XYZWPR
        4 UTool Schedule:     1
        5 Detect Schedule:  1
```

11. Select the Detection Schedule and Set Detection Schedule detail items. Move the cursor to Detect Schedule.

1. To change the number of the Detect Schedule, move the cursor to the schedule number, type the number of the schedule you want, and press ENTER.
2. To set up the detection schedule detail items, press F4, SELECT. You will see a screen similar to the following:

```
Detection Schedule
                                        1/18
   Detection Schedule [ 1]
   Sensor Type:               TOS ALL AXES
   I/O Setup
     Detection Port:         DI   [  1]
     Sensor Enable:          DOUT[  1]
   Motion Setup
     Detection Speed:        10.000  mm/sec
     Detection Distance:     50.000  mm
     Return Speed:           50.000  mm/sec
     Return to Start:        TRUE
       Return Distance:      0.000   mm
     Return Term Type:       Fine
     Learn Speed:            10.000  mm/sec
     Learn distance:         5.000   mm
     Air Cut Speed:          100.000 mm/sec
   TOS Setup
     Sensitivity:            10.000
     Torque Threshold:       100
   Circle Setup
     Fit Tolerance:         1.000   mm
     Radius Tolerance:      1.000   mm
     Number of Retry:       0
   Contact Position Register:  0
```

Note The default settings are good for most applications.

Set each item as desired. Move the cursor to the item you want to change and type the value.

Refer to Table 15–16 for information about detection schedule setup detail items.

12. When setup is complete, press F1, DONE. You will see the *i*RCalibration TCP Set main menu with Setup (Done) in its status display.

Note You must press DONE to change the Setup status. If you press PREV to return to the *i*RCalibration TCP Set main menu, the Setup status display will not change.

```
iRCalibration TCP Set
        1 Robot: Group [1]  Device: Sensor
          Calibration Type: New TCP
        2 Setup             (Done    )
        3 Teach Points      (Not Done)
        4 Create Program    (Not Done)
        5 Run: TCAL_G1      (Not Done)
        6 Update Utool[ 0]  (Not Done)
        7 View Result
```

If the teach pendant program already exists when you press DONE, you will see the following screen:

```
Program exists. Skip Teach/Create?
  Press F2 for YES
  Press F3 for NO
```

If the TP program already exists and has been taught, and you want to skip the teach/create step, press F2, YES, and proceed directly to Run. Refer to Procedure 15-16 to run the program.

If the TP program exists but you want to teach it, change it, or use another program, press F3, NO, and continue to Procedure 15-15 .

15.4.3.2 Creating a TCP Calibration Teach Pendant Program

In order to create a new TCP frame with *i*RCalibration TCP Set, you must first create a TCP calibration teach pendant program. You can create the program manually or you can have *i*RCalibration TCP Set create the program for you automatically. When *i*RCalibration TCP Set setup is complete, the *i*RCalibration TCP Set main menu will be displayed with the cursor at Teach Points.

- To have *i*RCalibration TCP Set create the teach pendant program automatically, refer to Procedure 15-15 .

Teaching Points

In order to set a TCP with *i*RCalibration TCP Set, you must first teach certain points to be used in your TCP calibration teach pendant program.

Review the following guidelines before you begin to teach points.

- If the starting TCP is all zero, an estimated TCP will be created and applied when the Initial TCP calibration program is created. The six or seven taught points are also modified to reflect the new estimated TCP. Be very careful that the rotations during program motion do not cause collisions. You might need to insert additional points to accomplish this.
- If the starting TCP is not all zero, then no estimated TCP is used and all program rotations will occur with respect to the starting TCP. Be very careful that the rotations during program motion do not cause collisions. You might need to insert additional points to accomplish this.
- To improve calibration, teach the calibration points carefully to be sure they are centered above the touch plate holes and with the tool shaft oriented perpendicular to the touch plate surface.
- Set the payload accurately, especially when the *i*RCalibration TCP Set reference positions are at extreme points of the robot's reach.

After you have established the *i*RCalibration TCP Set, you can use it in new and existing programs. You can apply *i*RCalibration TCP Set easily to new or existing programs without having to redefine positions as long as the original programs were created with an accurate tool center point.

You will need to teach positions for your program.

New TCP Program

For creating a New TCP, (including Cal-Plate calibration), there are six or seven taught points. Refer to Table 15–17 for a description of the points to be taught.

Note For New TCP, all points except the Orientation point are taught centered on and above the hole in the calibration plate,

Table 15–17. New TCP Program Points

Position	Description
Approach Position 1	This position must be located such that the robot can move to both Center Position 1 and Approach position 2 without collision. Also, the wrist must be able to rotate freely at this location without collision.
Center Position 1	Center Position 1 must be located on the side of the touch plate that is consistent with the"Plate Direction" setup value. This position should be rotated toward one extreme of orientation about the plate axis. This position should be taught centered over the touch plate circle and about one inch above the plate.
Rotate Position 1	Rotate Position 1 must be located on the same side of the touch plate as Center Position 1. This position should be rotated toward the other extreme of orientation about the plate axis, but the total rotation must be less than 180 degrees. This position should be taught centered over the touch plate circle and about one inch above the plate.
Approach Position 2	Approach Position 2 must be located such that the robot can move to both Center Position 2 and Approach position 1 without collision. Also, the wrist must be able to rotate freely at this location without collision.

Table 15–17. New TCP Program Points (Cont'd)

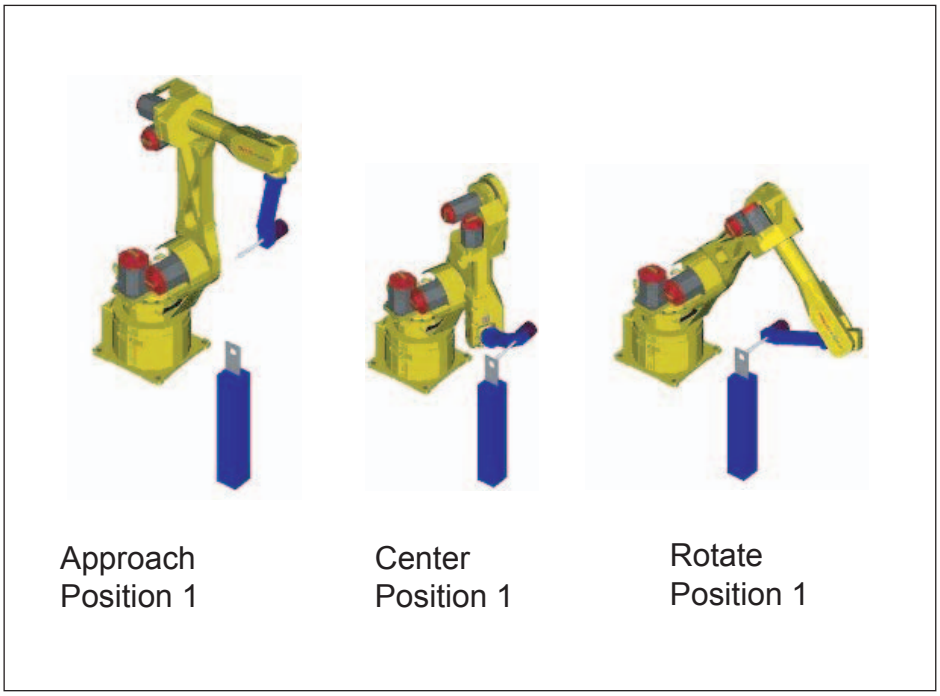
Position	Description
Center Position 2	Center Position 2 must be taught on the opposite side of the plate as Center Position 1. This position should be rotated toward one extreme of orientation about the plate axis. This position should be taught centered over the touch plate circle and about one inch above the plate.
Rotate Position 2	Rotate Position 2 must be located on the same side of the touch plate as Center Position 2. This position should be rotated toward the other extreme of orientation about the plate axis, but the total rotation must be less than 180 degrees. This position should be taught centered over the touch plate circle and about one inch above the plate.
Orientation Position	If you select "New-XYZWPR", you must teach the orientation position with the same orientation as Rotate Position 2 but with the tool shaft extending through the touch plate hole. This position defines the orientation of the tool and +Z will extend out from the tool (such as used by waterjet cutting) if the system variable$CB_VARS.$Z_OUT = TRUE, otherwise +Z will go through the tool (such as used by arc welding) if $CB_VARS.$Z_OUT = FALSE.

Note If "Cal-Plate" is selected, the approach positions should be taught at a pointer where the tool can change orientations to contact both sides of the plate without encountering obstacles. The Center position 1 defines the side of the plate where the TCP will reside. The final TCP will be at the surface of the plate and centered on the hole on the side 1–4 1. where the first contact is made. The position must be taught so the plate is approximately perpendicular to the contact pointer.

For NEW-XYZ or NEW-XYZWPR you must record 7 positions:

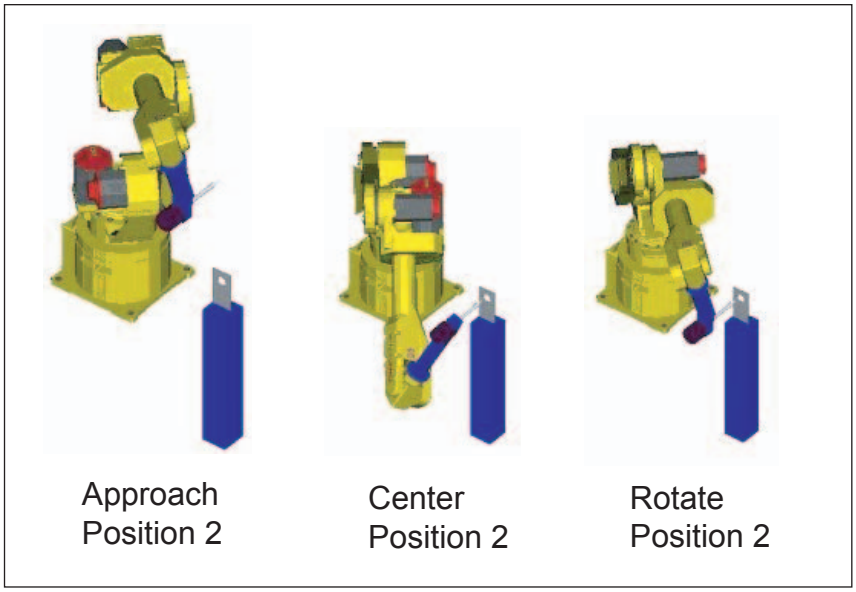
Teach three points on one side of the touch plate: the approach position 1, center position 1, and rotate position 1.

Figure 15–11. Example of First Three Positions



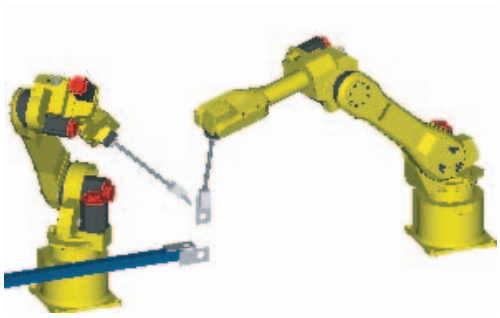
Teach four points on the other side of the touch plate: the approach position 2, the center position 2, the rotate position 2, and the orientation position (not shown in Figure 15–11). The orientation position is taught with the tool shaft extended through the circle.

Figure 15–12. Example of Second Three Positions

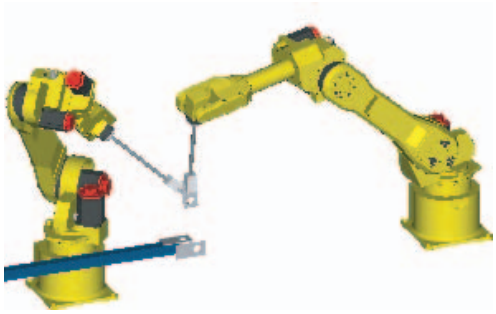


For Cal-Plate calibration, you must teach 6 positions. See Figure 15–13 for an example.

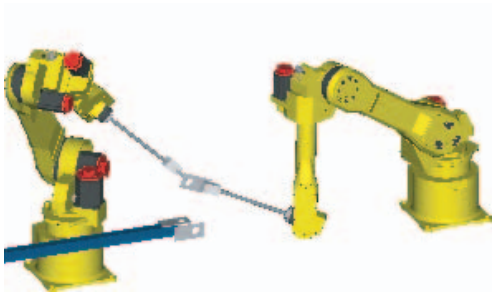
Figure 15–13. Cal-Plate Calibration Positions



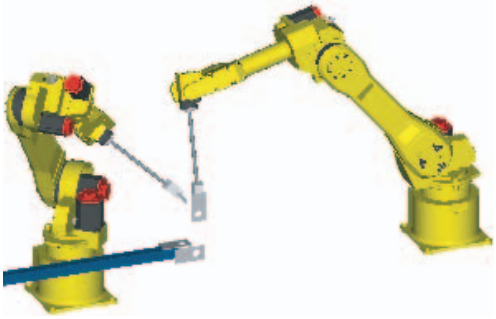
Approach Position 1



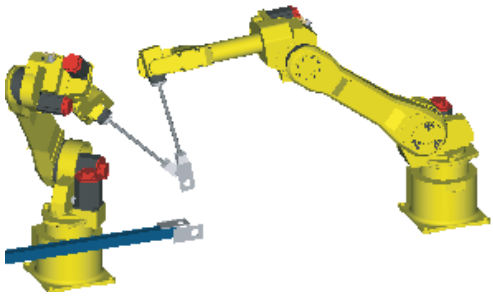
Center Position 1



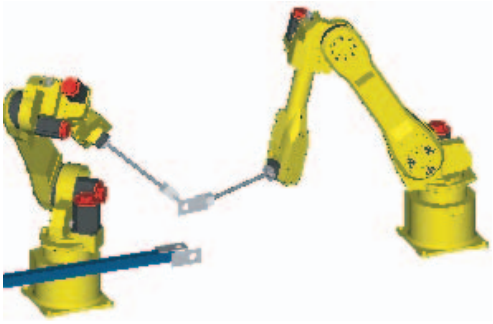
Rotate Position 1



Approach Position 2



Center Position 2



Rotate Position 2

When *i*RCalibration TCP Set setup is complete, the *i*RCalibration TCP Set main menu will be displayed with the cursor at Teach Points.

```
iRCalibration TCP Set
        1 Robot: Group [1]  Device: Sensor
          Calibration Type: New TCP
        2 Setup            New TCP
        3 Teach Points     (Not Done)
        4 Create Program   (Not Done)
        5 Run: TCAL_G1     (Not Done)
        6 Update Utool[ 0] (Not Done)
        7 View Result
```

However, if the teach pendant program already exists, you will see the following screen:

```
Program exists. Skip Teach/Create?
  Press F2 for YES
  Press F3 for NO
```

If you answer YES, the *i*RCalibration TCP Set main menu will be displayed with the cursor on Run and Teach Point and Create Program status set to Done, as shown in the following screen:

```
iRCalibration TCP Set
        1 Robot: Group [1]  Device: Sensor
          Calibration Type: New TCP
        2 Setup            (Done    )
        3 Teach Points     (Done    )
        4 Create Program   (Done    )
        5 Run: TCAL_G1     (Not Done)
        6 Update Utool[ 0] (Not Done)
        7 View Result
```

Procedure 15-15 Teaching Points (Creating the *i*RCalibration TCP Set Teach Pendant Program Automatically)

Conditions

- *i*RCalibration TCP Set Setup has been completed.
- The teach pendant program does not already exist, or you want to overwrite the existing program.

Steps

1. When you have completed *i*RCalibration TCP Set setup, the Setup status on the *i*RCalibration TCP Set main menu will be Done.

```
iRCalibration TCP Set
        1 Robot: Group [1]  Device: Sensor
          Calibration Type: New TCP
        2 Setup             (Done    )
        3 Teach Points      (Not Done)
        4 Create Program    (Not Done)
        5 Run: TCAL_G1      (Not Done)
        6 Update Utool[ 0]  (Not Done)
        7 View Result
```

Move the cursor to Teach Points and press ENTER. You will see a screen similar to the following:

```
New TCP Program Points
         Program Name:         TCAL_G1
         Robot Group:          1
         Calibration Type:     XYZ
         iRCalibration TCP Set Schedule:     1
         Detection Sched:      1
       1 Reference Frame:      WORLD
       2 Plate Direction:      -Z
       3 Plate Thickness:      6.350  mm
       4 Circle XY offset:     30.000  mm
       5 Circle Z offset:      10.000  mm

       6 Approach Position 1: UNINIT
       7 Center Position 1:   UNINIT
       8 Rotate Position 1:   UNINIT
       9 Approach Position 2: UNINIT
      10 Center Position 2:   UNINIT
      11 Rotate Position 2:   UNINIT
      12 Orientation Position:UNINIT
```

2. Move the cursor to Reference Frame and press F4, CHOICE. Set the Reference Frame to UFRAME or WORLD.
3. Move the cursor to Plate Direction and press F4, CHOICE. Select the plate's normal direction. The direction could be X, Y, Z, or -X, -Y, -Z.
4. Move the cursor to Plate Thickness, type the plate's thickness, and press ENTER.
5. Move the cursor to Circle XY offset, type the circular detections X and Y, and press ENTER.

6. Move the cursor to Circle Z offset, type the circular detection Z offset, and press ENTER.
7. To clear all recorded positions, press F1, CLEAR, then F2, YES.
8. Move the cursor to each point to be taught, and teach the point. Refer to Table 15–17 for information about the points.
   - Press F2, LOAD, to load the TCAL_G1 positions to the menu's position data.
   - Press F5, MV_LIN, to move to a position.
9. Press PREV to exit from the Teach Points menu.

```
iRCalibration TCP Set
        1 Robot: Group [1]  Device: Sensor
          Calibration Type: New TCP
        2 Setup            (Done    )
        3 Teach Points     (Done    )
        4 Create Program   (Not Done)
        5 Run: TCAL_G1     (Not Done)
        6 Update Utool[ 0] (Not Done)
        7 View Result
```

If all required positions are taught, the Teach Points status will be Done and the cursor will automatically move to Create Program. Otherwise, the cursor will stay on Teach Points and its status will remain unchanged.

10. To create the program, move the cursor to Create Program and press ENTER. When *i*RCalibration TCP Set has created the program, the Create Program status will change to Done.

Refer to Figure 15–14 for a created TCP calibration program with calibration type set to NEW-XYZWPR.

Figure 15–14. *i*RCalibration TCP Set NEW-XYZWPR Calibration Program Example

```
1:  UFRAME_NUM=1
   2:  UTOOL_NUM=1
   3:  UTool Start[1]
   4:J P[1:Approach pos. 1] 100mm/sec FINE
   5:L P[2:Center 1 outside] 100mm/sec FINE
   6:  Detect Circle[4]( 30.00, 10.00)World[Y]
   7:
   8:L P[3:Rotate 1 outside] 30deg/sec FINE
   9:  Detect Circle[4]( 30.00, 10.00)World[Y]
  10:
  11:L P[4:Center 1 outside] 30deg/sec FINE
  12:L P[5:Approach pos. 1] 100mm/sec FINE
  13:J P[6:Approach pos. 2] 100mm/sec FINE
```

```
14:L P[7:Center 2 outside] 100mm/sec FINE
15:  Detect Circle[4]( 30.00, 10.00)World[-Y]
16:
17:L P[8:Rotate 2 outside] 30deg/sec FINE
18:  Detect Circle[4]( 30.00, 10.00)World[-Y]
19:
20:L P[9:Orient inside] 100mm/sec FINE
21:  Detect Circle[4](  0.00,  0.00)World[-Y]
22:L P[10:Rotate 2 outside] 100mm/sec FINE
23:L P[11:Center 2 outside] 30deg/sec FINE
24:L P[12:Approach pos. 2] 100mm/sec FINE
25:J P[13:Approach pos. 1] 100mm/sec FINE
26:  UTool End [New-XYZWPR]
```

Figure 15–15. *i*RCalibration TCP Set Cal-Plate Calibration Program Example

```
 1:  UFRAME_NUM=0 ;
 2:  UTOOL_NUM=2 ;
 3:  UTool Start[1] ;
 4:J P[1:Approach pos. 1] 100mm/sec FINE    ;
 5:L P[2:Plate position] 100mm/sec FINE    ;
 6:  Detect Point[1] World[-X] ;
 7:L P[3:Plate position] 100mm/sec FINE    ;
 8:  Detect Point[1] World[-X] ;
 9:L P[4:Plate position] 100mm/sec FINE    ;
10:  Detect Point[1] World[-X] ;
11:L P[5:Center 1 outside] 100mm/sec FINE    ;
12:  Detect Circle[1]( 30.00, 10.00)World[-X] ;
13:   ;
14:L P[6:Rotate 1 outside] 30deg/sec FINE    ;
15:  Detect Circle[1]( 30.00, 10.00)World[-X] ;
16:   ;
17:L P[7:Center 1 outside] 30deg/sec FINE    ;
18:L P[8:Approach pos. 1] 100mm/sec FINE    ;
19:J P[9:Approach pos. 2] 100mm/sec FINE    ;
20:L P[10:Center 2 outside] 100mm/sec FINE    ;
21:  Detect Circle[1]( 30.00, 10.00)World[-X] ;
22:   ;
23:L P[11:Rotate 2 outside] 30deg/sec FINE    ;
24:  Detect Circle[1]( 30.00, 10.00)World[-X] ;
25:   ;
26:L P[12:Center 2 outside] 30deg/sec FINE    ;
27:L P[13:Approach pos. 2] 100mm/sec FINE    ;
28:J P[14:Approach pos. 1] 100mm/sec FINE    ;
29:  UTool End [Cal-Plate] ;
```

15.4.4 Running the *i*RCalibration TCP Set Program

In order to get an accurate new TCP with *i*RCalibration TCP Set, you must first create a TCP calibration teach pendant program.

Procedure 15-16 Running the *i*RCalibration TCP Set Program

Conditions

- The *i*RCalibration TCP Set Create Program status is Done.

Steps

1. At the *i*RCalibration TCP Set Main Menu

```
iRCalibration TCP Set
        1 Robot: Group [1]  Device: Sensor
        2 Calibration Type: New TCP
        3 Setup             (Done    )
        4 Teach Points      (Done    )
        5 Create Program    (Done    )
        6 Run: TCAL_G1      (Not Done)
        7 Update Utool[ 0]  (Not Done)
        8 View Result
```

Press ENTER or F4, SELECT, to run the TCP calibration program.

2. You have the following choices:

a. Be sure the teach pendant is ON and all faults are cleared.

b. Press SHIFT+F2 to run the TP program.

After the program is created, it is recommended to test run the *i*RCalibration TCP Set program in low override to make sure there is no collision with any obstacle. You should run *i*RCalibration TCP Set at 100% override to get the new TCP value.

The Run Status will change to Done after the TP program has been run successfully.

```
iRCalibration TCP Set
        1 Robot: Group [1]  Device: Sensor
          Calibration Type: New TCP
        2 Setup             (Done    )
        3 Teach Points      (Done    )
        4 Create Program    (Done    )
        5 Run: TCAL_G1      (Done    )
        6 Update Utool[ 0]  (Not Done)
        7 View Result
```

Updating UTool

The Update UTool menu allows you to update the UTool with the calibration result obtained from running the calibration program.

To update the UTool, move the cursor to Update UTool and press ENTER. If you want to check the TCP calibration result, press F4, SELECT. Refer to Procedure 15-18 .

Procedure 15-17 Updating the UTool

The Update UTool menu is not accessible unless you have completed all the steps before it.

1. At the *i*RCalibration TCP Set main menu, move the cursor to Update UTool and press ENTER. You will see a screen similar to the following:

```
 TCP Calibration Result
       Current TCP [G:1,  1]
         X:   0.018 Y:    0.206 Z:   −0.031
         W:  -0.000 P:   -0.000 R:    0.000
New TCP
        X:   0.028 Y:    0.306 Z:   −0.082
        W:  -0.000 P:   -0.000 R:    0.000
Mean Error:         0.000
Max. Error:         0.000
```

2. Press F2, UPDATE, to update the group 1, UTool frame 1.

Note The UTool is not automatically updated even if its schedule has auto update set to TRUE. This is different from previous version of *i*RCalibration TCP Set.

15.4.5 Displaying the UTool Log Screen

The *i*RCalibration TCP Set view menu allows you to see previous calibration results stored in the UTool log.

The UTool log is a buffer that holds previous tool frame information. The buffer can hold up to 20 items. When the buffer is full, it will replace the oldest data in the buffer with the new logged data. If you want to view frame information, you can display the UTool Log screen. To display the UTool log screen, use Procedure 15-18 .

Procedure 15-18 Displaying the UTool Log Screen

Steps

1. Press MENU.
2. Select UTILITIES.
3. Select *i*RCalibration.
4. Select *i*RCalibration TCP Set. You will see a screen similar to the following.

```
iRCalibration TCP Set
        1 Robot: Group [1]  Device: Sensor
          Calibration Type: New TCP
        2 Setup            (Done    )
        3 Teach Points     (Done    )
        4 Create Program   (Done    )
        5 Run: TCAL_G1     (Done    )
        6 Update Utool[ 0] (Not Done)
        7 View Result
```

5. Move the cursor to View Result and press ENTER.

```
 TCP Calibration Result
       Current TCP [G:1,  1]
         X:   0.018 Y:    0.206 Z:   −0.031
         W:  -0.000 P:   -0.000 R:    0.000
New TCP
        X:   0.028 Y:    0.306 Z:   −0.082
        W:  -0.000 P:   -0.000 R:    0.000
Mean Error:         0.000
Max. Error:         0.000
```

6. To display the UTool Log, press F4, VIEW LOG. You will see a screen similar to the following

```
User Tool Log
                                        11/40
   Date      Time      X(W)      Y(P)      Z(R)
01-JAN-XX    16:57   —164.8        4.2    353.1
 UTool Num [ 1]          2.0     -70.6     -4.8
01-JAN-XX    16:57    -164.8       4.2    353.1
 UTool Num [ 1]          2.0        3.1    -81.4
01-JAN-XX    16:57   -166.1        4.2    350.3
 UTool Num [ 1]          2.1     —70.6     -4.9
No XYZ Data
No WPR Data
No XYZ Data
```

Each entry spans two lines. The first line lists the date and time of the saved entry as well as x, y, and z information. The second line contains the frame number and w, p, and r information.

If entries are not used or if they have been cleared, you will see the following lines:

No XYZ Data

No WPR Data

7. Select one of the following:

- **If you want to save the entire log to the file "TOOL_LOG.DT,"** press F2, SAVE.
- **If you want to restore a previous UTool value,** move the cursor to the line that has the values you want to restore and press F3, APPLY. You will see a prompt box similar to the following.

```
Restore old Frame?

Press F2 for YES
Press F3 for NO
```

Move the cursor to YES and press ENTER.

- **If you want to clear the log and remove all data entries,** press F4, CLEAR. You will see a screen similar to the following.

```
Clear All Logged Data?

  Press F2 for YES
  Press F3 for NO
```

Move the cursor to YES and press ENTER.

15.4.6 *i*RCalibration TCP Set Error Recovery

If a *i*RCalibration TCP Set error occurs during TCP calibration, the robot goes back to the starting position and an error message is displayed. Review the error message, correct the problem, then restart the TCP calibration program.

15.4.6.1 Redoing a *i*RCalibration TCP Set Adjustment

To redo a *i*RCalibration TCP Set adjustment and restart *i*RCalibration TCP Set,

1. Review the error message for an indication of the problem.
2. Correct the problem.
3. Abort any program that is paused.
4. Enter the *i*RCalibration TCP Set program name in the setup screen.
5. Press DONE to exit.
6. When *i*RCalibration TCP Set asks "Program exists, skip Teach/Create?", reply YES.
7. Run the *i*RCalibration TCP Set program: move the cursor to Run and press ENTER.

15.4.6.2 *i*RCalibration TCP Set Fit Errors

Refer to Table 15–18 for information on troubleshooting *i*RCalibration TCP Set fit errors.

Table 15–18. *i*RCalibration TCP Set Fit Errors

Try This	Description
Verify that the tool is rigid.	Worn or damaged tooling components can cause variation in tool repeatability and can cause fit errors.
Verify the payload setting.	If the reference position is at an extreme position, improper payload settings can cause increased robot vibration, which affects *i*RCalibration TCP Set repeatability.
Check for a loose touch plate. *	If the touch plate becomes loosened due to improper installation, *i*RCalibration TCP Set repeatability will be affected.
Reduce motion speed. *	Under certain conditions, the default motion speed can cause too much vibration. For best performance, *i*RCalibration TCP Set uses increased acceleration values. Try lowering the motion speed from the default 100mm/sec to 25 or 50mm/sec.

Table 15–18. *i*RCalibration TCP Set Fit Errors (Cont'd)

Reduce search speed. *	Under certain conditions, the default motion speed can cause too much vibration. For best performance, *i*RCalibration TCP Set uses increased acceleration values. Try lowering the search speed from the default 10mm/sec to 5mm/sec.
Increase the retry count. *	*i*RCalibration TCP Set has significant safeguards against getting incorrect TCP corrections. You can increase the retry count from the default of 0 to either 3 or 4 to reduce the occurrence of *i*RCalibration TCP Set fit errors.
Increase the radius tolerance. *	The default radius tolerance of .5mm may not be large enough to allow for the normal variation of some tooling. You can try increasing the radius tolerance from .5mm to 1.0mm or more as your application allows.
Increase the fit tolerance.*	Most *i*RCalibration TCP Set errors are detected by radius tolerance, but if increasing the radius tolerance does not resolve the problem, you can increase the fit tolerance from .25mm to .5mm or more as your application allows.
If you are using TOS, increase the sensitivity tolerance value. *	Normally, the default value of 10 is adequate; however, you can increase this value to either 15 or 20.
If you are using TOS, increase the torque threshold. *	Normally, the default value of 100 is adequate. However, if the sensitivity tolerance increases have no effect, you can increase this value to 150.
Check for poor electrical continuity	If the I/O method is used, and if it relies on electrical continuity, then dirty contact surfaces on either the tool or the calibration plate will result in increased fit errors.

* Redo *i*RCalibration TCP Set mastering after making this change.

Refer to Section 15.4.4 .

15.4.7 *i*RCalibration TCP Set Troubleshooting

To troubleshoot *i*RCalibration TCP Set, you can

- Display the UTOOL Log screen. Refer to Section 15.4.5
- Run *i*RCalibration TCP Set to check the TCP. Refer to Section 15.4.4 .

15.5 iRCALIBRATION TCP SHIFT

15.5.1 Overview

Note *i*RCalibration TCP Shift is not available for all software application Tools.

The *i*RCalibration TCP Shift option, formerly TCPCal, provides a cost effective, easy-to-use solution to recover the Tool Center Point (TCP) automatically.

When you use *i*RCalibration TCP Shift you start with a fully functional system and an unknown or inaccurate TCP. You then set up *i*RCalibration TCP Shift to generate an accurate TCP.

*i*RCalibration TCP Shift works with any robot and tool where the TCP is at the end of cylindrically-shaped tooling. If the normal tooling is not cylindrical, or if the TCP is away from the tooling, a teach tip can be installed.

To use *i*RCalibration TCP Shift, you must do the following:

1. Install the *i*RCalibration TCP Shift software option and any required hardware.
2. Set up *i*RCalibration TCP Shift.
3. Create a TCP calibration TP program.
4. Run *i*RCalibration TCP Shift to recover the current TCP.

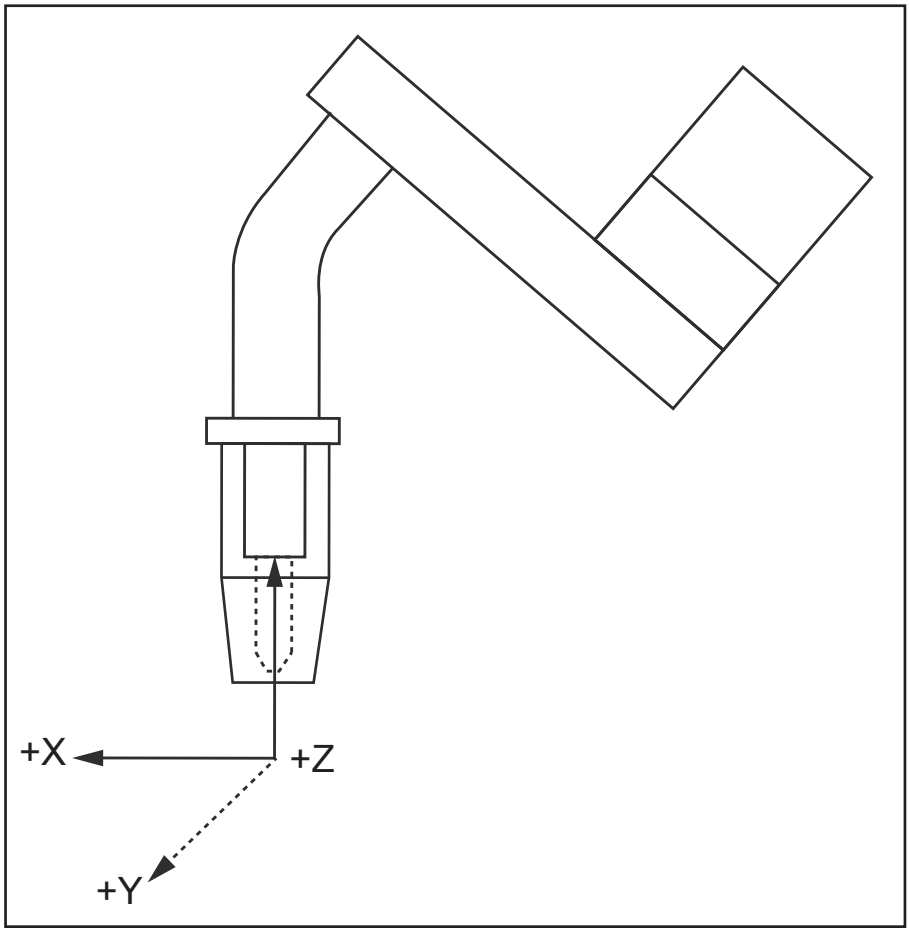
15.5.2 Preliminary Setup

Before you can set up and use *i*RCalibration TCP Shift you must do the following:

- install the *i*RCalibration TCP Shift software option. Refer to the *Software Installation Manual* for more information about installing software options.
- Install the touch plate on a stationary object in the robot workcell. Refer to Section 15.4.2.1

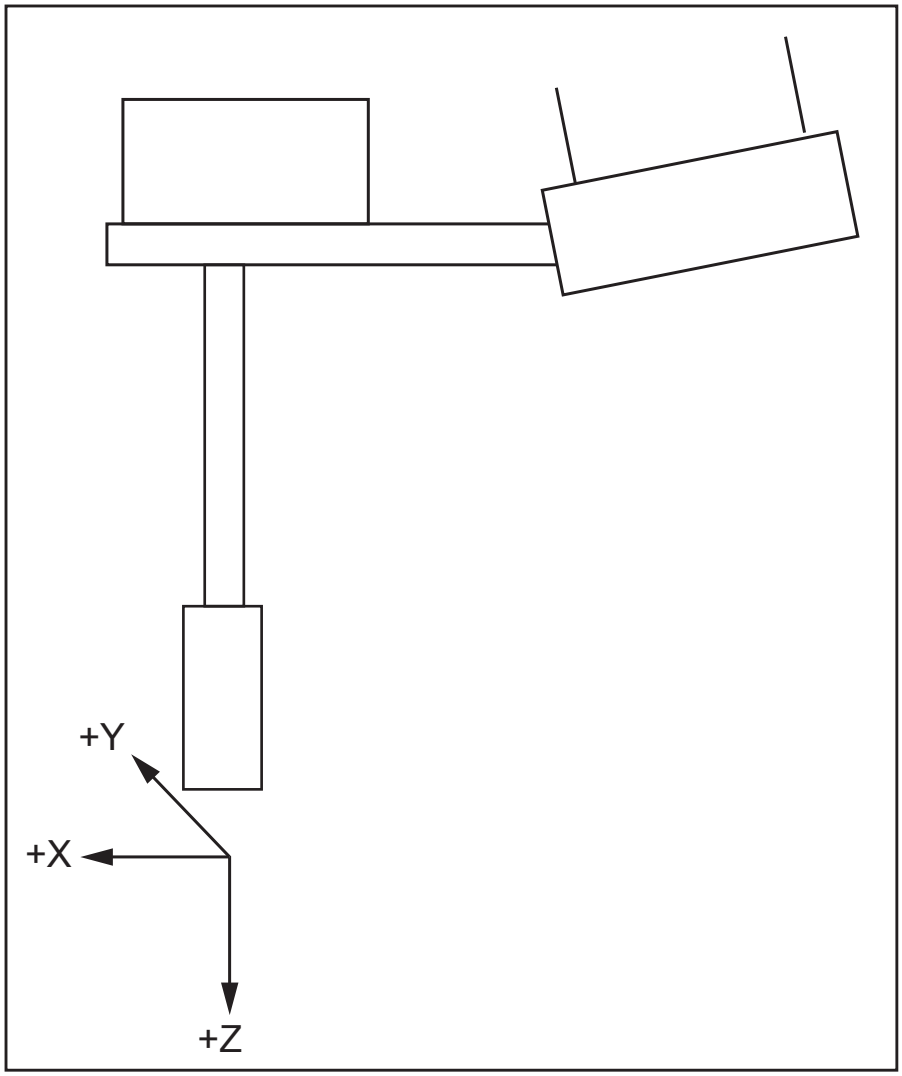
You will then use the *i*RCalibration TCP Shift software to recover the current tool center point. The following figures show typical arc welding and waterjet tool definition.

Figure 15–16. Typical Arc Welding tool definition



Typical Arc Welding tool definition $CB_VARS.$Z_OUT = FALSE, set automatically on initial start for ArcTool.

Figure 15–17. Typical Waterjet Tool Definition

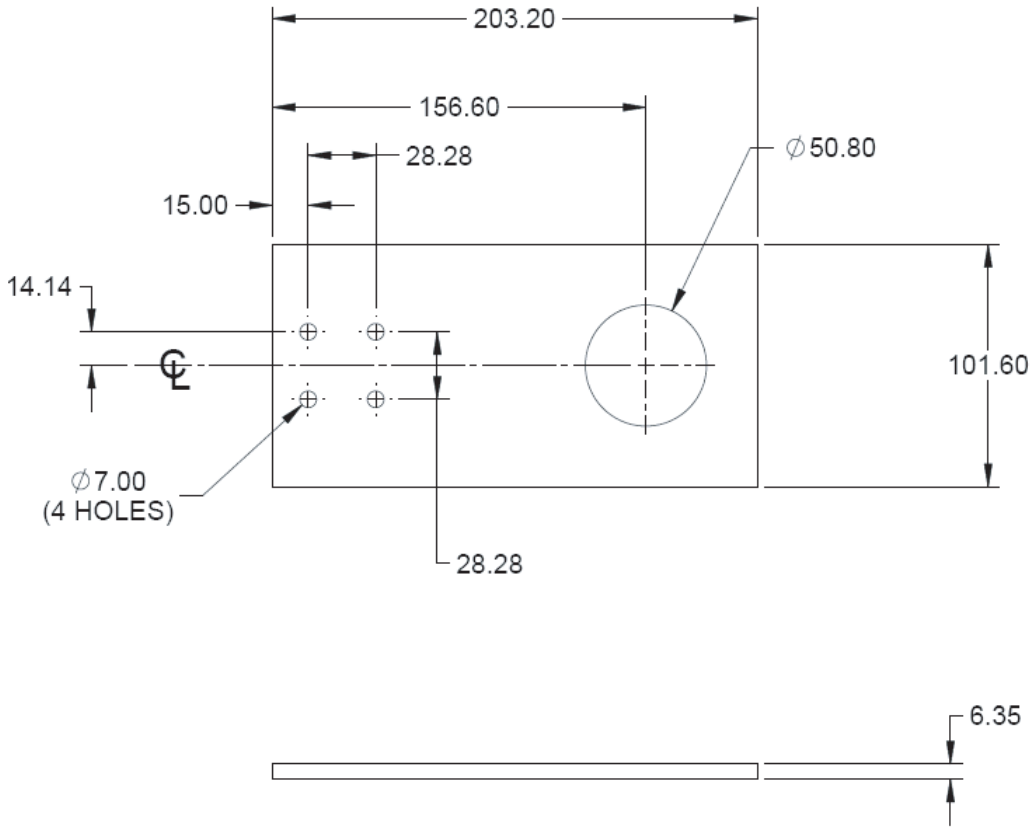


Typical Waterjet Tool Definition, $CB_VARS.$Z_OUT = TRUE, default for all applications except ArcTool

15.5.2.1 Touch Plate Installation

Before you can set up and run *i*RCalibration TCP Shift, you must install a touch plate on a stationary object for pointer type tools, within the robot workcell. See Figure 15–18 for an illustration of a recommended touch plate and pointer.

Figure 15–18. Recommended *i*RCalibration TCP Shift Touch Plate and Pointer Specifications



Use Procedure 15-19 to install and align the touch plate.

Procedure 15-19 Installing and Aligning the Touch Plate

Conditions

- The TOOL frame has been set up correctly.

  Refer to “Frames Setup” in the application-specific *Setup and Operations Manual* for more information.

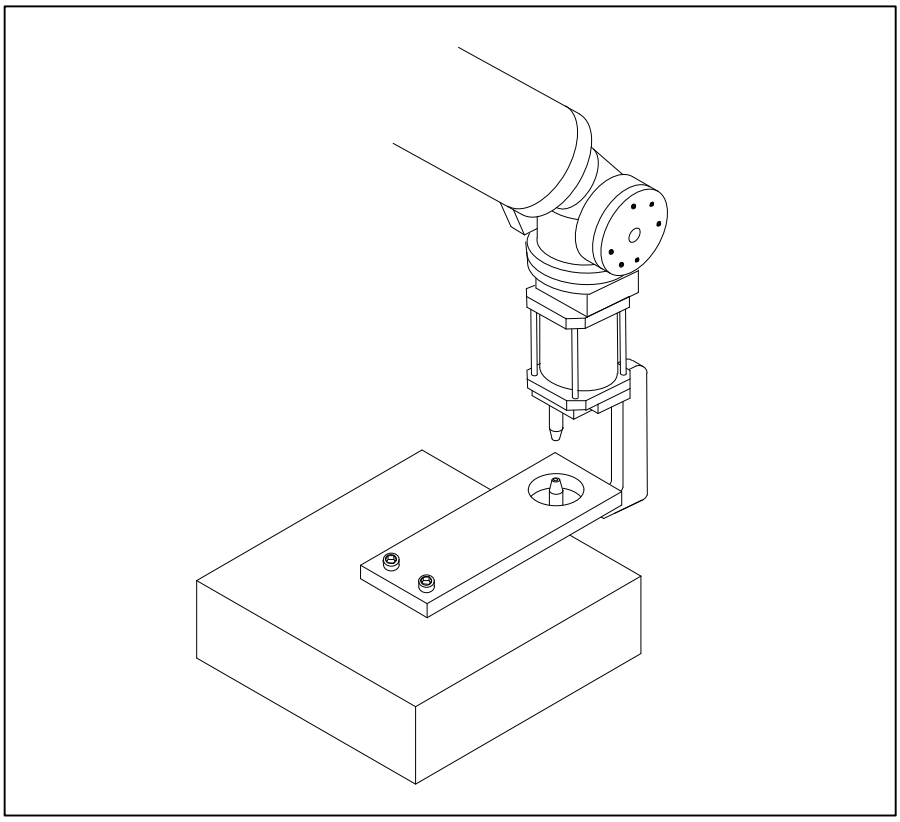
Steps

1. Install the touch plate (touch pointer for gripper) on a flat level surface well within the workcell. See Figure 15–19 .

   The touch plate or touch pointer should be securely fastened to a firm surface. If the touch plate position changes due to collisions or other events, the recovered TCP will be incorrect.

There should be ample clearance below the plate so that the tooling can be inserted through the circular opening in the touch plate.

Figure 15–19. Installing the Touch Plate



2. In most applications, the typical TCP definition is that the TOOL z axis is along the shaft used by *i*RCalibration TCP Shift. For arc welding applications, the TOOL +z direction extends from the TCP location into the torch body. For other applications, the TOOL +z direction extends away from the shaft at the TCP location. *i*RCalibration TCP Shift works with any TOOL frame definition.

   Installation and alignment of the touch plate is now complete.

   Note If you are using ArcTool and require compatibility with TorchMate, refer to the "TorchMate" chapter in the *ArcTool Setup and Operations Manual* for further information. *i*RCalibration TCP Shift can coexist with TorchMate.

15.5.3 *i*RCalibration TCP Shift Setup

Before you can use *i*RCalibration TCP Shift to recover the TCP, you must perform some setup procedures. Setup involves the initial setting of menu options within *i*RCalibration TCP Shift software.

Setting up and running *i*RCalibration TCP Shift includes the following processes:

- Selecting the motion group you want to calibrate
- Selecting the teach pendant program, the UTool you want to calibrate, and the calibration mode: XYZWPR, XYZ-align, XYZ, XY, Z, Gripper. Refer to Table 15–19 for details of the calibration modes.
- Setting up the UTool schedule. Refer to Table 15–20 for a description of the items on the UTool Schedule Detail Screen.
- Setting up the detection schedule. Refer to Table 15–21 for a description of the items on the *i*RCalibration TCP Shift Detection Schedule Detail Screen.
- Teaching points for the teach pendant calibration program. Refer to Procedure 15-21 .
- Creating the teach pendant calibration program from the taught points and running the calibration program. You can have *i*RCalibration TCP Shift create the program for you automatically (refer to Procedure 15-21 .
- Updating the UTool with the calibration results obtained from running the calibration program. Refer to Procedure 15-23 .

Table 15–19. Calibration Mode Descriptions

Calibration Mode	Description
XYZWPR	Recover TCP xyzwpr
XYZ_Align	Recover tool XYZ and alignment for spot guns.
XYZ	Recover Tool XYZ
XY	Recover tool xy
Z	Recover tool z
Gripper	Recover gripper location

Table 15–20. *i*RCalibration TCP Shift UTOOL Schedule Item Description

ITEM	DESCRIPTION
Auto Update Default: TRUE	This item indicates whether *i*RCalibration TCP Shift's UTOOL END instruction will update the corresponding $mnutool or not. If you set Auto Update to TRUE and run the TCP calibration program, the $mnutool will be updated based on the offset.
Store Offset Default: FALSE	This item indicates whether or not *i*RCalibration TCP Shift's UTOOL END instruction will update the position register with the TCP offset.

Table 15–20. *i*RCalibration TCP Shift UTOOL Schedule Item Description (Cont'd)

Tool Z Offset Units: mm Range: – 500.000– 500.000 Default: 0	This item is the distance from the point of the tooling that contacts the plate to the actual TCP. This is used when XYZWPR mode is chosen to adjust the XYZ correction at the TCP properly when there is a tool Z change. For spot welding and other applications where the contact point is the actual TCP location, this value should be zero. For Waterjet, arc welding, and other applications where the contact point is above the TCP at the time of contact, this value should be set to the distance from the contact point to the actual TCP.
Max. XYZ Offset Units: mm Range: 0.1 — 1000 Default:	This item is the distance in mm from the previous TCP to the new TCP for a single correction. If a single correction is larger than the max. XYZ offset, an error message will be posted. If the out DOUT port is not zero, *i*RCalibration TCP Shift will also set the port to ON.
Max. Orent Offset Units: deg Range: 0.01 — –20 Default:	This item is the orientation change in degrees from the previous TCP to the new TCP for a single correction. If a single correction is larger than the max. orientation offset, an error will be posted.
Max. Alignment Offset Units: mm Range: 0 — 500 Default: 10	This item is the maximum alignment change allowed in mm from the previous TCP to the new TCP. If the alignment change is greater than the alignment offset, an error will be posted. This item is in effect when the XYZ-Align calibration mode is selected.
Deviation Register Range: 0 — 999 Default: 0	This item indicates the register number where *i*RCalibration TCP Shift will update with the change between the previous TCP and the new TCP. If this item is 0, no update is performed.
Accum. Register Range: 0 — 999 Default:	This item is the change between the new TCP and the original mastered TCP. If this item is not 0, *i*RCalibration TCP Shift will update the register with the accumulated deviation value.
Accumulated Error Enable Default: FALSE	If this item is TRUE, *i*RCalibration TCP Shift will check the accumulated deviation against the accumulated error threshold. If the deviation is greater than 0, an error will be posted.
Accumulated Offset Threshold Default: 25 Min = 0 Max = 500mm	When the Accumulated Offset Threshold is exceeded and the function is enabled, *i*RCalibration TCP Shift will post and ACAL-060, Excessive Accumulated Offset warning.

When you set up a detection schedule, you

- determine the sensor

- determine the detection motion
- determine TOS

Note Default values are usually acceptable.

Table 15–21. *i*RCalibration TCP Shift Detection Schedule Description

ITEM	DESCRIPTION
Sensor Type Values: TOS Wrist, I/O, TOS ALL AXES Default: TOS ALL AXES	This item indicates the kind of sensor used for contact detection. The three modes are: • **TOS Wrist** (Torque Observer Sensor on Wrist) - the contact torque will be monitored, and only the observed torque on the wrist will be used to determine the contact position. When TOS Wrist is selected, no I/O connection is required for contact detection. TOS Wrist is very sensitive and can be more prone to false detections in certain cases than the more robust TOS All Axes method. TOS Wrist is not recommended except for special cases where it can verified that false detections will not occur. • **I/O** - the specified I/O will be monitored and only the state of this input will be used to determine the contact position. Use the detection port for contact detection with I/O. Refer to Detection Port item in this table for more information. • **TOS ALL AXES** (Torque Observer Sensor, All Axes) - The contact torque will be monitored on all robot axes, and only the observed torque will be used to determine the contact position. This is the default value. When TOS ALL AXES is selected, no I/O connection is required for contact detection.
Detection Port Values: RDI, DI, WDI, WSI Note: AIN is not supported in *i*RCalibration TCP Shift.	This item indicates the desired input port type. The port types are: • **RDI** - Robot Digital Input • **DI** - normal Digital Input • **WDI** - Weld Digital Input • **WSI** - Wire Stick circuit Input. The port type that you specify is followed by the desired port number that will go from OFF to ON when contact is detected. The WDI and WSI ports are typically used for arc welding. You will set this item up if you selected I/O as the Sensor Type. You do not need to set this up if you selected TOS Wrist or TOS All Axes as the Sensor Type.

Table 15–21. *i*RCalibration TCP Shift Detection Schedule Description (Cont'd)

Sensor Enable Values: RDO, DOUT, WDO, or WSO Default: RDO	This item indicates the I/O sensor enable digital output port type. The port types are: • **RDO** - Robot Digital Output • **DOUT** - normal Digital Output • **WDO** - Weld Digital Output • **WSO** - Wire Stick circuit Output. The port type that you specify is followed by the desired port number that will be used to enable the I/O detection circuit. The WDO and WSO ports are typically used for arc welding. You will set this item up if you selected I/O as the Sensor Type. Set the port number to 0 (zero) if you are not using this feature.
Detection Speed Units: mm/sec Range: 1.0 - 100 Default: 10	This item is the search speed for contact detection. This speed is used for both TOS and I/O searches. The value can be increased if cycle time is of great concern (although increasing the speed will also affect the resultant accuracy and repeatability) , or reduced if vibration or other disturbances cause *i*RCalibration TCP Shift to retry excessively.
Detection Distance Units: mm Range: 1 - 1000 Default: 50	This item is the maximum distance that the robot will move while looking for the signal.
Return Speed Units: mm/sec Range: 5 - 5000 Default: 50	This item is the search return speed. This speed is used for both TOS and I/O searches.
Return to Start Default: Yes	When this item is set to YES, the robot will move back to the position before it starts the search motion. When this item is set to NO, the robot will move back to a distance specified by the next item (Return distance)
Return Distance Units: mm Range: 1 - 1000 Default: 50	When return to start is set to FALSE, the robot will move back to the distance specified. This function applies to DETECT POINT instructions only.
Return term type Default: FINE	This item is the termination type for return motion.

Table 15–21. *i*RCalibration TCP Shift Detection Schedule Description (Cont'd)

Air Cut Speed Units: mm/sec Range: 1 - 1000 Default: 100	This item is the non-search motion speed.
Learn Distance Units: mm Range: 0.5– 20.0 Default: 5.0	For "TOS All Axes" method only, there is a short motion before each search to identify dynamic torque values. Learn Distance is the distance in millimeters of this short motion, the direction is determined at the time of the motion.
Learn Speed Units: mm/sec Range: 0.5 – 50.0 Default: 10.0	For "TOS All Axes" method only, this is the speed of the learning motion before each search.
Sensitivity Tolerance Units: max. motor torque/sec. Range: 1 - 999 Default: 10	This item is the primary sensitivity tolerance of TOS Wrist and is not used for TOS All-Axes. It is the rate of change of the disturbance torque. The actual units vary with motor model, gear ratio, and amplifier current, and are affected by tooling length. However, the effect for calibration is largely independent of these factors and a value of 10 is considered normal operation for typical installations. You should not use a value of less than 10 or misdetections might occur. This value can be increased if false detections occur frequently. **Warning** **Be careful when you change this number. Changing the sensitivity tolerance might cause a false detection. This could injure personnel or damage equipment.**
Torque Threshold Units: max. motor torque Range: 1 to 999 Default: 100	For **TOS All Axes** , this item is the only measure of contact detection. During the learning motion a torque level is memorized and that level plus **Torque Threshold** must be exceeded for contact to be determined. If false detections are being experienced then this value should be increased. **For TOS Wrist** , this item is the level of disturbance torque that is used as another means for contact detection. The default value of 100 is considered normal operation for typical installations. If increasing the value of **Sensitivity Tolerance** does not make false detections disappear, then this value should be increased until the false detections disappear; then, reset the value of **Sensitivity Tolerance**.
Fit Tolerance Units: mm Range: 0.1 - 100 Default: 0.5	For Detect Circle instruction: The robot detects 4 points inside the circle and the calibration software will fit a circle to the 4 detected positions. The fit error is defined as the maximum distance from the 4 points to the circle. If the fitting error exceeds the tolerance, the calibration software will post an error.

Table 15–21. *i*RCalibration TCP Shift Detection Schedule Description (Cont'd)

Radius Tolerance Units: mm Range: 0.1— 100 Default: 0.5	The Detect Circle instruction stores the circle's radius during mastering. At run time, if the computed circle radius differs from the stored radius value and difference exceeds the tolerance, the calibration software will post an error.
Number of Retry Range: 0 - 9 Default: 0	If the number of retry is >0, the calibration software will re-run the detect circle instruction if the current circle detection has either a fit error or a radius error.
Contact Position Register Range 0-100 Default 0	If the number is non-zero. *i*RCalibration TCP Shift records the contact position to the specified position register.

15.5.3.1 Setting Up *i*RCalibration TCP Shift

*i*RCalibration TCP Shift uses a wizard-style menu. It shows the steps that need be done to complete the TCP calibration process. Each step of the process has its own status display that shows where you are in the setup process. You cannot skip steps and you cannot go to the next step without completing the current step.

*i*RCalibration TCP Shift also includes a menu item, View Result, that lets you review the previous calibration result and log history.

Procedure 15-20 Setting Up *i*RCalibration TCP Shift

Conditions

- The *i*RCalibration TCP Shift software option is loaded.
- Preliminary setup is complete. Refer to Section 15.4.2 .

Steps

1. Press MENUS.
2. Select UTILITY.
3. Press F1, [TYPE].
4. Select *i*RCalibration. You will see a screen similar to the following:

```
UTILITIES iRCalibration
   1 TCP Shift
   2 TCP Set
   3 Frame Shift
   4 Frame Set
   5 Master Recovery
   6 Master Set
```

5. Select *i*RCalibration TCP Shift and press ENTER. You will see a screen similar to the following:

```
iRCalibration TCP Shift
        1 Robot: Group [1]  Device: Sensor
          Calibration Type: TCP Recovery
        2 Setup            (Not Done)
        3 Teach Points     (Not Done)
        4 Create Program   (Not Done)
        5 Run: TCAL_G1     (Not Done)
        6 Update UTool[ 1](Not Done)
        7 View Result
```

6. Move the cursor to Robot: Group [1], type the group number of the motion group you want to calibrate, and press ENTER.

Note Device is always Sensor. You cannot change it.

7. **Specify the Calibration Program Name**. The *i*RCalibration TCP Shift Setup menu allows you to specify the TCP calibration TP program name, the UTool number, and the calibration mode, and select the *i*RCalibration TCP Shift calibration schedule and the detection schedule.

Note The Group number is provided for reference. You cannot change it in this menu.

a. Move the cursor to Setup and press ENTER or F4, SELECT. You will see a screen similar to the following:

```
TCP Recovery Setup
        1 Program name: TCAL_G1    Group[1]
        2 UTool Number:     1
        3 Calibration Mode: XYZWPR
        4 UTool Schedule:     1
        5 Detect Schedule:  1
```

b. The default program name is TCAL_Gx, where x is the group number. If you want to specify a different program name, press ENTER and use the function keys.

8. **Select the UTool.** Move the cursor to UTool Number:, type the number of the Tool Frame you want to calibrate, and press ENTER.

9. **Select the Calibration Mode**. Move the cursor to Calibration Mode. Press F4, [CHOICE] and make your selection.

- The calibration mode choices are XYZWPR, XYZ_Align, XYZ, XY, Z and Gripper.

Note The TOOL frame is relative to the touch plate, and the TOOL frame z axis corresponds to the touch plate z axis. Because the touch plate is always aligned with its own x-y axis, regardless of the physical mounting of the plate in WORLD space, the WORLD y-z axis and the touch plate z axis both align with the WORLD x axis. It might be helpful to think in terms of correcting for the TOOL axes that correspond to PLATE xy, PLATE xyz, PLATE xyzwpr, and PLATE

Refer to Table 15–14 for more information.

10. **Select the UTool Schedule and Set UTool Schedule detail items.** Move the cursor to UTool Schedule. Refer to Table 15–15 for information on the UTool Schedule items.

1. To change the number of the TCP, move the cursor to the schedule number, type the number of the UTool Schedule you want, and press ENTER.
2. To see the detail screen for the selected UTool, press ENTER. You will see a screen similar to the following:

```
UTool Schedule                  1/12
   Schedule Number[  1]
   1 Auto Upate:             TRUE
   2 Store Offset: FALSE
   3               In PR[0  ]
   4 Tool Z Offset:          0.000 mm
   5 Max. XYZ Offset:       30.000 mm
   6      Report on DOUT[0  ]
   7 Max. Orient Offset:    5.000  deg
   8 Max. Alignment Offset:10.000 mm
   9 Deviation Register:    0
  10 Accum. Offset Register: 0
  11 Accum. Offset Enabled:  FALSE
  12              Threshold: 25.000 mm
```

Set each item as desired. For information on the UTool Schedule Setup items, refer to Table 15–15 .

Note The default settings are good for most applications.

When you have finished setting up the UTool schedule, press PREV to return to the *i*RCalibration TCP Shift Setup Menu:

```
TCP Recovery Setup
        1 Program name: TCAL_G1    Group[1]
        2 UTool Number:     1
        3 Calibration Mode: XYZWPR
        4 UTool Schedule:     1
        5 Detect Schedule:  1
```

11. Select the Detection Schedule and Set Detection Schedule detail items. Move the cursor to Detect Schedule.

1. To change the number of the Detect Schedule, move the cursor to the schedule number, type the number of the schedule you want, and press ENTER.
2. To set up the detection schedule detail items, press F4, SELECT. You will see a screen similar to the following:

```
Detection Schedule
                                      1/18
   Detection Schedule [ 1]
   Sensor Type:               TOS ALL AXES
   I/O Setup
     Detection Port:         DI  [  1]
     Sensor Enable:          DOUT[  1]
   Motion Setup
     Detection Speed:        10.000  mm/sec
     Detection Distance:     50.000  mm
     Return Speed:           50.000  mm/sec
     Return to Start:        TRUE
       Return Distance:      0.000   mm
     Return Term Type:       Fine
     Learn Speed:            10.000  mm/sec
     Learn distance:         5.000   mm
     Air Cut Speed:          100.000 mm/sec
   TOS Setup
     Sensitivity:            10.000
     Torque Threshold:       100
   Circle Setup
     Fit Tolerance:         1.000   mm
     Radius Tolerance:      1.000   mm
     Number of Retry:       0
   Contact Position Register:  0
```

Note The default settings are good for most applications.

Set each item as desired. Move the cursor to the item you want to change and type the value.

Refer to Table 15–21 for information about detection schedule setup detail items.

12. When setup is complete, press F1, DONE. You will see the *i*RCalibration TCP Shift main menu with Setup (Done) in its status display.

Note You must press DONE to change the Setup status. If you press PREV to return to the *i*RCalibration TCP Shift main menu, the Setup status display will not change.

```
iRCalibration TCP Shift
        1 Robot: Group [1]  Device: Sensor
          Calibration Type: TCP Recovery
        2 Setup            (Done    )
        3 Teach Points     (Not Done)
        4 Create Program   (Not Done)
        5 Run: TCAL_G1     (Not Done)
        6 Update Utool[ 0] (Not Done)
        7 View Result
```

If the teach pendant program already exists when you press DONE, you will see the following screen:

```
Program exists. Skip Teach/Create?
  Press F2 for YES
  Press F3 for NO
```

If the TP program already exists and has been taught, and you want to skip the teach/create step, press F2, YES, and proceed directly to Run. Refer to Procedure 15-22 to run the program.

If the TP program exists but you want to teach it, change it, or use another program, press F3, NO, and continue to Procedure 15-21 .

15.5.3.2 Creating a TCP Calibration Teach Pendant Program

In order to master a TCP with *i*RCalibration TCP Shift, you must first create a TCP calibration teach pendant program. You can have *i*RCalibration TCP Shift create the program for you automatically.

When *i*RCalibration TCP Shift setup is complete, the *i*RCalibration TCP Shift main menu will be displayed with the cursor at Teach Points.

- To have *i*RCalibration TCP Shift create the teach pendant program automatically, refer to Procedure 15-21 .

Teaching Points

In order to recover a TCP with *i*RCalibration TCP Shift, you must first teach certain points to be used in your TCP calibration teach pendant program.

Note If the calibration type is Gripper, you have to create the TCP calibration program,. The *i*RCalibration TCP Shift software can not create a gripper recovery program

TCP Recovery Program

For creating a TCP Recovery Program, there are four taught points. Refer to Table 15–22 for a description of the points to be taught.

Table 15–22. New TCP Program Points

Position	Description
Circle Approach	This position must be located such that the robot can move to both Center Position 1 and Approach position 2 without collision. Also, the wrist must be able to rotate freely at this location without collision.
Center Position 1	Center Position 1 must be located on the side of the touch plate that is consistent with the“Plate Direction” setup value. This position should be taught centered over the touch plate circle and about one inch above the plate.
Center Position 2	This position is for XYZWPR recovery only. It should be taught inside the hole of the touch plate.
Plate Position	The plate position should be taught above the plate and outside of the hole of the touch plate so the TCP will touch the plate when robot is moving in the plate direction.

Teach four points on the other side of the touch plate: the approach position 2, the center position 2, the rotate position 2, and the orientation position (not shown in Figure 15–12). The orientation position is taught with the tool shaft extended through the circle.

For Cal-Plate calibration, you must teach 6 positions. See Figure 15–13 for an example.

When *i*RCalibration TCP Shift setup is complete, the *i*RCalibration TCP Shift main menu will be displayed with the cursor at Teach Points.

```
iRCalibration TCP Shift
        1 Robot: Group [1]  Device: Sensor
          Calibration Type: TCP Recovery
        2 Setup            (Done    )
        3 Teach Points     (Not Done)
        4 Create Program   (Not Done)
        5 Run: TCAL_G1     (Not Done)
        6 Update Utool[ 0] (Not Done)
        7 View Result
```

However, if the teach pendant program already exists, you will see the following screen:

```
Program exists. Skip Teach/Create?
  Press F2 for YES
  Press F3 for NO
```

If you answer YES, the *i*RCalibration TCP Shift main menu will be displayed with the cursor on Run and Teach Point and Create Program status set to Done, as shown in the following screen:

```
iRCalibration TCP Shift
        1 Robot: Group [1]  Device: Sensor
          Calibration Type: New TCP
        2 Setup            (Done    )
        3 Teach Points     (Done    )
        4 Create Program   (Done    )
        5 Run: TCAL_G1     (Not Done)
        6 Update Utool[ 0] (Not Done)
        7 View Result
```

Procedure 15-21 Teaching Points (Creating the *i*RCalibration TCP Shift Teach Pendant Program Automatically)

Conditions

- *i*RCalibration TCP Shift Setup has been completed.
- The teach pendant program does not already exist, or you want to overwrite the existing program.

Steps

1. When you have completed *i*RCalibration TCP Shift setup, the Setup status on the *i*RCalibration TCP Shift main menu will be Done.

```
iRCalibration TCP Shift
        1 Robot: Group [1]  Device: Sensor
          Calibration Type: TCP Recovery
        2 Setup            (Done    )
        3 Teach Points     (Not Done)
        4 Create Program   (Not Done)
        5 Run: TCAL_G1     (Not Done)
        6 Update Utool[ 0] (Not Done)
        7 View Result
```

Move the cursor to Teach Points and press ENTER. You will see a screen similar to the following:

```
 TCP Program Points
      Program Name:        TCAL_G1
      Robot Group:         1
      Calibration Type:    XYZ
      iRCalibration TCP Shift Schedule:     1
      Detection Sched:     1
    1 Reference Frame:     WORLD
    2 Plate Direction:     -Z
    3 Plate Thickness:     6.350  mm
    4 Circle XY offset:    30.000  mm
    5 Circle Z offset:     10.000  mm

    6 Circle Approach: UNINIT
    7 Center Position 1:   UNINIT
    8 Center Position 2:   UNINIT
    9 Plate Position: UNINIT
```

2. Move the cursor to Reference Frame and press F4, CHOICE. Set the Reference Frame to UFRAME or WORLD.

3. Move the cursor to Plate Direction and press F4, CHOICE. Select the plate's normal direction. The direction could be X, Y, Z, or -X, -Y, -Z.

4. Move the cursor to Plate Thickness, type the plate's thickness, and press ENTER.

5. Move the cursor to Circle XY offset, type the circular detections X and Y, and press ENTER.

6. Move the cursor to Circle Z offset, type the circular detection Z offset, and press ENTER.

7. To clear all recorded positions, press F1, CLEAR, then F2, YES.

8. Move the cursor to each point to be taught, and teach the point. Refer to Table 15–22 for information about the points.

- Press F2, LOAD, to load the TCAL_G1 positions to the menu's position data.
- Press F5, MV_LIN, to move to a position.

9. Press PREV to exit from the Teach Points menu.

```
iRCalibration TCP Shift
        1 Robot: Group [1]  Device: Sensor
          Calibration Type: TCP Recovery
        2 Setup            (Done    )
        3 Teach Points     (Done    )
        4 Create Program   (Not Done)
        5 Run: TCAL_G1     (Not Done)
        6 Update Utool[ 0] (Not Done)
        7 View Result
```

If all required positions are taught, the Teach Points status will be Done and the cursor will automatically move to Create Program. Otherwise, the cursor will stay on Teach Points and its status will remain unchanged.

10. To create the program, move the cursor to Create Program and press ENTER. When *i*RCalibration TCP Shift has created the program, the Create Program status will change to Done.

Refer to Figure 15–20 for a created TCP calibration program with calibration type set to NEW-XYZWPR.

Figure 15–20. *i*RCalibration TCP Shift XYZWPR Calibration Program Example

```
 1:  UFRAME_NUM=1
 2:  UTOOL_NUM=1
 3:  UTool Start[1]
 4:J P[1:Approach pos.] 100mm/sec FINE
 5:L P[2:Plate position] 100mm/sec FINE
 6:  Detect Point[1] World [-Z]
 7:
 8:L P[3:Approach Pos.] 30deg/sec FINE
 9:L P[4:Center 1 Inside] 100mm/sec FINE
11:  Detect Circle[1]( 30.00, 10.00)World[-Z]
12:
13:L P[5:Approach Pos.] 100mm/sec FINE
14:  UTool End [XYZ]
```

15.5.3.3 Custom *i*RCalibration TCP Shift Program Creation

*i*RCalibration TCP Shift is very flexible for custom iRCalibration TCP Shift program creation. Custom programs can be used for TCP Recovery in any combination of X, Y, Z translational

directions. Custom iRCalibration TCP recovery programs for partial orientation are not supported. If you need to recover a TCP with orientation by custom programming you must use the Gripper mode and recover six degrees of freedom.

For custom TCP XYZ recovery, it is not necessary to have special UFrame settings. However, if a UFrame is used it cannot change location with respect to the contact surface from the time of mastering to the time of recovery.

*i*RCalibration TCP Shift supports the recovery types of XYZ, XY, and Z. These types indicate how many searches are allowed for the recovery. XYZ requires three searches, XY requires two searches, and Z requires one search. However, the actual search directions do not need to correspond to the indicated directions. Each search direction must be perpendicular to the others, though.

Note that using custom *i*RCalibration TCP Shift programming, a wide variety of sensors can be used. If TOS is used, the touch surface can be virtually any rigid object. The touch points do not need to be near each other, but must be perpendicular to each other with respect to the UTool. You can even use a single proximity sensor and rotate the tool 90 degrees between searches but use the same World search direction for each search.

Figure 15–21. Example of Manually Created XYZ Search TPP Program

```
 1:  UFRAME_NUM=1
 2:  UTOOL_NUM=1
 3:L P[1:Home Position] 100mm/sec FINE
 4:  UTool Start[1]
 5:L P[2:Via Pos 1] 100mm/sec FINE
 6:L P[3:Search Pos 1] 100mm/sec FINE
 7:  Detect Point[4] UFrame[X]
 8:L P[4:Via Pos 2] 100mm/sec FINE
 9:L P[5:Search Pos 2] 100mm/sec FINE
10:  Detect Point[4] UFrame[Y]
11:L P[6:Via Pos 3] 100mm/sec FINE
12:L P[7:Search Pos 3] 100mm/sec FINE
13:  Detect Point[4] UFrame[-Z]
14:  UTool End XYZ
15:L P[8:Via Pos 4] 100mm/sec FINE
16:L P[9:Home Position] 100mm/sec FINE
```

The three searches can be in any order, but they must all be perpendicular to each other. This could also be done with a single Find Circle instruction with XY and Z parameters, or a Find Circle with parameters both zero and another search perpendicular to the plane of the circle.

Figure 15–22. Example of Manually Created XY Search TP Program

```
 1:  UFRAME_NUM=1
 2:  UTOOL_NUM=1
```

```
  3:L P[1:Home Position] 100mm/sec FINE
  4:  UTool Start[1]
  5:L P[2:Via Pos 1] 100mm/sec FINE
  6:L P[3:Search Pos 1] 100mm/sec FINE
  7:  Detect Point[4] UFrame[X]
  8:L P[4:Via Pos 2] 100mm/sec FINE
  9:L P[5:Search Pos 2] 100mm/sec FINE
 10:  Detect Point[4] UFrame[-Z]
 11:  UTool End XY
 12:L P[6:Via Pos 3] 100mm/sec FINE
 13:L P[7:Home Position] 100mm/sec FINE
```

Notice the UTool End XY instruction on line 8. This tells *i*RCalibration TCP Shift that this program will find two perpendicular corrections along the search direction axes, UFrame X and UFrame -Z. It does not mean that the correction will necessarily be in any frame X or Y direction.

Figure 15–23. Example of Manually Created Z Search Teach Pendant Program

```
  1:  UFRAME_NUM=1
  2:  UTOOL_NUM=1
  3:L P[1:Home Position] 100mm/sec FINE
  4:  UTool Start[1]
  5:L P[2:Via Pos 1] 100mm/sec FINE
  6:L P[3:Search Pos 1] 100mm/sec FINE
  7:  Detect Point[4] UFrame[X]
  8:  UTool End Z
  9:L P[4:Via Pos 2] 100mm/sec FINE
 10:L P[5:Home Position] 100mm/sec FINE
```

Notice the UTool End Z instruction on line 8. This tells *i*RCalibration TCP Shift that this program will find a single correction along the search direction axis, UFrame X. It does not mean that the correction will necessarily be in any frame Z direction.

Figure 15–24. Example of Manually Created XYZ Search TP Program Using Detect Circle Instruction (with XY and Z offsets)

```
1:  UFRAME_NUM=1
   2:  UTOOL_NUM=1
   3:L P[1:Home Position] 100mm/sec FINE
   4:  UTool Start[1]
   5:L P[2:Via Pos 1] 100mm/sec FINE
   6:L P[3:Search Pos 1] 100mm/sec FINE
   7:  Detect Circle[4] ( 30.00, 10.00)
       UFrame[X]
   8:  UTool End XYZ
```

```
   9:L P[4:Via Pos 2] 100mm/sec FINE
  10:L P[5:Home Position] 100mm/sec FINE
```

Notice the Detect Circle instruction on line 7. The parameter of 30 (the XY offset parameter) indicates that the robot will move the tool 30mm perpendicular to the search direction (i.e. it will move in the UFrame YZ plane) before performing a search along the UFrame X direction. After detecting the surface, the robot will return to P3. Then it will move along the UFrame X direction until it is 10mm (the Z offset parameter) past the point where the surface was detected in the first search. Then the robot will perform four searches in the UFrame YZ plane to find the center of the circle then return to P3.

Figure 15–25. Example of Manually Created XY Search TP Program Using Detect Circle Instruction (without XY and Z offsets)

```
1:  UFRAME_NUM=1
   2:  UTOOL_NUM=1
   3:L P[1:Home Position] 100mm/sec FINE
   4:  UTool Start[1]
   5:L P[2:Via Pos 1] 100mm/sec FINE
   6:L P[3:Search Pos 1] 100mm/sec FINE
   7:  Detect Circle[4] ( 0.00, 0.00)
       UFrame[X]
   8:  UTool End XYZ
   9:L P[4:Via Pos 2] 100mm/sec FINE
  10:L P[5:Home Position] 100mm/sec FINE
```

Notice the Detect Circle instruction on line 7. The parameters of (0.00, 0.00) indicate that the robot will not perform the surface detection. The robot tool must already be inside the circle. The robot will perform four searches in the UFrame YZ plane to find the center of the circle then return to P3.

Figure 15–26. Example of Standard iRCalibration TCP Shift-Generated Program with Update Frame Added

```
1:  UFRAME_NUM=1
   2:  UTOOL_NUM=1
   3:L P[1:Approach pos.] 100mm/sec FINE
   4:  UTool Start[1]
   5:L P[2:Plate position] 100mm/sec FINE
   6:  Detect Point[4] World[Y]
   7:  Update Tool
   8:L P[3:Approach pos.] 100mm/sec FINE
   9:L P[4:Center 1 inside] 100mm/sec FINE
  10:  Detect Circle[4](  0.00,  0.00)World[Y]
  11:  Update Tool
  12:L P[5:Center 2 inside] 100mm/sec FINE
  13:  Detect Circle[4](  0.00,  0.00)World[Y]
```

```
14:L P[6:Approach pos.] 100mm/sec FINE
15:  UTool End XYZWP
```

Notice the Update Tool instruction at line 7 and line 11. The first Update Tool will compensate for the detection in the plate direction. This method is especially important if there is a large variation in the Tool Z direction. Then the next detection, Detect Circle on line 10 will have the proper amount of the tool protruding through the circle. After this detection, the second Update Tool instruction is executed to help center the tool on the circle. This update may or may not be useful. If the TCP deviation was due to a significant orientation change of the tool then this Update Tool may actually make the next Detect Circle instruction closer to the edge of the circle. Regardless of this, for large orientation changes it is recommended to run *i*RCalibration TCP Shift twice.

15.5.3.4 Gripper Program Creation

The following is an example of a TCP Gripper recovery program. This program is manually created and must contain at least six searches where at least three are non-colinear and have a common search direction, at least two have a common search direction approximately perpendicular to the others and at least on with a search direction approximately perpendicular to the others. The points should be separated from each other.

The UFrame used must be located at the contact point. Typically the contact point is a pointer type object with UFrame set to the tip. The TCP Shift program searches cause the gripper to make contact with the pointer tip. Any of the detection methods can be used.

Multiple UFrames and multiple detection points can be used in the same TCP Shift gripper program. Each UFrame must correspond to the current detection point. This is useful for large grippers or for where unidirectional sensors such as proximity sensors are used for detection.

```
 1:  UFRAME_NUM=1
 2:  UTOOL_NUM=1
 3:L P[1:Home Position] 100mm/sec FINE
 4:  UTool Start[1]
 5:L P[2:Via Pos 1] 100mm/sec FINE
 6:L P[3:Search Pos 1] 100mm/sec FINE
 7:  Detect Point[4] UFrame[X]
 8:L P[4:Via Pos 2] 100mm/sec FINE
 9:L P[5:Search Pos 2] 100mm/sec FINE
10:  Detect Point[4] UFrame[X]
11:L P[6:Via Pos 3] 100mm/sec FINE
12:L P[7:Search Pos 3] 100mm/sec FINE
13:  Detect Point[4] UFrame[X]
14:L P[8:Via Pos 4] 100mm/sec FINE
15:L P[9:Search Pos 4] 100mm/sec FINE
16:  Detect Point[4] UFrame[Y]
17:L P[10:Via Pos 5] 100mm/sec FINE
```

```
18:L P[11:Search Pos 5] 100mm/sec FINE
19:  Detect Point[4] UFrame[Y]
20:L P[12:Via Pos 6] 100mm/sec FINE
21:L P[13:Search Pos 6] 100mm/sec FINE
22:  Detect Point[4] UFrame[-Z]
23:  UTool End Gripper
24:L P[14:Via Pos 7] 100mm/sec FINE
25:L P[15:Home Position] 100mm/sec FINE
```

This is an example of a simple six point program for recovering a gripper UTool. The searches can be done in any order but in this example the first three searches have a parallel search direction. They must not be colinear and must have significant distance between them. The next two searches have a search direction perpendicular to the first three and have a significant distance between them. The last search direction is perpendicular to all the other search directions. More points can be used. It is recommended to run TCP Shift iteratively until the UTool correction is small.

```
 1:  UFRAME_NUM=1
 2:  UTOOL_NUM=1
 3:L P[1:Home Position] 100mm/sec FINE
 4:  UTool Start[1]
 5:L P[2:Via Pos 1] 100mm/sec FINE
 6:L P[3:Search Pos 1] 100mm/sec FINE
 7:  Detect Point[4] UFrame[X]
 8:L P[4:Via Pos 2] 100mm/sec FINE
 9:L P[5:Search Pos 2] 100mm/sec FINE
10:  Detect Point[4] UFrame[X]
11:L P[6:Via Pos 3] 100mm/sec FINE
12:  UFRAME_NUM=2
13:L P[7:Search Pos 3] 100mm/sec FINE
14:  Detect Point[4] UFrame[X]
15:L P[8:Via Pos 4] 100mm/sec FINE
16:  UFRAME_NUM=3
17:L P[9:Search Pos 4] 100mm/sec FINE
18:  Detect Point[4] UFrame[Y]
19:L P[10:Via Pos 5] 100mm/sec FINE
20:L P[11:Search Pos 5] 100mm/sec FINE
21:  Detect Point[4] UFrame[Y]
22:L P[12:Via Pos 6] 100mm/sec FINE
23:L P[13:Search Pos 6] 100mm/sec FINE
24:  Detect Point[4] UFrame[-Z]
25:  UTool End Gripper
26:L P[14:Via Pos 7] 100mm/sec FINE
27:  UFRAME_NUM=1
28:L P[15:Home Position] 100mm/sec FINE
```

Notice that on line 12 and 16 different UFrame numbers are set. The TCP Shift Gripper method supports using multiple UFrame numbers. However, each UFrame location must closely correspond to the sensor location or contact surface location. This example shows one way of recovering the UTool of a large gripper with minimal robot motion and rotation by using multiple sensors corresponding to multiple UFrame numbers. You must not change the UTool number in any *i*RCalibration TCP Shift program.

15.5.4 Running the *i*RCalibration TCP Shift Program

In order to recover an existing TCP with *i*RCalibration TCP Shift, you must first create a TCP calibration teach pendant program.

Procedure 15-22 Running the *i*RCalibration TCP Shift Program

Conditions

- The *i*RCalibration TCP Shift Create Program status is Done.

Steps

1. At the *i*RCalibration TCP Shift Main Menu

```
iRCalibration TCP Shift
        1 Robot: Group [1]  Device: Sensor
        2 Calibration Type: TCP Recovery
        3 Setup            (Done    )
        4 Teach Points     (Done    )
        5 Create Program   (Done    )
        6 Run: TCAL_G1     (Not Done)
        7 Update Utool[ 0] (Not Done)
        8 View Result
```

Press ENTER or F4, SELECT, to run the TCP calibration program.

2. You have the following choices:

a. Be sure the teach pendant is ON and all faults are cleared.

b. Press SHIFT+F2 to run the teach pendant program.

After the program is created, it is recommended to test run the *i*RCalibration TCP Shift program in low override to make sure there is no collision with any obstacle. You should run *i*RCalibration TCP Shift at 100% override for mastering/recovery of the TCP.

The Run Status will change to Done after the teach pendant program has been run successfully.

```
iRCalibration TCP Shift
        1 Robot: Group [1]  Device: Sensor
          Calibration Type: New TCP
        2 Setup             (Done    )
        3 Teach Points      (Done    )
        4 Create Program    (Done    )
        5 Run: TCAL_G1      (Done    )
        6 Update Utool[ 0] (Not Done)
        7 View Result
```

Updating UTool

The Update UTool menu allows you to update the UTool with the calibration result obtained from running the calibration program.

To update the UTool, move the cursor to Update UTool and press ENTER. If you want to check the TCP calibration result, press F4, SELECT. Refer to Procedure 15-23 .

Procedure 15-23 Updating the UTool

The Update UTool menu is not accessible unless you have completed all the steps before it.

1. At the *i*RCalibration TCP Shift main menu, move the cursor to Update UTool and press ENTER. You will see a screen similar to the following:

```
 TCP Calibration Result
       TCP Offset [G:1,  1]
         X:   0.018 Y:    0.206 Z:   −0.031
         W:  -0.000 P:   -0.000 R:    0.000
TCP Difference:             0.2091
Accumulated Offset:         0.291
```

2. Press F2, UPDATE, to update the group 1, UTool frame 1.

Note The UTool is not automatically updated even if its schedule has auto update set to TRUE. This is different from previous version of *i*RCalibration TCP Shift.

15.5.5 Displaying the UTool Log Screen

The *i*RCalibration TCP Shift view menu allows you to see previous calibration results stored in the UTool log.

The UTool log is a buffer that holds previous tool frame information. The buffer can hold up to 20 items. When the buffer is full, it will replace the oldest data in the buffer with the new logged data. If you want to view frame information, you can display the UTool Log screen. To display the UTool log screen, use Procedure 15-24 .

Procedure 15-24 Displaying the UTool Log Screen

Steps

1. Press MENU.
2. Select UTILITIES.
3. Select *i*RCalibration.
4. Select *i*RCalibration TCP Shift. You will see a screen similar to the following.

```
iRCalibration TCP Shift
        1 Robot: Group [1]  Device: Sensor
          Calibration Type: TCP Recovery
        2 Setup            (Done    )
        3 Teach Points     (Done    )
        4 Create Program   (Done    )
        5 Run: TCAL_G1     (Done    )
        6 Update Utool[ 0] (Not Done)
        7 View Result
```

5. Move the cursor to View Result and press ENTER.

```
 TCP Calibration Result
       TCP Offset [G:1,  1]
         X:   0.018 Y:    0.206 Z:   −0.031
         W:  -0.000 P:   -0.000 R:    0.000
TCP Difference:             0.209
Accumulated Offset:         0.209
```

6. To display the UTool Log, press F4, VIEW LOG. You will see a screen similar to the following

```
User Tool Log
                                          11/40
   Date      Time      X(W)      Y(P)      Z(R)
01-JAN-XX    16:57   —164.8        4.2    353.1
 UTool Num [ 1]          2.0     -70.6     -4.8
01-JAN-XX    16:57    -164.8       4.2    353.1
 UTool Num [ 1]          2.0        3.1    -81.4
01-JAN-XX    16:57   -166.1        4.2    350.3
 UTool Num [ 1]          2.1      —70.6     -4.9
No XYZ Data
No WPR Data
No XYZ Data
```

Each entry spans two lines. The first line lists the date and time of the saved entry as well as x, y, and z information. The second line contains the frame number and w, p, and r information.

If entries are not used or if they have been cleared, you will see the following lines:

No XYZ Data

No WPR Data

7. Select one of the following:

- **If you want to save the entire log to the file "TOOL_LOG.DT,"** press F2, SAVE.
- **If you want to restore a previous UTool value,** move the cursor to the line that has the values you want to restore and press F3, APPLY. You will see a prompt box similar to the following.

```
Restore old Frame?

Press F2 for YES
Press F3 for NO
```

Move the cursor to YES and press ENTER.

- **If you want to clear the log and remove all data entries,** press F4, CLEAR. You will see a screen similar to the following.

```
Clear All Logged Data?

  Press F2 for YES
  Press F3 for NO
```

Move the cursor to YES and press ENTER.

15.5.6 *i*RCalibration TCP Shift Error Recovery

If a *i*RCalibration TCP Shift error occurs during TCP calibration, the robot goes back to the starting position and an error message is displayed. Review the error message, correct the problem, then restart the TCP calibration program.

15.5.6.1 Redoing a *i*RCalibration TCP Shift Adjustment

To redo a *i*RCalibration TCP Shift adjustment and restart *i*RCalibration TCP Shift,

1. Review the error message for an indication of the problem.
2. Correct the problem.
3. Abort any program that is paused.
4. Enter the *i*RCalibration TCP Shift program name in the setup screen.
5. Press DONE to exit.
6. When *i*RCalibration TCP Shift asks "Program exists, skip Teach/Create?", reply YES.
7. Run the *i*RCalibration TCP Shift program: move the cursor to Run and press ENTER.

15.5.6.2 *i*RCalibration TCP Shift Fit Errors

Refer to Table 15–23 for information on troubleshooting *i*RCalibration TCP Shift fit errors.

Table 15–23. *i*RCalibration TCP Shift Fit Errors

Try This	Description
Verify that the tool is rigid.	Worn or damaged tooling components can cause variation in tool repeatability and can cause fit errors.
Verify the payload setting.	If the reference position is at an extreme position, improper payload settings can cause increased robot vibration, which affects *i*RCalibration TCP Shift repeatability.
Check for a loose touch plate. *	If the touch plate becomes loosened due to improper installation, *i*RCalibration TCP Shift repeatability will be affected.
Reduce motion speed. *	Under certain conditions, the default motion speed can cause too much vibration. For best performance, *i*RCalibration TCP Shift uses increased acceleration values. Try lowering the motion speed from the default 100mm/sec to 25 or 50mm/sec.

Table 15–23. *i*RCalibration TCP Shift Fit Errors (Cont'd)

Reduce search speed. *	Under certain conditions, the default motion speed can cause too much vibration. For best performance, *i*RCalibration TCP Shift uses increased acceleration values. Try lowering the search speed from the default 10mm/sec to 5mm/sec.
Increase the retry count. *	*i*RCalibration TCP Shift has significant safeguards against getting incorrect TCP corrections. You can increase the retry count from the default of 0 to either 3 or 4 to reduce the occurrence of *i*RCalibration TCP Shift fit errors.
Increase the radius tolerance. *	The default radius tolerance of .5mm may not be large enough to allow for the normal variation of some tooling. You can try increasing the radius tolerance from .5mm to 1.0mm or more as your application allows.
Increase the fit tolerance.*	Most *i*RCalibration TCP Shift errors are detected by radius tolerance, but if increasing the radius tolerance does not resolve the problem, you can increase the fit tolerance from .25mm to .5mm or more as your application allows.
If you are using TOS, increase the sensitivity tolerance value. *	Normally, the default value of 10 is adequate; however, you can increase this value to either 15 or 20.
If you are using TOS, increase the torque threshold. *	Normally, the default value of 100 is adequate. However, if the sensitivity tolerance increases have no effect, you can increase this value to 150.
Check for poor electrical continuity	If the I/O method is used, and if it relies on electrical continuity, then dirty contact surfaces on either the tool or the calibration plate will result in increased fit errors.

* Redo *i*RCalibration TCP Shift mastering after making this change.

Refer to Section 15.5.4 .

15.5.7 *i*RCalibration TCP Shift Troubleshooting

To troubleshoot *i*RCalibration TCP Shift, you can

- Display the UTOOL Log screen. Refer to Section 15.4.5
- Run *i*RCalibration TCP Shift to check the TCP. Refer to Section 15.5.4 .

15.6 *i*RCALIBRATION FRAME SET

15.6.1 Overview

*i*RCalibration Frame Set allows you to set up a cell, work, or part frame by using Torque Observer Sensor technology or Digital I/O using contact or proximity sensing. It also allows you to set up a frame based on the offset data. So the offline programs can be easily down load to the controller without frame offset error.

Limitations

The following limitations apply to *i*RCalibration Frame Set.

- *i*RCalibration Frame Set instructions will not be executed in single step test run mode.
- The TCP must be located near the contact surface of the tool.

Compatibility

*i*RCalibration Frame Set is compatible with the following:

- *i*RCalibration Frame Set can be used with robots with extended axes.
- *i*RCalibration Frame Set can be used with multiple groups in single-group programs.

Hardware Requirements

The following hardware is required to use *i*RCalibration Frame Set.

- Teaching tip, if desired (highly recommended for arc welding)
- I/O circuitry, if used

Software Requirements

The following software is required to use *i*RCalibration Frame Set.

- *i*RCalibration Frame Set software option

Installation

You must install the *i*RCalibration Frame Set software option before you can use *i*RCalibration Frame Set. Refer to the *Software Installation Manual* for more information about installing software options.

You do not need to install any hardware if TOS (Torque Observer Sensor) is used for contact detection. You will need to set up I/O ports for contact detection if you use I/O for contact detection. Refer to the "I/O Setup" chapter in this manual for information on setting up I/O ports.

You might also need to install a teaching tip as the robot's TCP before running the frame search teach pendant program, and take the teaching tip out after the frame search teach pendant program is complete.

15.6.2 Preliminary Setup

FANUC Robotics recommends that you set up *i*RCalibration Frame Set prior to starting production.

Note *i*RCalibration Frame Set always uses the current program's $MNUTOOL and $MNUFRAME for searching and position recording.

Perform the following steps prior to using *i*RCalibration Frame Set:

1. Define a Tool Center Point (TCP), including WPR if you will be performing tool-relative searches. Refer to the "Frames Setup section" or Section 15.4 , "*i*RCalibration TCP Set". For Arc Welding, refer to 6 Point UTOOL setup in the application-specific *Setup and Operations Manual*.
2. Install a teaching tip, if desired. This is highly recommended for arc welding and other applications in which the tooling is not conducive to repetitive contact with a part.
3. You must define a UFRAME. $MNUFRAMENUM[group_num] for that particular group must be non-zero.
4. If you are using I/O for detection, install and set up circuitry.

15.6.3 *i*RCalibration Frame Set Setup

Before you can use *i*RCalibration Frame Set, you must perform some setup procedures. Setup involves the initial setting of menu options within *i*RCalibration Frame Set software.

Refer to Section 15.6.2 for preliminary setup recommendations.

Setting up and running *i*RCalibration Frame Set includes the following processes:

1. Selecting the motion group you want to calibrate
2. Selecting the teach pendant program, the UFRAME you want to set, and the calibration mode (New or Nominal).
3. Setting up the Frame schedule. Refer to Table 15–24 for a description of the items on the Frame Schedule Detail Screen.
4. Setting up the detection schedule. Refer to Table 15–25 for a description of the items on the *i*RCalibration Frame Set Detection Schedule Detail Screen.
5. Teaching points for the teach pendant calibration program. Refer to Procedure 15-26 .

6. Creating the teach pendant calibration program from the taught points and running the calibration program. Refer to Section 15.6.4 .
7. Updating the UFrame with the calibration results obtained from running the calibration program. Refer to Procedure 15-29 .

Table 15–24. Frame Schedule Detail Schedule Setup

ITEM	DESCRIPTION
Auto Update Value: TRUE	This item indicates whether the *i*RCalibration Frame Set FIND END instruction will update the corresponding $mnuframe or not. If you set Auto Update to TRUE and run the *i*RCalibration Frame Set teach pendant program, the current $mnuframe will be updated based on the offset.
Store Offset Value: TRUE or FALSE Default: FALSE	This item stores the cell calibration offset to a specific position register.
in PR[0]	If this is non-zero and "Store Offset" is TRUE, then the UFrame offset is stored in the specified position register.
Max XYZ Offset: Min: 0.01 mm Max: 100 mm Default: 2.0 mm	This item indicates the tolerance for any detection point shifting. If the distance between the nominal (reference) detection point and the newly detected point is larger than this value, *i*RCalibration Frame Set will post a warning message.
Report on DOUT[0]:	If this is non-zero and *i*RCalibration Frame Set detects an error then this output port will be pulsed high for 500 milliseconds.
Fit Error Register.	If this item is non-zero then *i*RCalibration Frame Set will store the mean fit error of the detection points in this register. This value is the average deviation from nominal in the search direction of each detection point after the offset is applied
Max Orient Change: Min: 0.01 degrees Max: 10 degrees Default: 2.0 degrees	This item indicates the maximum tolerance for orientation changes. If the orientation of the frame exceeds this tolerance, *i*RCalibration Frame Set will post a warning message.
Mean Error Register Range: 0 — 999 Default: 0	If the register number is greater than 0, *i*RCalibration Frame Set writes the mean error to this register.
Max. Error Register Range: 0 — 999 Default: 0	If this item is greater than 0, *i*RCalibration Frame Set writes the computed maximum error to this register.

Table 15–24. Frame Schedule Detail Schedule Setup (Cont'd)

Max Dev Register:	If this item is non-zero, *i*RCalibration Frame Set will store the maximum deviation in this register. This is the distance between any of the nominal (reference) detection points and the corresponding newly detected point before the offset is applied.
CD Pair Number: Range: 0-4	This item is only used in conjunction with the Coordinated Motion option. If this item is non-zero, *i*RCalibration Frame Set will update the leader frame associated with the specified CD Pair Number. Dynamic-UFrame must be enabled and an active follower UFrame must be set. The Dynamic-UFrame feature of Coordinated Motion can be enabled by setting the system variable $CD_PARAM.$DYN_UFRAME=TRUE. The default is FALSE. Once Dynamic-UFrame is enabled, and a CD-Pair leader group and follower group are selected a menu item appears on the Coordinated Motion Setup menu:, "Follower UFrame Number". This value determines the active dynamic UFrame for the CD pair. By default, this dynamic UFrame is relative to the Leader Frame associated with the CD Pair. Refer to the *Coordinated Motion Setup and Operations* manual for more information.

When you set up a detection schedule, you

- Determine the sensor
- Determine the detection motion
- Determine TOS

Note Default values are usually acceptable.

Table 15–25. *i*RCalibration Frame Set Detection Schedule Description

ITEM	DESCRIPTION
Sensor Type Values: TOS Wrist, I/O, TOS ALL AXES Default: TOS ALL AXES	This item indicates the kind of sensor used for contact detection. The modes are: • **TOS Wrist** (Torque Observer Sensor on Wrist) - the contact torque will be monitored, and only the observed torque on the wrist will be used to determine the contact position. When TOS Wrist is selected, no I/O connection is required for contact detection. TOS Wrist is very sensitive and can be more prone to false detections in certain cases than the more robust TOS All Axes method. TOS Wrist is not recommended except for special cases where it can verified that false detections will not occur. • **I/O** — the specified I/O will be monitored and only the state of this input will be used to determine the contact position. Use the detection port for contact detection with I/O. Refer to Detection Port item in this table for more information. • **TOS ALL AXES** (Torque Observer Sensor, All Axes) - The contact torque will be monitored on all robot axes, and only the observed torque will be used to determine the contact position. This is the default value. When TOS ALL AXES is selected, no I/O connection is required for contact detection.
Detection Port Values: RDI, DI, WDI, WSI, AIN	This item indicates the desired input port type. The port type that you specify is followed by the desired port number that will go from OFF to ON when contact is detected. The WDI and WSI ports are typically used for arc welding. You will set this item up if you selected I/O as the Sensor Type. You do not need to set this up if you selected TOS Wrist or TOS All Axes as the Sensor Type. The port types are: • **RDI** — Robot Digital Input • **DI** - normal Digital Input • **WDI** - Weld Digital Input • **WSI** - Wire Stick circuit Input. • **AIN** - Analog Input. To enable analog sensor support, you must have a data file that resides in the FRS: directory before running *i*RCalibration Frame Set. The data file has a fixed name: cb_ainX,dt, where X is a digit from 1 to 9. Set the port number to be the same as AinPortNumber setting in the data file. Refer to the Preliminary Setup section for more information.

Table 15–25. *i*RCalibration Frame Set Detection Schedule Description (Cont'd)

ITEM	DESCRIPTION
Sensor Enable Values: RDO, DOUT, WDO, or WSO Default: RDO	This item indicates the I/O sensor enable digital output port type. The port types are: • **RDO** - Robot Digital Output • **DOUT** - normal Digital Output • **WDO** - Weld Digital Output • **WSO** - Wire Stick circuit Output. The port type that you specify is followed by the desired port number that will be used to enable the I/O detection circuit. The WDO and WSO ports are typically used for arc welding. You will set this item up if you selected I/O as the Sensor Type. Set the port number to 0 (zero) if you are not using this feature.
Detection Speed Units: mm/sec Range: 1.0 - 100 Default: 10	This item is the search speed for contact detection. This speed is used for both TOS and I/O searches. The value can be increased if cycle time is of great concern (although increasing the speed will also affect the resultant accuracy and repeatability) , or reduced if vibration or other disturbances cause *i*RCalibration Frame Set to retry excessively.
Detection Distance Units: mm Range: 1 - 1000 Default: 50	This item is the maximum distance that the robot will move while looking for the signal. If no contact is detected within the distance, an error will be posted.
Return Speed Units: mm/sec Range: 5 - 5000 Default: 50	This item is the search return speed. This speed is used for both TOS and I/O searches.
Return to Start Default: Yes	When this item is set to YES, the robot will move back to the position before it starts the search motion. This item applies to DETECT POINT instructions only. When this item is set to NO, the robot will move back to a distance specified by the next item (Return distance)
Return Distance Units: mm Range: 1 - 1000 Default: 50	When return to start is set to FALSE, the robot will move back to the distance specified.
Return term type Default: FINE	This item is the termination type for return motion.
Air Cut Speed Units: mm/sec Range: 1 - 1000 Default: 100	This item is the non-search motion speed.

Table 15–25. *i*RCalibration Frame Set Detection Schedule Description (Cont'd)

ITEM	DESCRIPTION
Learn Distance Units: mm Range: 0.5– 20.0 Default: 5.0	For "TOS All Axes" method only, there is a short motion before each search to identify dynamic torque values. Learn Distance is the distance in millimeters of this short motion, the direction is determined at the time of the motion.
Learn Speed Units: mm/sec Range: 0.5 – 50.0 Default: 10.0	For "TOS All Axes" method only, this is the speed of the learning motion before each search.
Sensitivity Tolerance Units: max. motor torque/sec. Range: 1 - 999 Default: 10	This item is the primary sensitivity tolerance of TOS Wrist and is not used for TOS All-Axes. It is the rate of change of the disturbance torque. The actual units vary with motor model, gear ratio, and amplifier current, and are affected by tooling length. However, the effect for calibration is largely independent of these factors and a value of 10 is considered normal operation for typical installations. You should not use a value of less than 10 or misdetections might occur. This value can be increased if false detections occur frequently. **Warning** **Be careful when you change this number. Changing the sensitivity tolerance might cause a false detection. This could injure personnel or damage equipment.**
Torque Threshold Units: max. motor torque Range: 1 to 999 Default: 100	For TOS All Axes, this item is the only measure of contact detection. During the learning motion a torque level is memorized and that level plus Torque Threshold must be exceeded for contact to be determined. If false detections are being experienced then this value should be increased. For TOS Wrist, this item is the level of disturbance torque that is used as another means for contact detection. The default value of 100 is considered normal operation for typical installations. If increasing the value of **Sensitivity Tolerance** does not make false detections disappear, then this value should be increased until the false detections disappear; then, reset the value of **Sensitivity Tolerance** .
Fit Tolerance Units: mm Range: 0.1 - 100 Default: 0.5	For Detect Circle instruction: The robot detects 4 points inside the circle and the calibration software will fit a circle to the 4 detected positions. The fit error is defined as the maximum distance from the 4 points to the circle. If the fitting error exceeds the tolerance, the calibration software will post an error.
Radius Tolerance Units: mm Range: 0.1— 100 Default: 0.5	The Detect Circle instruction stores the circle's radius during mastering. At run time, if the computed circle radius differs from the stored radius value and difference exceeds the tolerance, the calibration software will post an error.

Table 15–25. *i*RCalibration Frame Set Detection Schedule Description (Cont'd)

ITEM	DESCRIPTION
Number or Retry Range: 0 — 9 Default:	If the number of retry is >0, the calibration software will re-run the detect circle instruction if the current circle detection has either a fit error or a radius error.
Contact Position Register	If the position register number is any number other than 0, *i*RCalibration Frame Set will copy the contact position to the position register.

15.6.3.1 Setting up *i*RCalibration Frame Set

*i*RCalibration Frame Set uses a wizard-style menu. It shows the steps that need be done to complete the Cell calibration process. Each step of the process has its own status display that shows where you are in the setup process. You cannot skip steps and you cannot go to the next step without completing the current step.

*i*RCalibration Frame Set also includes a menu item, View Result, that lets you review the previous calibration result and log history.

Procedure 15-25 Setting up *i*RCalibration Frame Set

- The *i*RCalibration Frame Set software option is loaded.
- Preliminary setup is complete. Refer to Section 15.6.2 .

Steps

1. Press MENUS.
2. Select UTILITY.
3. Press F1, [TYPE].
4. Select iRCalibration. You will see a screen similar to the following:

```
UTILITIES iRCalibration
   1 TCP Shift
   2 TCP Set
   3 Frame Shift
   4 Frame Set
   5 Master Recovery
   6 Master Set
```

5. Select iRCalibration Frame Set and press ENTER. You will see a screen similar to the following:

```
iRCalibration Frame Set
        1 Robot: Group [1]  Device: Sensor
          Calibration Type: New UFrame
        2 Setup            (Not Done)
        3 Teach Points     (Not Done)
        4 Create Program   (Not Done)
        5 Run: CCAL_G1     (Not Done)
        6 Update UFrame[ 0](Not Done)
        7 View Result
```

6. Move the cursor to Robot: Group [1], type the group number of the motion group you want to calibrate, and press ENTER.

Note Calibration Type is always New UFrame. You cannot change it.

7. Specify the Calibration Program Name. The *i*RCalibration Frame Set Setup menu allows you to specify the Cell calibration TP program name, the UFrame number, and the calibration mode, and select the *i*RCalibration Frame Set Frame schedule and the detection schedule.

Note The Group number is provided for reference. You cannot change it in this menu.

a. Move the cursor to Setup and press ENTER. You will see a screen similar to the following:

```
iRCalibration Frame Setup
        1 Program name: CCAL_G1    Group[1]
        2 Uframe Number:    1
        3 Calibration Mode: New
        4 Frame Schedule:  1
        5 Detect Schedule:  1
```

b. The default program name is CCAL_Gx, where x is the group number. If you want to specify a different program name, press ENTER and use the function keys, or press SELECT to select it from the list of programs. Refer to the "Writing and Modifying a Program" section of your application-specific Setup and Operations manual for more information.

8. Select the UFrame. Move the cursor to UFrame Number:, type the number of the User Frame you want to calibrate, and press ENTER.

9. Select the Calibration Mode. Move the cursor to Calibration Mode. Press F4, CHOICE and select Standard or Dynamic-UFrame.

- New will create a new user frame.
- Nominal will create a new user frame with respect to the user input data. Refer to Section 15.6.5 .

10. Select the Frame Schedule and Set Frame Schedule detail items. Move the cursor to Frame Schedule.

- To change the number of the Uframe, move the cursor to the schedule number, type the number of the Frame Schedule you want, and press ENTER.
- To see the detail screen for the selected uFrame, press ENTER. You will see a screen similar to the following:

```
Frame Schedule                1/9
   Schedule Number[  1]
   1 Auto Upate:            TRUE
   2 Store Offset: FALSE
   3               In PR[0  ]
   4 Max. XYZ Offset:      30.000 mm
   5      Report on DOUT[0  ]
   6 Max. Orient Offset:   5.000  deg
   7 Fit Error Register:   0
   8 Max. Dev. Register:   0
   9 CD Pair Number:       0
```

Set each item as desired. For information on the Frame Schedule Setup items, refer to Table 15–24 .

Note The default settings are good for most applications.

When you have finished setting up the Frame Schedule, press PREV to return to the *i*RCalibration Frame Set Setup Menu.

```
iRCalibration Frame Setup
        1 Program name: CCAL_G1    Group[1]
        2 Uframe Number:    1
        3 Calibration Mode: New
        4 Frame Schedule:  1
        5 Detect Schedule:  1
```

11. Select the Detection Schedule and Set Detection Schedule detail items. Move the cursor to Detect Schedule.

- To change the number of the Detect Schedule, move the cursor to the schedule number, type the number of the schedule you want, and press ENTER.
- To set up the detection schedule detail items, press F4, SELECT. You will see a screen similar to the following:

```
Detection Schedule
                                        1/18
   Detection Schedule [ 1]
   Sensor Type:               TOS ALL AXES
   I/O Setup
     Detection Port:         DI   [  1]
     Sensor Enable:          DOUT[  1]
   Motion Setup
     Detection Speed:        10.000  mm/sec
     Detection Distance:     50.000  mm
     Return Speed:           50.000  mm/sec
     Return to Start:        TRUE
       Return Distance:      0.000   mm
     Return Term Type:       Fine
     Learn Speed:            10.000  mm/sec
     Learn distance:         5.000   mm
     Air Cut Speed:          100.000 mm/sec
   TOS Setup
     Sensitivity:            10.000
     Torque Threshold:       100
   Circle Setup
     Fit Tolerance:         1.000   mm
     Radius Tolerance:      1.000   mm
     Number of Retry:       0
   Contact Position Register:  0
```

Set each item as desired. Move the cursor to the item you want to change and type the value.

Note The default settings are good for most applications.

Refer to Table 15–25 for information about detection schedule setup detail items.

12. When setup is complete, press F1, DONE. You will see the *i*RCalibration Frame Set main menu with Setup (Done) in its status display.

Note You must press DONE to change the Setup status. If you press PREV to return to the *i*RCalibration Frame Set main menu, the Setup status display will not change.

```
iRCalibration Frame Set
        1 Robot: Group [1]  Device: Sensor
          Calibration Type: New UFrame
        2 Setup            (Done    )
        3 Teach Points     (Not Done)
        4 Create Program   (Not Done)
        5 Run: CCAL_G1     (Not Done)
        6 Update UFrame[ 0](Not Done)
        7 View Result
```

If the TP program already exists when you press DONE, you will see the following screen:

```
Program exists. Skip Teach/Create?
  Press F2 for YES
  Press F3 for NO
```

If the teach pendant program already exists and has been taught, and you want to skip the teach/create step, press F2, YES, and proceed directly to Run. Refer to Procedure 15-28 to run the program.

If the teach pendant program exists but you want to teach it, change it, or use another program, press F3, NO, and continue to Section 15.6.3.2 .

15.6.3.2 Creating a Frame Calibration Teach Pendant Program

In order to master a frame with *i*RCalibration Frame Set, you must first create a frame calibration teach pendant program. You can create the program manually or you can have *i*RCalibration Frame Set create the program for you automatically. The frame calibration teach pendant program will detect contact with the fixture for mastering and recovery of the frame. You need one program for each frame you master.

When *i*RCalibration Frame Set setup is complete, the *i*RCalibration Frame Set main menu will be displayed with the cursor at Teach Points.

```
iRCalibration Frame Set
        1 Robot: Group [1]  Device: Sensor
          Calibration Type: UFrame Recovery
        2 Setup            (Done    )
        3 Teach Points     (Not Done)
        4 Create Program   (Not Done)
        5 Run: CCAL_G1     (Not Done)
        6 Update UFrame[ 0](Not Done)
        7 View Result
```

You can let *i*RCalibration Frame Set create the teach pendant program automatically, or you can create your own teach pendant program.

- **To have *i*RCalibration Frame Set create the teach pendant program automatically**, continue to Procedure 15-26 .
- **To create the teach pendant program manually**, refer to Procedure 15-27 .

Procedure 15-26 Teaching Points (Creating the *i*RCalibration Frame Set Teach Pendant Program Automatically)

Condition

- *i*RCalibration Frame Set Setup has been completed.
- The teach pendant program does not already exist, or you want to overwrite the existing program.

1. When you have completed *i*RCalibration Frame Set setup, the Setup status on the *i*RCalibration Frame Set main menu will be Done.

```
iRCalibration Frame Set
        1 Robot: Group [1]  Device: Sensor
          Calibration Type: New UFrame
        2 Setup            (Done    )
        3 Teach Points     (Not Done)
        4 Create Program   (Not Done)
        5 Run: CCAL_G1     (Not Done)
        6 Update UFrame[ 0](Not Done)
        7 View Result
```

Move the cursor to Teach Points and press ENTER. You will see a screen similar to the following:

```
Standard iRCalibration Frame Set Points
                                                1/8
    Program Name:          CCAL_G1
    Robot Group:       1
    iRCalibration Frame Set Schedule:  1
    Detection Sched:   1

    Home Position: UNINIT
    Circle XY offset:  20.00  mm
    Circle Z offset:    10.00  mm
    P1:Via:[UNINIT  ]Srch:[UNINIT  ]Dir:X
    P2:Via:[UNINIT  ]Srch:[UNINIT  ]Dir:X
    P3:Via:[UNINIT  ]Srch:[UNINIT  ]Dir:X
```

2. To clear all recorded positions, press F1, CLEAR, then F2, YES.
3. Teach the positions.

   - **If the teach pendant program already exists**, press F2, LOAD, to load the positions to the menu's position data.
   - **If the teach pendant program does not exist,** teach the Home position (the position where the *i*RCalibration Frame Set program will start and finish) and each of the three Via points and Search points by using the F4, MV_JNT, and F5, MV_LIN, keys to move to positions.

     For each search point choose a search direction of +/-X, +/-Y, or +/-Z. Each search direction will be with respect to UFrame. You can later change the generated calibration program to search with respect to UTool if that is a better search direction for your application. Searching with respect to World in a *i*RCalibration Frame Set program is not recommended

     Press F3, RECORD, to record the positions.

     Note The robot will do a circle search at each search point. The first search point defines the origin of the user frame. The second search point defines the X direction of the user frame and the third point defines the Y direction of the user frame.

4. When you have finished, press PREV to return to the *i*RCalibration Frame Set Main Menu.

```
iRCalibration Frame Set
        1 Robot: Group [1]  Device: Sensor
          Calibration Type: UFrame Recovery
        2 Setup            (Done    )
        3 Teach Points     (Done    )
        4 Create Program   (Not Done)
        5 Run: CCAL_G1     (Not Done)
        6 Update UFrame[ 0](Not Done)
        7 View Result
```

If all required positions are taught, the Teach Points status will be **Done** and the cursor will move automatically to the next step. Otherwise, the cursor will stay at Teach Points and its status will remain unchanged.

5. To create the program, move the cursor to Create Program and press ENTER. When *i*RCalibration Frame Set has created the program, the Create Program status will change to Done.

The following program is an example of a *i*RCalibration Frame Set Teach Pendant Program generated by *i*RCalibration Frame Set.

```
1:  UFRAME_NUM=1
   2:  UTOOL_NUM=1
   3:L P[1:Home Position] 100mm/sec FINE
   4:  UFrame Start[1]
   5:L P[2:Via Pos 1] 100mm/sec FINE
   6:L P[3:Search Pos 1] 100mm/sec FINE
   7:  Detect Circle[1] ( 20.00, 10.00) UFrame[X]
   8:L P[4:Via Pos 2] 100mm/sec FINE
   9:L P[5:Search Pos 2] 100mm/sec FINE
  10:  Detect Circle[1] (20.00, 10.00) UFrame[X]
  11:L P[6:Via Pos 3] 100mm/sec FINE
  12:L P[7:Search Pos 3] 100mm/sec FINE
  13:  Detect Circle[1] (20.00, 10.00) UFrame[X]
  14:  L P[8:Home Position] 100mm/sec FINE
  15:  UFrame End[New]
```

6. Single step through the *i*RCalibration Frame Set calibration program in low override to make sure that there is no collision with any obstacle.

Guidelines for Creating a Frame Calibration Teach Pendant Program Manually

Be familiar with the following guidelines as you create the frame search teach pendant program.

- The contact surface should be relatively flat.
- The search direction must be approximately perpendicular to the contact surface.
- The robot will return to the starting point (or as specified by “return distance” in the detection schedule) after each search.
- Multiple searches (in different directions) can be done at the same starting point as long as the search direction is approximately perpendicular to the contact surface.
- If the contact surface is not approximately perpendicular to a UFRAME or WORLD axis then search frame must be TOOL. You must orient the tool so that one of the TOOL axes is perpendicular to the contact surface.
- The first circle defines the origin of the user frame. The second circle defines the X direction of the user frame and the third circle defines the Y axis of the user frame.
- The UFRAME cannot be changed in the calibration program.

- The order that you teach the points is important. The order defines where the origin point is.

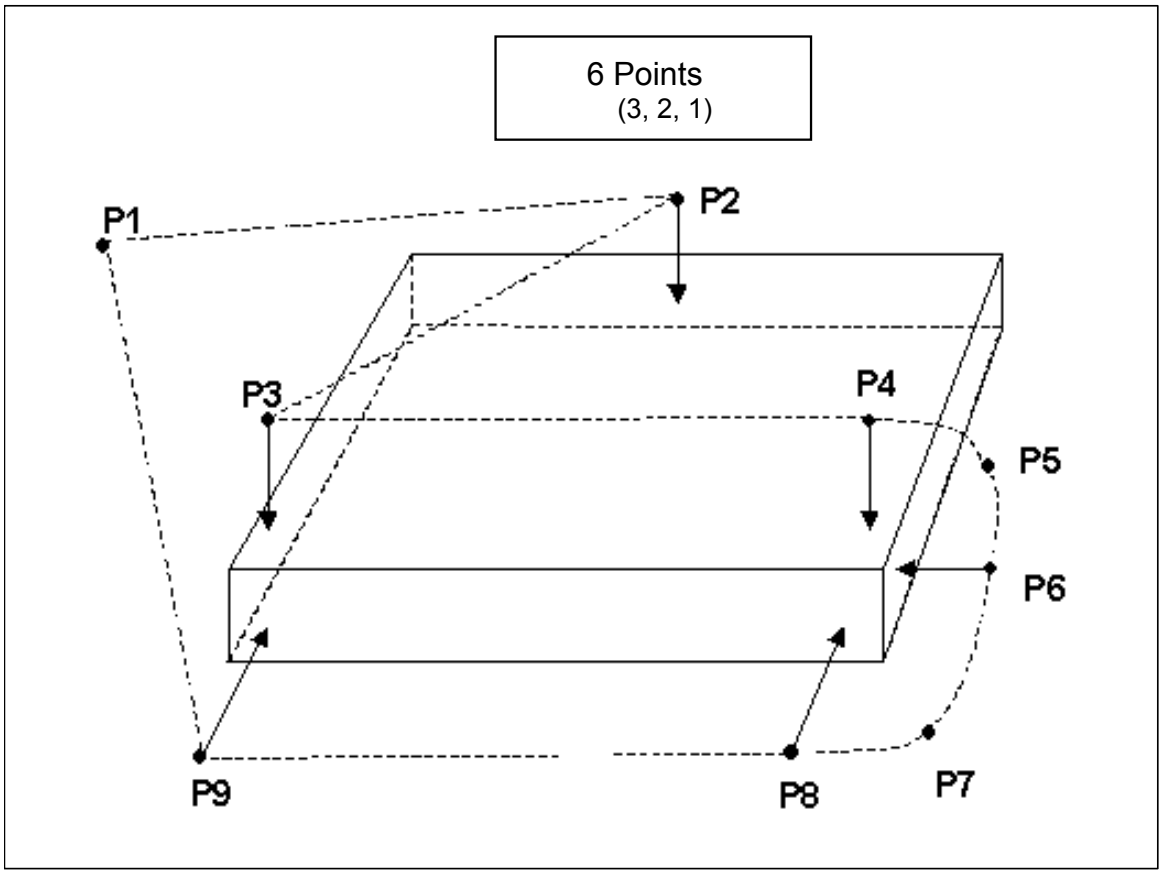
Procedure 15-27 Creating a Frame Calibration Teach Pendant Program Manually

1. Press F2, CREATE.
2. Create a teach pendant program. (See Figure 15–27 for a program example. Refer to the "Planning and Creating a Program" chapter for information on creating a teach pendant program.)

Figure 15–27. 6 Points Search (3-2-1) (Corresponds to Figure 15–28)

```
 1:  UFRAME_NUM=1
 2:  UTOOL_NUM=1
 3:L P[1:Home Position] 100mm/sec FINE
 4:  UFrame Start[1]
 5:L P[2:Search Pos 1] 100mm/sec FINE
 6:  Detect Point[4] UFrame[-Z]
 7:L P[3:Search Pos 2] 100mm/sec FINE
 8:  Detect Point[4] UFrame[-Z]
 9:L P[4:Search Pos 3] 100mm/sec FINE
10:  Detect Point[4] UFrame[-Z]
11:  Update Frame
12:L P[5:Via Pos 4] 100mm/sec FINE
13:L P[6:Search Pos 4] 100mm/sec FINE
14:  Detect Point[4] UFrame[-Y]
15:  Update Frame
16:L P[7:Via Pos 5] 100mm/sec FINE
17:L P[8:Search Pos 5] 100mm/sec FINE
18:  Detect Point[4] UFrame[X]
19:L P[9:Search Pos 6] 100mm/sec FINE
20:  Detect Point[4] UFrame[X]
21:  UFrame End  [Standard / Dynamic-UFrame]
22:L P[1:Home Position] 100mm/sec FINE
```

Figure 15–28. 6 points (3-2-1) for x, y, z, w, p, r Offset



This program is very similar to the one created automatically by *i*RCalibration Frame Set. It has fewer Via points and has two extra instructions, Update Frame, on line 11 and 15.

- The first Update Frame instruction on line 11 allows the XY plane to be corrected for orientation. So, the searches in the X and Y directions are at the proper height with respect to the top of the plane.
- The second Update Frame instruction on line 15 provides an additional update for the new detection in the Y direction. This allows the detections on lines 19 and 20 to be at the proper location with respect to P6.

You can have as few as one and as many as 24 detections in a *i*RCalibration Frame Set program, although it is rare to have more than 12. You can intermix Detect Circle and Detect Point instructions. If there are fewer than six points before Uframe End or before an Update Frame instruction, then the points must meet the following restrictions:

- All searches must be approximately perpendicular or parallel to each other
- There must be no more than three parallel searches
- There must be no more than two parallel searches that are colinear

- There must be significant distance between all parallel searches

It is recommended to repeat *i*RCalibration Frame Set program execution until the max deviation is small.

Figure 15–29. Example Program for Automatic Operation

```
1:  R[1:Loop Counter]=0
   2:  LBL[1]
   3:  CALL ACELL2A
   4:  R[1:Loop Counter]=
       R[1:Loop Counter]+1
   5:  IF R[2:iRCalibration Frame Set Max Dev]<=.5,
       JMP LBL[2]
   6:  IF R[1:Loop Counter]<5,
       JMP LBL[1]
   7:  MESSAGE[ERROR]
   8:  PAUSE
   9:  LBL[2]
```

The Maximum Deviation register is R[2]. In this example, if the maximum deviation is not less than 0.5mm in five iterations of the *i*RCalibration Frame Set program ACELL2A then the program will pause and a message issued.

15.6.4 Running the *i*RCalibration Frame Set Program and Updating the User Frame

In order to create a new user frame with *i*RCalibration Frame Set, you must first create a frame calibration teach pendant program. The frame calibration teach pendant program will detect contact with the fixture to set up the initial user frame.

Procedure 15-28 Running the *i*RCalibration Frame Set Program

Conditions

- The *i*RCalibration Frame Set Create Program status is Done.

Steps

1. At the *i*RCalibration Frame Set Main Menu

```
iRCalibration Frame Set
        1 Robot: Group [1]  Device: Sensor
          Calibration Type: New UFrame
        2 Setup            (Done    )
        3 Teach Points     (Done    )
        4 Create Program   (Done    )
        5 Run: CCAL_G1     (Not Done)
        6 Update UFrame[ 0](Not Done)
        7 View Result
```

Press ENTER or F4, SELECT, to run the Cell calibration program.

2. You will see the following message:

```
Execute Calibration Program?
  Press F2 for YES
  Press F3 for NO
```

a. Be sure the TP is ON and all faults are cleared.

b. Press SHIFT+F2 to run the TP program.

The Run Status will change to Done after the teach pendant program has been run successfully.

```
iRCalibration Frame Set
        1 Robot: Group [1]  Device: Sensor
          Calibration Type: UFrame Recovery
        2 Setup            (Done    )
        3 Teach Points     (Done    )
        4 Create Program   (Done    )
        5 Run: CCAL_G1     (Done    )
        6 Update UFrame[ 0](Not Done)
        7 View Result
```

To create a new UFrame, move the cursor to View Result and press ENTER or F4, SELECT to check the user frame calibration result.

Procedure 15-29 Updating the User Frame

The Update UFrame menu is not accessible unless you have completed all the steps before it.

1. At the *i*RCalibration Frame Set main menu, move the cursor to Update UFrame and press ENTER. You will see a screen similar to the following:

```
Frame Calibration Result

 New UFrame[G:1, 1]
   X: 988.805 Y:   -3.351 Z:  393.492
   W: -89.925 P:    0.073 R:    0.000
```

2. Press F2, UPDATE, to update the group 1, user frame 1.

Note User frame is not automatically updated even its schedule has auto update set to TRUE. This is different from previous version of *i*RCalibration Frame Set.

15.6.5 Using *i*RCalibration Frame Set with Nominal Data

The *i*RCalibration Frame Set with Nominal data allows user to set up the user frame in respect to other frames, such as the part coordinate frame used in the offline simulation. To use the *i*RCalibration Frame set with Nominal data, select the Calibration Mode to Nominal in the setup screen. The Setup screen will display Nominal Data entry that allow you to type in the corresponding positions in other frame coordinate for the three search positions. Press Done to exit from the Nominal data screen.

15.6.6 Displaying the UFrame Log Screen

The UFrame log is a buffer that holds previous user frame or tool frame information. The buffer can hold up to 20 items. When the buffer is full, it will replace the oldest data in the buffer with the new logged data. If you want to view frame information, you can display the Frame Log screen. To display the frame log screen, use Procedure 15-30 .

Procedure 15-30 Displaying the UFrame Log Screen

Steps

1. Press MENU.
2. Select UTILITIES.
3. Select iRCalibration.
4. Select iRCalibration Frame Set. You will see a screen similar to the following.

```
iRCalibration Frame Set
        1 Robot: Group [1]  Device: Sensor
          Calibration Type: UFrame Recovery
        2 Setup             (Not Done)
        3 Teach Points      (Not Done)
        4 Create Program    (Not Done)
        5 Run: CCAL_G1      (Not Done)
        6 Update UFrame[ 0](Not Done)
        7 View Result
```

5. Move the cursor to View Result and press ENTER.

```
 Frame Calibration Result

  UFrame Offset:[G:1, Num:  ]
    X: 988.805 Y:   -3.351 Z:  393.492
    W: -89.925 P:    0.073 R:    0.000

  UFrame XYZ Change:       0.217 mm
  UFrame Angular Change:   0.030 deg
  Mean Fit Error:          0.000 mm
  Maximum Deviation:       0.163 mm
```

6. To display the UFrame Log, press F4, VIEW LOG. You will see a screen similar to the following

```
User Frame Log
                                        13/40
   Date     Time     X(W)     Y(P)     Z(R)
01-JAN-XX   16:57    989.3      -3.3   394.1
 UFrame Num[ 1]      -89.9        .0      .5
01-JAN-XX   16:57    989.3      -3.3   394.1
 UFrame Num[ 1]      -89.9        .0      .5
No XYZ Data
No WPR Data
No XYZ Data
No WPR Data
No XYZ Data
```

Each entry spans two lines. The first line lists the date and time of the saved entry as well as x, y, and z information. The second line contains the frame number and w, p, and r information.

If entries are not used or if they have been cleared, you will see the following lines:

NO XYZ DATA

NO WPR DATA

7. Select one of the following:

- **If you want to save the entire log to the file "UFRM_LOG.DT,"** press F2, SAVE.
- **If you want to restore a previous UFrame value,** move the cursor to the line that has the values you want to restore and press F3, APPLY. You will see a prompt box similar to the following.

```
User Frame Log
     Restore old Frame?
           YES        NO
```

Move the cursor to YES and press ENTER.

- **If you want to clear the log and remove all data entries,** press F4, CLEAR. You will see a screen similar to the following.

```
User Frame Log
     Clear All Logged Data?
           YES        NO
```

Move the cursor to YES and press ENTER.

15.6.7 Troubleshooting *i*RCalibration Frame Set

The following errors might occur during the search motion. Refer to Table 15–26 .

Table 15–26. Troubleshooting *i*RCalibration Frame Set

Error	Cause	Remedy
Robot does not make contact with the part.	Search starts too far from part.	• Increase the search distance if the robot is moving along a correct search direction • Re-teach the search start position so it is closer to the part
	Search direction is not correct.	• Change the search direction in the FIND instruction to make sure the robot is moving to approach the part • Re-orient the tool so a TOOL search is in the right direction
Excessive vibration when robot is returning from the contact position back to the search start position.	Motion speed is too high.	• Reduce the return speed.
Frame correction is inconsistent from search to search (menu or automatic).	Search points too close.	Make sure that search points with the same search direction are far apart from each other.
	Irregular part or part with changing dimensions.	• Use more search points • Search part from opposing sides if possible • Make sure contact surface is as smooth as possible in the vicinity of the contact point • Make sure that the same surface is contacted that was contacted when mastering was done
Frame correction is inconsistent from search to search (Menu operation).	Convergence tolerance is too high	Lower the convergence tolerance, be sure to allow for part deviation if applicable.

Table 15–26. Troubleshooting *i*RCalibration Frame Set (Cont'd)

Error	Cause	Remedy
Frame correction is inconsistent from search to search (automatic program operation)	Search iteration is not done.	• Call CELLFIND multiple times • Use register defined in schedule to check for convergence tolerance and repeat CELLFIND call until register value is within tolerance
	Program master had one or more bad searches.	• Make sure during mastering that good contact is made with the proper surface • Make sure no false detections occurred during mastering
	TCP has changed since program master was done.	• Highly recommend that TCPMate is used in conjunction with *i*RCalibration Frame Set to ensure TCP is properly corrected • Run TCPMate before running *i*RCalibration Frame Set • Make sure TCPMate is mastered before or at the same time *i*RCalibration Frame Set is mastered.
	Contact surfaces have changed dimensions since program mastering.	• Increase convergence tolerance • Choose contact surfaces that do not change dimension with respect to each other • Use more search points and include opposing surfaces

15.7 iRCALIBRATION FRAME SHIFT

15.7.1 Overview

*i*RCalibration Frame Shift, formerly CellCal, is a calibration option that compensates for frame changes by establishing a MASTER frame that runs a frame search teach pendant program. After the frame is changed, *i*RCalibration Frame Shift runs the same frame search teach pendant program again and computes the frame offset.

*I*RCalibration Frame Shift allows you to set up a cell, work, or part frame by using Torque Observer Sensor technology or Digital I/O using contact or proximity sensing. *I*RCalibration Frame Shift provides a simple utility to compensate for change of tooling during production. *i*RCalibration Frame Shift reduces production down time by eliminating the time-consuming process of touching up all positions in all part programs.

*i*RCalibration Frame Shift also makes cell cloning possible by compensating for the differences in tooling in a different cell. By combining *i*RCalibration Frame Shift with *I*RCalibration Master Set and *i*RCalibration TCP Set software, a part program that has been taught in Cell 1 can usually be transferred and run in Cell 2 without any touch-ups.

Limitations

The following limitations apply to *i*RCalibration Frame Shift.

- *i*RCalibration Frame Shift cannot be used with Line Tracking and other options that use a different run-time reference frame other than the current $MNUFRAME. Coordinated motion is supported only when Dynamic-UFrame is used. Refer to Section 15.7.5 and the *Coordinated Motion Setup and Operations* manual for more information.
- *i*RCalibration Frame Shift instructions will not be executed in single step test run mode.
- The TCP must be located near the contact surface of the tool. The relationship between the TCP and the contact surface cannot change from the time a *i*RCalibration Frame Shift program is mastered until it is used. iRCalibration TCP Set can be used, and is recommended, to maintain this relationship between the TCP and the physical tooling.
- The TCP can change in a *i*RCalibration Frame Shift program to reflect a different contact surface on the tool. The UFrame cannot change in a *i*RCalibration Frame Shift program.

Compatibility

*i*RCalibration Frame Shift is compatible with the following:

- *i*RCalibration Frame Shift can be used with robots with extended axes.
- *i*RCalibration Frame Shift can be used with multiple groups in single-group programs.

Hardware Requirements

The following hardware is required to use *i*RCalibration Frame Shift.

- Teaching tip, if desired (highly recommended for arc welding)
- I/O circuitry, if used

Software Requirements

The following software is required to use *i*RCalibration Frame Shift.

- *i*RCalibration Frame Shift software option

Installation

You must install the *i*RCalibration Frame Shift software option before you can use *i*RCalibration Frame Shift. Refer to the *Software Installation Manual* for more information about installing software options.

You do not need to install any hardware if TOS (Torque Observer Sensor) is used for contact detection. You will need to set up I/O ports for contact detection if you use I/O for contact detection. Refer to the "I/O Setup" chapter in this manual for information on setting up I/O ports.

You might also need to install a teaching tip as the robot's TCP before running the frame search teach pendant program, and take the teaching tip out after the frame search teach pendant program is complete.

15.7.2 Preliminary Setup

FANUC Robotics recommends that you set up *i*RCalibration Frame Shift prior to starting production.

Note *i*RCalibration Frame Shift always uses the current program's $MNUTOOL and $MNUFRAME for searching and position recording.

Perform the following steps prior to using *i*RCalibration Frame Shift:

1. Define a Tool Center Point (TCP), including WPR if you will be performing tool-relative searches. Refer to the "Frames Setup section" or Section 15.4 , "*i*RCalibration TCP Shift". For Arc Welding, refer to 6 Point UTOOL setup in the application-specific *Setup and Operations Manual*.
2. Install a teaching tip, if desired. This is highly recommended for arc welding and other applications in which the tooling is not conducive to repetitive contact with a part.
3. You must define a UFRAME. $MNUFRAMENUM[group_num] for that particular group must be non-zero This UFRAME will be adjusted when recovery is done to compensate for a change in part or tooling location.

4. If you are using I/O for detection, install and set up circuitry.
5. If you are using Analog Sensor, you must have a data file that resides in the FRS: directory. The data file has a fix name: cb_ainX,dt, where X is a digit from 1 to 9., The data file is very similar to the analog data file supported in the touch sensing option. It has the following format:

```
AinPortNumber 2
MaxAnalogInput 1000
MinAnalogInput 100
TotalDataPair 4
DistSensorToPart  AnalogInput
400.0             100
200.0             400
100.0             700
-100.0            1000
```

The maximum allowed TotalDataPair is 10

6. Set up *i*RCalibration TCP Shift for maintaining the same relationship between the tool frame and the physical TCP. Since *i*RCalibration Frame Shift determines the UFRAME based on touching surfaces with the robot tool, any misalignment of the tool frame and physical TCP will adversely affect the accuracy of the recovered UFRAME. *i*RCalibration TCP Shift should be used to recover the tool frame after every minor collision. Very often there is a slight change to the tool even when there seems to be no damage.

Note Refer to Figure 15–30 for Arc Weld tool definition and Figure 15–31 for a typical Waterjet tool definition.

Figure 15–30. Typical Arc Weld Tool Definition

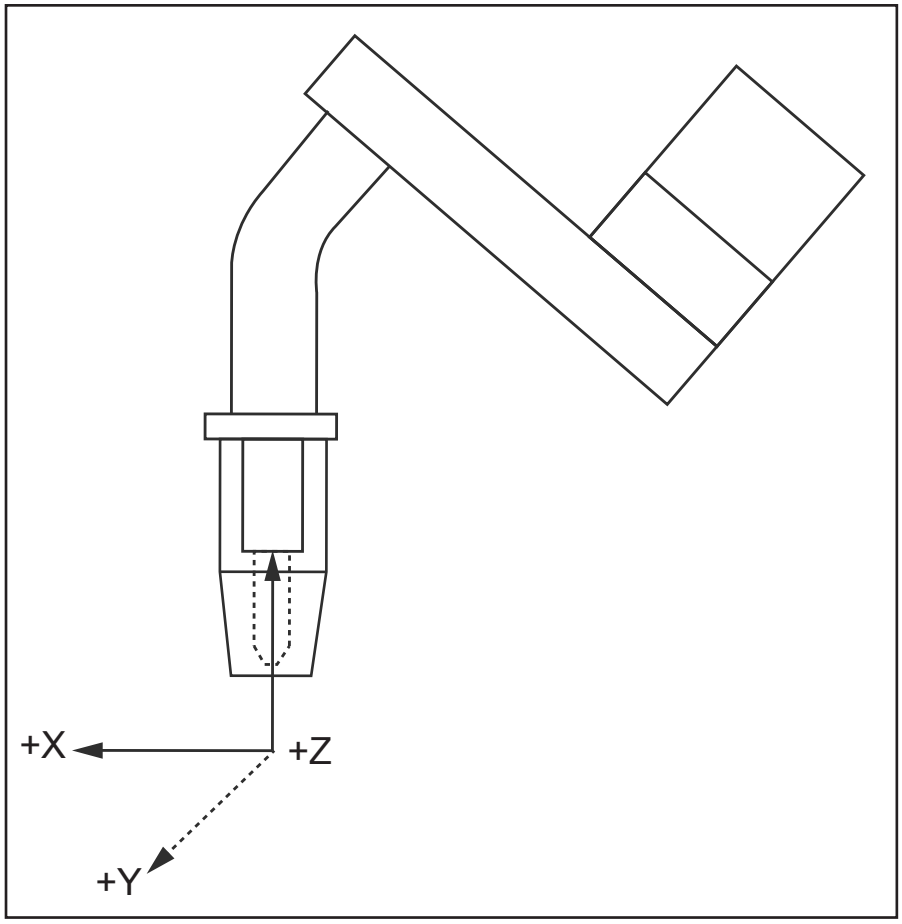
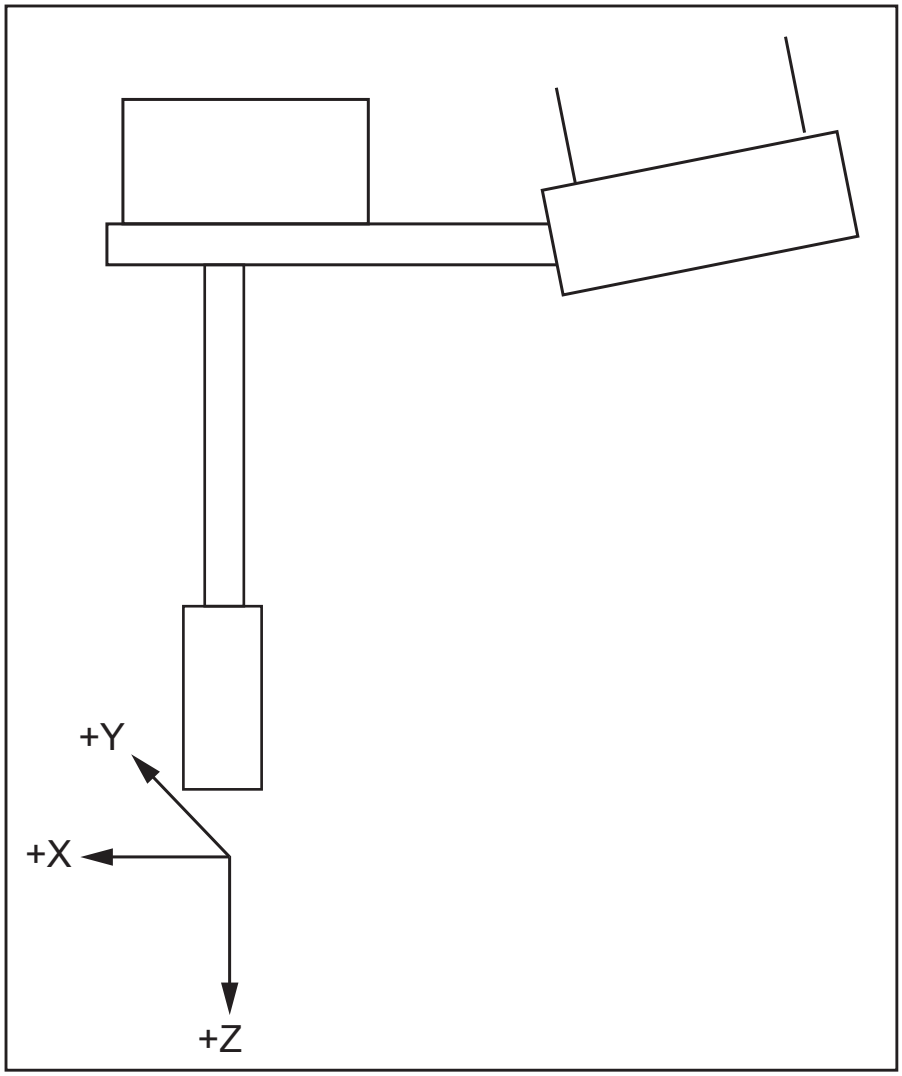


Figure 15–31. Typical Waterjet Tool Definition



15.7.3 *i*RCalibration Frame Shift Setup

Before you can use *i*RCalibration Frame Shift to master or recover a frame, you must perform some setup procedures. Setup involves the initial setting of menu options within *i*RCalibration Frame Shift software.

Refer to Section 15.7.2 for preliminary setup recommendations.

Setting up and running *i*RCalibration Frame Shift includes the following processes:

1. Selecting the motion group you want to calibrate

2. Selecting the teach pendant program, the UFRAME you want to calibrate, and the calibration mode (standard or dynamic UFrame). Refer to Section 15.7.5 for information about using *i*RCalibration Frame Shift with dynamic UFrame.
3. Setting up the Frame schedule. Refer to Table 15–27 for a description of the items on the Frame Schedule Detail Screen.
4. Setting up the detection schedule. Refer to Table 15–28 for a description of the items on the *i*RCalibration Frame Shift Detection Schedule Detail Screen.
5. Teaching points for the teach pendant calibration program. Refer to Procedure 15-32 .
6. Creating the teach pendant calibration program from the taught points and running the calibration program. Refer to Section 15.7.4 .
7. Updating the UFrame with the calibration results obtained from running the calibration program. Refer to Procedure 15-35 .

Table 15–27. Frame Schedule Detail Schedule Setup

ITEM	DESCRIPTION
Auto Update Value: TRUE	This item indicates whether the *i*RCalibration Frame Shift FIND END instruction will update the corresponding $mnuframe or not. If you set Auto Update to TRUE and run the *i*RCalibration Frame Shift teach pendant program, the current $mnuframe will be updated based on the offset.
Store Offset Value: TRUE or FALSE Default: FALSE	This item stores the cell calibration offset to a specific position register.
in PR[0]	If this is non-zero and "Store Offset" is TRUE, then the UFrame offset is stored in the specified position register.
Max XYZ Offset: Min: 0.01 mm Max: 100 mm Default: 2.0 mm	This item indicates the tolerance for any detection point shifting. If the distance between the nominal (reference) detection point and the newly detected point is larger than this value, *i*RCalibration Frame Shift will post a warning message.
Report on DOUT[0]:	If this is non-zero and *i*RCalibration Frame Shift detects an error then this output port will be pulsed high for 500 milliseconds.
Fit Error Register.	If this item is non-zero then *i*RCalibration Frame Shift will store the mean fit error of the detection points in this register. This value is the average deviation from nominal in the search direction of each detection point after the offset is applied
Max Orient Change: Min: 0.01 degrees Max: 10 degrees Default: 2.0 degrees	This item indicates the maximum tolerance for orientation changes. If the orientation of the frame exceeds this tolerance, *i*RCalibration Frame Shift will post a warning message.

Table 15–27. Frame Schedule Detail Schedule Setup (Cont'd)

Mean Error Register Range: 0 — 999 Default: 0	If the register number is greater than 0, *i*RCalibration Frame Shift writes the mean error to this register.
Max. Error Register Range: 0 — 999 Default: 0	If this item is greater than 0, *i*RCalibration Frame Shift writes the computed maximum error to this register.
Max Dev Register:	If this item is non-zero, *i*RCalibration Frame Shift will store the maximum deviation in this register. This is the distance between any of the nominal (reference) detection points and the corresponding newly detected point before the offset is applied.
CD Pair Number: Range: 0-4	This item is only used in conjunction with the Coordinated Motion option. If this item is non-zero, *i*RCalibration Frame Shift will update the leader frame associated with the specified CD Pair Number. Dynamic-UFrame must be enabled and an active follower UFrame must be set. The Dynamic-UFrame feature of Coordinated Motion can be enabled by setting the system variable $CD_PARAM.$DYN_UFRAME=TRUE. The default is FALSE. Once Dynamic-UFrame is enabled, and a CD-Pair leader group and follower group are selected a menu item appears on the Coordinated Motion Setup menu:, "Follower UFrame Number". This value determines the active dynamic UFrame for the CD pair. By default, this dynamic UFrame is relative to the Leader Frame associated with the CD Pair. Refer to the *Coordinated Motion Setup and Operations* manual for more information.

When you set up a detection schedule, you

- Determine the sensor
- Determine the detection motion
- Determine TOS

Note Default values are usually acceptable.

Table 15–28. *i*RCalibration Frame Shift Detection Schedule Description

ITEM	DESCRIPTION
Sensor Type Values: TOS Wrist, I/O, TOS ALL AXES Default: TOS ALL AXES	This item indicates the kind of sensor used for contact detection. The modes are: • **TOS Wrist** (Torque Observer Sensor on Wrist) - the contact torque will be monitored, and only the observed torque on the wrist will be used to determine the contact position. When TOS Wrist is selected, no I/O connection is required for contact detection. TOS Wrist is very sensitive and can be more prone to false detections in certain cases than the more robust TOS All Axes method. TOS Wrist is not recommended except for special cases where it can verified that false detections will not occur. • **I/O** — the specified I/O will be monitored and only the state of this input will be used to determine the contact position. Use the detection port for contact detection with I/O. Refer to Detection Port item in this table for more information. • **TOS ALL AXES** (Torque Observer Sensor, All Axes) - The contact torque will be monitored on all robot axes, and only the observed torque will be used to determine the contact position. This is the default value. When TOS ALL AXES is selected, no I/O connection is required for contact detection.
Detection Port Values: RDI, DI, WDI, WSI, AIN	This item indicates the desired input port type. The port type that you specify is followed by the desired port number that will go from OFF to ON when contact is detected. The WDI and WSI ports are typically used for arc welding. You will set this item up if you selected I/O as the Sensor Type. You do not need to set this up if you selected TOS Wrist or TOS All Axes as the Sensor Type. The port types are: • **RDI** — Robot Digital Input • **DI** - normal Digital Input • **WDI** - Weld Digital Input • **WSI** - Wire Stick circuit Input. • **AIN** - Analog Input. To enable analog sensor support, you must have a data file that resides in the FRS: directory before running *i*RCalibration Frame Shift. The data file has a fixed name: cb_ainX,dt, where X is a digit from 1 to 9. Set the port number to be the same as AinPortNumber setting in the data file. Refer to the Preliminary Setup section for more information.

Table 15–28. *i*RCalibration Frame Shift Detection Schedule Description (Cont'd)

ITEM	DESCRIPTION
Sensor Enable Values: RDO, DOUT, WDO, or WSO Default: RDO	This item indicates the I/O sensor enable digital output port type. The port types are: • **RDO** - Robot Digital Output • **DOUT** - normal Digital Output • **WDO** - Weld Digital Output • **WSO** - Wire Stick circuit Output. The port type that you specify is followed by the desired port number that will be used to enable the I/O detection circuit. The WDO and WSO ports are typically used for arc welding. You will set this item up if you selected I/O as the Sensor Type. Set the port number to 0 (zero) if you are not using this feature.
Detection Speed Units: mm/sec Range: 1.0 - 100 Default: 10	This item is the search speed for contact detection. This speed is used for both TOS and I/O searches. The value can be increased if cycle time is of great concern (although increasing the speed will also affect the resultant accuracy and repeatability) , or reduced if vibration or other disturbances cause *i*RCalibration Frame Shift to retry excessively.
Detection Distance Units: mm Range: 1 - 1000 Default: 50	This item is the maximum distance that the robot will move while looking for the signal. If no contact is detected within the distance, an error will be posted.
Return Speed Units: mm/sec Range: 5 - 5000 Default: 50	This item is the search return speed. This speed is used for both TOS and I/O searches.
Return to Start Default: Yes	When this item is set to YES, the robot will move back to the position before it starts the search motion. This item applies to DETECT POINT instructions only. When this item is set to NO, the robot will move back to a distance specified by the next item (Return distance)
Return Distance Units: mm Range: 1 - 1000 Default: 50	When return to start is set to FALSE, the robot will move back to the distance specified.
Return term type Default: FINE	This item is the termination type for return motion.
Air Cut Speed Units: mm/sec Range: 1 - 1000 Default: 100	This item is the non-search motion speed.

Table 15–28. *i*RCalibration Frame Shift Detection Schedule Description (Cont'd)

ITEM	DESCRIPTION
Learn Distance Units: mm Range: 0.5– 20.0 Default: 5.0	For "TOS All Axes" method only, there is a short motion before each search to identify dynamic torque values. Learn Distance is the distance in millimeters of this short motion, the direction is determined at the time of the motion.
Learn Speed Units: mm/sec Range: 0.5 – 50.0 Default: 10.0	For "TOS All Axes" method only, this is the speed of the learning motion before each search.
Sensitivity Tolerance Units: max. motor torque/sec. Range: 1 - 999 Default: 10	This item is the primary sensitivity tolerance of TOS Wrist and is not used for TOS All-Axes. It is the rate of change of the disturbance torque. The actual units vary with motor model, gear ratio, and amplifier current, and are affected by tooling length. However, the effect for calibration is largely independent of these factors and a value of 10 is considered normal operation for typical installations. You should not use a value of less than 10 or misdetections might occur. This value can be increased if false detections occur frequently. **Warning** **Be careful when you change this number. Changing the sensitivity tolerance might cause a false detection. This could injure personnel or damage equipment.**
Torque Threshold Units: max. motor torque Range: 1 to 999 Default: 100	For TOS All Axes, this item is the only measure of contact detection. During the learning motion a torque level is memorized and that level plus Torque Threshold must be exceeded for contact to be determined. If false detections are being experienced then this value should be increased. For TOS Wrist, this item is the level of disturbance torque that is used as another means for contact detection. The default value of 100 is considered normal operation for typical installations. If increasing the value of **Sensitivity Tolerance** does not make false detections disappear, then this value should be increased until the false detections disappear; then, reset the value of **Sensitivity Tolerance** .
Fit Tolerance Units: mm Range: 0.1 - 100 Default: 0.5	For Detect Circle instruction: The robot detects 4 points inside the circle and the calibration software will fit a circle to the 4 detected positions. The fit error is defined as the maximum distance from the 4 points to the circle. If the fitting error exceeds the tolerance, the calibration software will post an error.
Radius Tolerance Units: mm Range: 0.1— 100 Default: 0.5	The Detect Circle instruction stores the circle's radius during mastering. At run time, if the computed circle radius differs from the stored radius value and difference exceeds the tolerance, the calibration software will post an error.

Table 15–28. *i*RCalibration Frame Shift Detection Schedule Description (Cont'd)

ITEM	DESCRIPTION
Number or Retry Range: 0 — 9 Default:	If the number of retry is >0, the calibration software will re-run the detect circle instruction if the current circle detection has either a fit error or a radius error.
Contact Position Register	If the position register number is any number other than 0, *i*RCalibration Frame Shift will copy the contact position to the position register.

15.7.3.1 Setting up *i*RCalibration Frame Shift

*i*RCalibration Frame Shift uses a wizard-style menu. It shows the steps that need be done to complete the Cell calibration process. Each step of the process has its own status display that shows where you are in the setup process. You cannot skip steps and you cannot go to the next step without completing the current step.

*i*RCalibration Frame Shift also includes a menu item, View Result, that lets you review the previous calibration result and log history.

Procedure 15-31 Setting up *i*RCalibration Frame Shift

- The *i*RCalibration Frame Shift software option is loaded.
- Preliminary setup is complete. Refer to Section 15.7.2 .

Steps

1. Press MENUS.
2. Select UTILITY.
3. Press F1, [TYPE].
4. Select iRCalibration. You will see a screen similar to the following:

```
UTILITIES iRCalibration
   1 TCP Shift
   2 TCP Set
   3 Frame Shift
   4 Frame Set
   5 Master Recovery
   6 Master Set
```

5. Select *i*RCalibration Frame Shift and press ENTER. You will see a screen similar to the following:

```
iRCalibration Frame Shift
        1 Robot: Group [1]  Device: Sensor
          Calibration Type: UFrame Recovery
        2 Setup            (Not Done)
        3 Teach Points     (Not Done)
        4 Create Program   (Not Done)
        5 Run: CCAL_G1     (Not Done)
        6 Update UFrame[ 0](Not Done)
        7 View Result
```

6. Move the cursor to Robot: Group [1], type the group number of the motion group you want to calibrate, and press ENTER.

Note Calibration Type is always UFrame Recovery. You cannot change it.

If you are not using analog sensor, proceed to Step 7 .

7. **If you are using analog sensor,** you must have a data file that resides in the FRS: directory before you run *i*RCalibration Frame Shift. The data file has a fixed name: cb_ainX,dt, where X is a digit from 1 to 9., The data file is very similar to the analog data file supported in the touch sensing option. It has the following format:

```
AinPortNumber 2
MaxAnalogInput 1000
MinAnalogInput 100
TotalDataPair 4
DistSensorToPart  AnalogInput
400.0             100
200.0             400
100.0             700
-100.0            1000
```

8. **Specify the Calibration Program Name**. The *i*RCalibration Frame Shift Setup menu allows you to specify the Cell calibration teach pendant program name, the UFrame number, and the calibration mode, and select the *i*RCalibration Frame Shift Frame schedule and the detection schedule.

Note The Group number is provided for reference. You cannot change it in this menu.

a. Move the cursor to Setup and press ENTER. You will see a screen similar to the following:

```
iRCalibration Frame Setup
        1 Program name: CCAL_G1    Group[1]
        2 Uframe Number:    0
        3 Calibration Mode: Standard
        4 Frame Schedule:  1
        5 Detect Schedule:  1
```

b. The default program name is CCAL_Gx, where x is the group number. If you want to specify a different program name, press ENTER and use the function keys, or press SELECT to select it from the list of programs. Refer to the "Writing and Modifying a Program" section of your application-specific Setup and Operations manual for more information.

9. Select the UFrame. Move the cursor to UFrame Number:, type the number of the User Frame you want to calibrate, and press ENTER.

10. Select the Calibration Mode. Move the cursor to Calibration Mode. Press F4, CHOICE and select Standard or Dynamic-UFrame.

- Standard will perform User Frame Recovery.
- Dynamic will set up the Dynamic User Frame between a coordinated pair. For information on Dynamic UFrame calibration mode, refer to Section 15.7.5 .

11. Select the Frame Schedule and Set Frame Schedule detail items. Move the cursor to Frame Schedule.

- To change the number of the Uframe, move the cursor to the schedule number, type the number of the Frame Schedule you want, and press ENTER.
- To see the detail screen for the selected uFrame, press ENTER. You will see a screen similar to the following:

```
Frame Schedule                1/9
   Schedule Number[  1]
   1 Auto Upate:            TRUE
   2 Store Offset: FALSE
   3               In PR[0  ]
   4 Max. XYZ Offset:      30.000 mm
   5      Report on DOUT[0  ]
   6 Max. Orient Offset:   5.000  deg
   7 Fit Error Register:   0
   8 Max. Dev. Register:   0
   9 CD Pair Number:       0
```

Set each item as desired. For information on the Frame Schedule Setup items, refer to the *General Setup* chapter for more information.

Note The default settings are good for most applications.

When you have finished setting up the Frame Schedule, press PREV to return to the *i*RCalibration Frame Shift Setup Menu.

```
iRCalibration Frame Shiftup
        1 Program name: CCAL_G1    Group[1]
        2 Uframe Number:    0
        3 Calibration Mode: Standard
        4 Frame Schedule:  1
        5 Detect Schedule:  1
```

12. **Select the Detection Schedule and Set Detection Schedule detail items.** Move the cursor to Detect Schedule.

    - To change the number of the Detect Schedule, move the cursor to the schedule number, type the number of the schedule you want, and press ENTER.
    - To set up the detection schedule detail items, press F4, SELECT. You will see a screen similar to the following:

```
Detection Schedule
                                        1/18
   Detection Schedule [ 1]
   Sensor Type:               TOS ALL AXES
   I/O Setup
     Detection Port:         DI   [  1]
     Sensor Enable:          DOUT[  1]
   Motion Setup
     Detection Speed:        10.000  mm/sec
     Detection Distance:     50.000  mm
     Return Speed:           50.000  mm/sec
     Return to Start:        TRUE
       Return Distance:      0.000   mm
     Return Term Type:       Fine
     Learn Speed:            10.000  mm/sec
     Learn distance:         5.000   mm
     Air Cut Speed:          100.000 mm/sec
   TOS Setup
     Sensitivity:            10.000
     Torque Threshold:       100
   Circle Setup
     Fit Tolerance:         1.000   mm
     Radius Tolerance:      1.000   mm
     Number of Retry:       0
   Contact Position Register:  0
```

Set each item as desired. Move the cursor to the item you want to change and type the value.

Note The default settings are good for most applications.

Refer to Table 15–28 for information about detection schedule setup detail items.

13. When setup is complete, press F1, DONE. You will see the *i*RCalibration Frame Shift main menu with Setup (Done) in its status display.

Note You must press DONE to change the Setup status. If you press PREV to return to the *i*RCalibration Frame Shift main menu, the Setup status display will not change.

```
iRCalibration Frame Shift
        1 Robot: Group [1]  Device: Sensor
          Calibration Type: UFrame Recovery
        2 Setup            (Done    )
        3 Teach Points     (Not Done)
        4 Create Program   (Not Done)
        5 Run: CCAL_G1     (Not Done)
        6 Update UFrame[ 0](Not Done)
        7 View Result
```

If the teach pendant program already exists when you press DONE, you will see the following screen:

```
Program exists. Skip Teach/Create?
  Press F2 for YES
  Press F3 for NO
```

If the teach pendant program already exists and has been taught, and you want to skip the teach/create step, press F2, YES, and proceed directly to Run. Refer to Procedure 15-34 to run the program.

If the teach pendant program exists but you want to teach it, change it, or use another program, press F3, NO, and continue to Section 15.7.3.2 .

15.7.3.2 Creating a Frame Calibration Teach Pendant Program

In order to master a frame with *i*RCalibration Frame Shift, you must first create a frame calibration teach pendant program. You can create the program manually or you can have *i*RCalibration Frame Shift create the program for you automatically. The frame calibration teach pendant program will detect contact with the fixture for mastering and recovery of the frame. You need one program for each frame you master.

When *i*RCalibration Frame Shift setup is complete, the *i*RCalibration Frame Shift main menu will be displayed with the cursor at Teach Points.

```
iRCalibration Frame Shift
        1 Robot: Group [1]  Device: Sensor
          Calibration Type: UFrame Recovery
        2 Setup            (Done    )
        3 Teach Points     (Not Done)
        4 Create Program   (Not Done)
        5 Run: CCAL_G1     (Not Done)
        6 Update UFrame[ 0](Not Done)
        7 View Result
```

You can let *i*RCalibration Frame Shift create the teach pendant program automatically, or you can create your own teach pendant program.

- **To have *i*RCalibration Frame Shift create the teach pendant program automatically**, continue to Procedure 15-32 .
- **To create the teach pendant program manually**, refer to Procedure 15-33 .

Procedure 15-32 Teaching Points (Creating the *i*RCalibration Frame Shift Teach Pendant Program Automatically)

Condition

- *i*RCalibration Frame Shift Setup has been completed.
- The teach pendant program does not already exist, or you want to overwrite the existing program.

1. When you have completed *i*RCalibration Frame Shift setup, the Setup status on the *i*RCalibration Frame Shift main menu will be Done.

```
iRCalibration Frame Shift
        1 Robot: Group [1]  Device: Sensor
          Calibration Type: UFrame Recovery
        2 Setup            (Done    )
        3 Teach Points     (Not Done)
        4 Create Program   (Not Done)
        5 Run: CCAL_G1     (Not Done)
        6 Update UFrame[ 0](Not Done)
        7 View Result
```

Move the cursor to Teach Points and press ENTER. You will see a screen similar to the following:

```
Standard iRCalibration Frame Shift Points
                                                1/8
    Program Name:          CCAL_G1
    Robot Group:       1
    iRCalibration Frame Shift Schedule:  1
    Detection Sched:   1

    Home Position: UNINIT
    P1:Via:[UNINIT  ]Srch:[UNINIT  ]Dir:X
    P2:Via:[UNINIT  ]Srch:[UNINIT  ]Dir:X
    P3:Via:[UNINIT  ]Srch:[UNINIT  ]Dir:X
    P4:Via:[UNINIT  ]Srch:[UNINIT  ]Dir:X
    P5:Via:[UNINIT  ]Srch:[UNINIT  ]Dir:X
    65:Via:[UNINIT  ]Srch:[UNINIT  ]Dir:X
    Via To Home Position: UNINIT
```

2. To clear all recorded positions, press F1, CLEAR, then F2, YES.
3. Teach the positions.

   - **If the teach pendant program already exists**, press F2, LOAD, to load the positions to the menu's position data.
   - **If the teach pendant program does not exist,** teach the Home position (the position where the *i*RCalibration Frame Shift program will start and finish) and each of the six Via points and Search points by using the F4, MV_JNT, and F5, MV_LIN, keys to move to positions.

     For each search point choose a search direction of +/-X, +/-Y, or +/-Z. Each search direction will be with respect to UFrame. You can later change the generated calibration program to search with respect to UTool if that is a better search direction for your application. Searching with respect to World in a *i*RCalibration Frame Shift program is not recommended

     Press F3, RECORD, to record the positions.

     Note When Dynamic-Uframe is selected, the *i*RCalibration Frame Shift frame updates the Leader Frame for the coordinated motion pair, "CD Pair Number," identified in the Frame Schedule. Refer to the *Coordinated Motion* manual for information on how to enable the Dynamic Uframe feature that makes the follower Uframe relative to the coordinated motion pair leader group position.

4. When you have finished, press PREV to return to the *i*RCalibration Frame Shift Main Menu.

```
iRCalibration Frame Shift
        1 Robot: Group [1]  Device: Sensor
          Calibration Type: UFrame Recovery
        2 Setup             (Done    )
        3 Teach Points      (Done    )
        4 Create Program    (Not Done)
        5 Run: CCAL_G1      (Not Done)
        6 Update UFrame[ 0](Not Done)
        7 View Result
```

If all required positions are taught, the Teach Points status will be **Done** and the cursor will move automatically to the next step. Otherwise, the cursor will stay at Teach Points and its status will remain unchanged.

5. To create the program, move the cursor to Create Program and press ENTER. When *i*RCalibration Frame Shift has created the program, the Create Program status will change to Done.

The following program is an example of a User frame recovery Teach Pendant Program generated by *i*RCalibration Frame Shift.

```
1:  UFRAME_NUM=1
   2:  UTOOL_NUM=1
   3:L P[1:Home Position] 100mm/sec FINE
   4:  UFrame Start[1]
   5:L P[2:Via Pos 1] 100mm/sec FINE
   6:L P[3:Search Pos 1] 100mm/sec FINE
   7:  Detect Point[4] UFrame[-Z]
   8:L P[4:Via Pos 2] 100mm/sec FINE
   9:L P[5:Search Pos 2] 100mm/sec FINE
  10:  Detect Point[4] UFrame[-Z]
  11:L P[6:Via Pos 3] 100mm/sec FINE
  12:L P[7:Search Pos 3] 100mm/sec FINE
  13:  Detect Point[4] UFrame[-Z]
  14:L P[8:Via Pos 4] 100mm/sec FINE
  15:L P[9:Search Pos 4] 100mm/sec FINE
  16:  Detect Point[4] UFrame[Y]
  17:L P[10:Via Pos 5] 100mm/sec FINE
  18:L P[11:Search Pos 5] 100mm/sec FINE
  19:  Detect Point[4] UFrame[Y]
  20:L P[12:Via Pos 6] 100mm/sec FINE
  21:L P[13:Search Pos 6] 100mm/sec FINE
  22:  Detect Point[4] UFrame[X]
  23:  UFrame End
  24:L P[14:Via Pos 7] 100mm/sec FINE
  25:L P[15:Home Position] 100mm/sec FINE
```

6. Single step through the *i*RCalibration Frame Shift calibration program in low override to make sure that there is no collision with any obstacle.

Guidelines for Creating a Frame Calibration Teach Pendant Program Manually

Be familiar with the following guidelines as you create the frame search teach pendant program.

- The contact surface should be relatively flat.
- The search direction must be approximately perpendicular to the contact surface.
- The robot will return to the starting point (or as specified by “return distance” in the detection schedule) after each search.
- Multiple searches (in different directions) can be done at the same starting point as long as the search direction is approximately perpendicular to the contact surface.
- If the contact surface is not approximately perpendicular to a UFRAME or WORLD axis then search frame must be TOOL. You must orient the tool so that one of the TOOL axes is perpendicular to the contact surface.
- The “Update Frame” instruction can be used to apply an incremental frame update.
  - The“Update Frame” instruction will always update the current UFRAME regardless of the value of the “Auto Update” field in the Frame Schedule.
  - When “Update Frame” is used, the final error and position change value only applies to the detection points since the last “ Update Frame” instruction was executed.
  - Multiple“Update Frame” instructions can be included in a program.
  - All searches before an “Update Frame” instruction must be either parallel or perpendicular to each other.
  - If “Update Frame” instruction is used, no more than three parallel searches can be executed before the instruction.
- Search directions can be mixed in the same program (UFRAME, UTOOL, or WORLD can be used).
- The UFRAME cannot be changed in the calibration program.
- The order that you teach the points does not matter, however, it is important that each search direction has enough points that are far from each other.

Figure 15–32. Non-uniform Surface (Example 1)

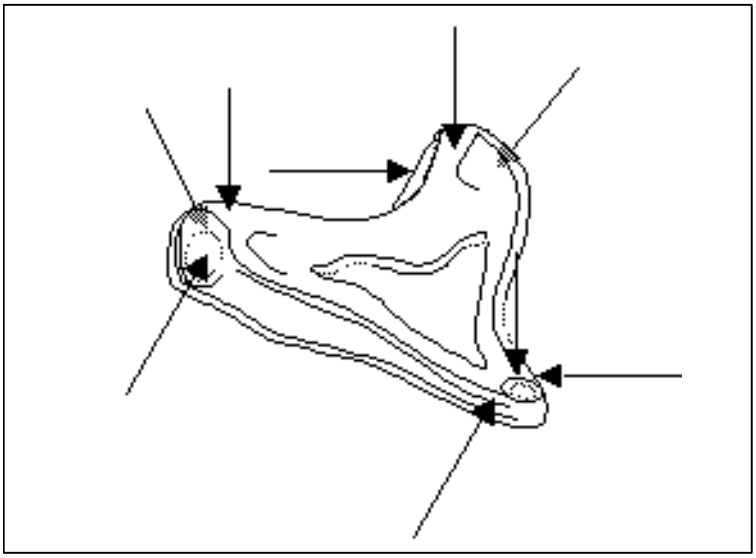
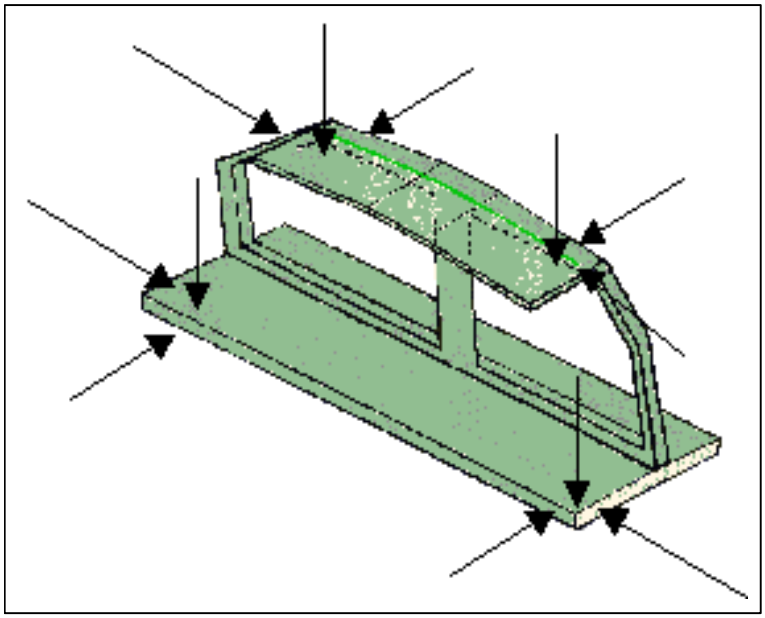


Figure 15–33. Non-uniform Surface (Example 2)



Procedure 15-33 Creating a Frame Calibration Teach Pendant Program Manually

Conditions

- The *i*RCalibration Frame Shift software option is loaded.
- You have set up a UFrame calibration and detection schedule. (Refer to Procedure 15-31 .)

you have completed *i*RCalibration Frame Shift Setup and the Setup status is Done.

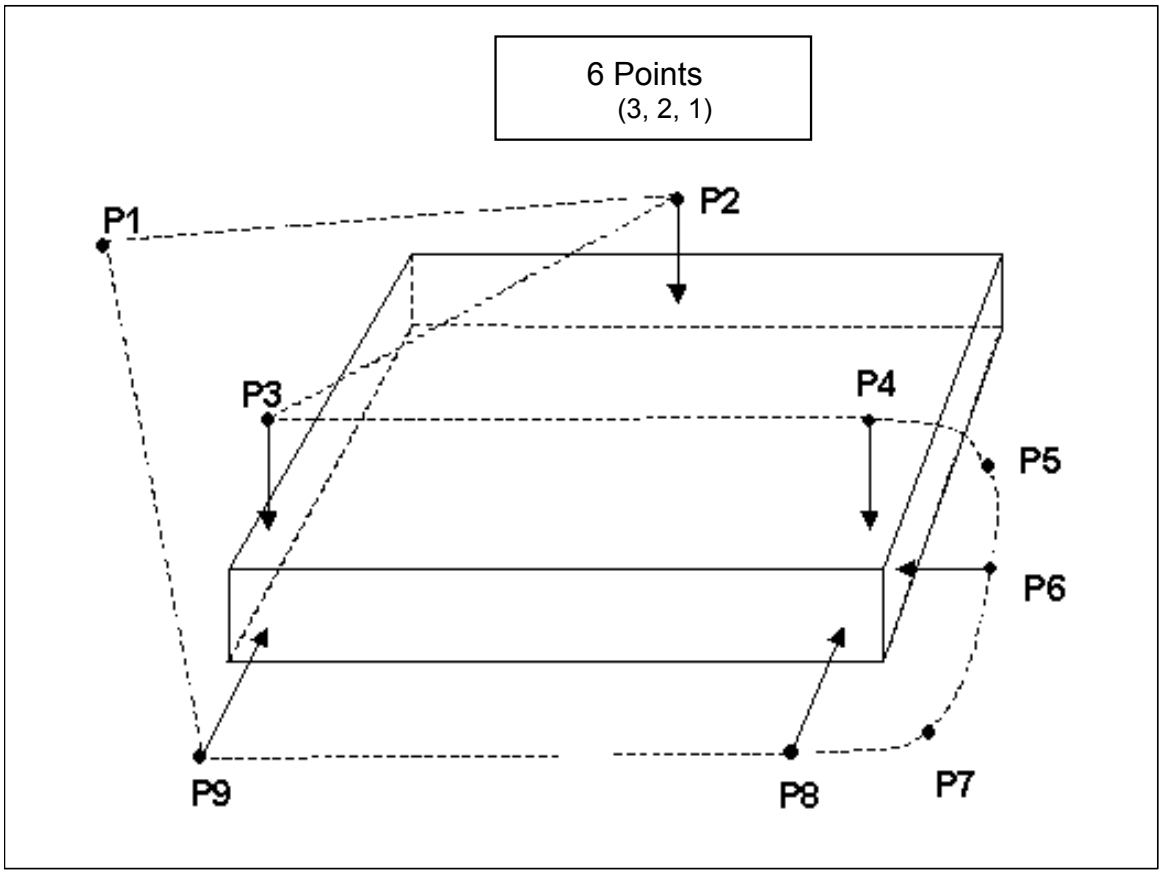
Steps

1. Press SELECT.
2. If F2, CREATE, is not displayed, press NEXT, >.
3. Continuously press the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.
4. Press F2, CREATE.
5. Create a teach pendant program. (See Figure 15–34 for a program example. Refer to the "Planning and Creating a Program" chapter for information on creating a teach pendant program.)

Figure 15–34. 6 Points Search (3-2-1) (Corresponds to Figure 15–35)

```
 1:  UFRAME_NUM=1
 2:  UTOOL_NUM=1
 3:L P[1:Home Position] 100mm/sec FINE
 4:  UFrame Start[1]
 5:L P[2:Search Pos 1] 100mm/sec FINE
 6:  Detect Point[4] UFrame[-Z]
 7:L P[3:Search Pos 2] 100mm/sec FINE
 8:  Detect Point[4] UFrame[-Z]
 9:L P[4:Search Pos 3] 100mm/sec FINE
10:  Detect Point[4] UFrame[-Z]
11:  Update Frame
12:L P[5:Via Pos 4] 100mm/sec FINE
13:L P[6:Search Pos 4] 100mm/sec FINE
14:  Detect Point[4] UFrame[-Y]
15:  Update Frame
16:L P[7:Via Pos 5] 100mm/sec FINE
17:L P[8:Search Pos 5] 100mm/sec FINE
18:  Detect Point[4] UFrame[X]
19:L P[9:Search Pos 6] 100mm/sec FINE
20:  Detect Point[4] UFrame[X]
21:  UFrame End  [Standard / Dynamic-UFrame]
22:L P[1:Home Position] 100mm/sec FINE
```

Figure 15–35. 6 points (3-2-1) for x, y, z, w, p, r Offset



This program is very similar to the one created automatically by *i*RCalibration Frame Shift. It has fewer Via points and has two extra instructions, Update Frame, on line 11 and 15.

- The first Update Frame instruction on line 11 allows the XY plane to be corrected for orientation. So, the searches in the X and Y directions are at the proper height with respect to the top of the plane.
- The second Update Frame instruction on line 15 provides an additional update for the new detection in the Y direction. This allows the detections on lines 19 and 20 to be at the proper location with respect to P6.

You can have as few as one and as many as 24 detections in a *i*RCalibration Frame Shift program, although it is rare to have more than 12. You can intermix Detect Circle and Detect Point instructions. If there are fewer than six points before Uframe End or before an Update Frame instruction, then the points must meet the following restrictions:

- All searches must be approximately perpendicular or parallel to each other
- There must be no more than three parallel searches
- There must be no more than two parallel searches that are colinear

- There must be significant distance between all parallel searches

It is recommended to repeat *i*RCalibration Frame Shift program execution until the max deviation is small.

Figure 15–36. Example program for automatic operation.

```
1:  R[1:Loop Counter]=0
   2:  LBL[1]
   3:  CALL ACELL2A
   4:  R[1:Loop Counter]=
       R[1:Loop Counter]+1
   5:  IF R[2:iRCalibration Frame Shift Max Dev]<=.5,
       JMP LBL[2]
   6:  IF R[1:Loop Counter]<5,
       JMP LBL[1]
   7:  MESSAGE[ERROR]
   8:  PAUSE
   9:  LBL[2]
```

The Maximum Deviation register is R[2]. In this example, if the maximum deviation is not less than 0.5mm in five iterations of the *i*RCalibration Frame Shift program ACELL2A then the program will pause and a message issued.

Note that if you manually create a *i*RCalibration Frame Shift program or a program that calls a *i*RCalibration Frame Shift program you must increase the stack size.

```
Select                 G2       JOINT 100 %
          666424 Bytes free        19/47
 No.  Program name      Comment
 15  ANEWTCP         [                    ]
 16  ATCP4           [                    ]
 17  ATCPXYZ         [                    ]
 18  ATERRJOB    VR  [                    ]
 19  AUTOCELL        [                    ]
 20  CBMOVTO4        [                    ]
 21  CB_SRCH1        [                    ]
 22  CB_SRCH2        [                    ]
 23  CB_SRCH4        [                    ]
```

6. Press Select and move the cursor to your program. Press F2, DETAIL

```
Program Detail        G2     JOINT 100 %
                                   1/7
 Creation Date:          21-FEB-2006
 Modification Date:      21-FEB-2006
 Copy Source:      [               ]
 Positions: FALSE  Size:    256 Byte
  1  Program name:        [AUTOCELL]
  2  Sub Type:     [None            ]
  3  Comment:      [                ]
  4  Group Mask:   [*,1,*,*,*,*,*,*]
  5  Write protect:       [OFF     ]
```

7. Press the Up-Arrow key and move the cursor to the Stack size field.

```
Program Detail        G2     JOINT 100 %
                                   1/7
 Copy Source:      [               ]
 Positions: FALSE  Size:    256 Byte
  1  Program name:        [AUTOCELL]
  2  Sub Type:     [None            ]
  3  Comment:      [                ]
  4  Group Mask:   [*,1,*,*,*,*,*,*]
  5  Write protect:       [OFF     ]
  6  Ignore pause:        [OFF     ]
  7  Stack size:          [     300]
```

8. Enter 1500 for the new Stack size and press F1, END.

```
Program Detail        G2     JOINT 100 %
                                   1/7
 Copy Source:      [               ]
 Positions: FALSE  Size:    256 Byte
  1  Program name:        [AUTOCELL]
  2  Sub Type:     [None            ]
  3  Comment:      [                ]
  4  Group Mask:   [*,1,*,*,*,*,*,*]
  5  Write protect:       [OFF     ]
  6  Ignore pause:        [OFF     ]
  7  Stack size:          [1500    ]
```

9.

10. Move the robot to the fixture of which the UFRAME is defined. Record positions along the path so the robot can safely arrive at the first search position.

11. Press F1, [INST].
12. Select Calibration.
13. Select UFRAME START[] instruction.
14. Enter the *i*RCalibration Frame Shift Schedule number.
15. Record a detection start position.
16. Add DETECT POINT[] instructions after the detection start position.
17. Repeat Step 15 and Step 16 the appropriate number of times.
18. Add UFRAME END instruction at the end of all the searches.
19. Move the robot back to home position.
20. Exit the teach pendant editor.

15.7.4 Running the *i*RCalibration Frame Shift Program and Updating the User Frame

In order to master a frame with *i*RCalibration Frame Shift, you must first create a frame calibration teach pendant program. The frame calibration teach pendant program will detect contact with the fixture for mastering and recovery of the frame.

Procedure 15-34 Running the *i*RCalibration Frame Shift Program

Conditions

- The *i*RCalibration Frame Shift Create Program status is Done.

Steps

1. At the *i*RCalibration Frame Shift Main Menu

```
iRCalibration Frame Shift
        1 Robot: Group [1]  Device: Sensor
          Calibration Type: UFrame Recovery
        2 Setup            (Done    )
        3 Teach Points     (Done    )
        4 Create Program   (Done    )
        5 Run: CCAL_G1     (Not Done)
        6 Update UFrame[ 0](Not Done)
        7 View Result
```

Press ENTER or F4, SELECT, to run the Cell calibration program.

2. You have the following choices:

- **If the teach pendant program has been mastered before**, you will see the following message:

```
OK to run recovery program?
  Press F2 for YES
  Press F3 for NO
  Press F5 for Remaster
```

To run the recovery program , press F2, YES.

To abort, press F3, NO.

To re-master the teach pendant program, press F5, REMASTER.

- **If the teach pendant program has not been mastered before**, you will see the following message:

```
OK to run calibration program?
  Press F2 for YES
  Press F3 for NO
```

To abort the process, press F3, NO.

To run the calibration program, press F2, YES. You will see the following screen:

```
Execute Calibration Program?
  Press F2 for YES
  Press F3 for NO
```

a. Be sure the teach pendant is ON and all faults are cleared.

b. Press SHIFT+F2 to run the teach pendant program.

It is recommended to run the *i*RCalibration Frame Shift program in low override to make sure there is no collision with any obstacle

The Run Status will change to Done after the teach pendant program has been run successfully.

```
iRCalibration Frame Shift
        1 Robot: Group [1]  Device: Sensor
          Calibration Type: UFrame Recovery
        2 Setup             (Done    )
        3 Teach Points      (Done    )
        4 Create Program    (Done    )
        5 Run: CCAL_G1      (Done    )
        6 Update UFrame[ 0](Not Done)
        7 View Result
```

The first time you run the recovery program, *i*RCalibration Frame Shift masters the UFrame. Once the UFrame is mastered, you can recover it by running the *i*RCalibration Frame Shift program again. For mastering, no update is required.

For UFrame recovery, move the cursor to View Result and press ENTER or F4, SELECT to check the user frame calibration result.

Procedure 15-35 Updating the User Frame

The Update UFrame menu is not accessible unless you have completed all the steps before it.

1. At the *i*RCalibration Frame Shift main menu, move the cursor to Update UFrame and press ENTER. You will see a screen similar to the following:

```
Frame Calibration Result

 UFrame Offset:[G:1, Num:  ]
   X: 988.805 Y:   -3.351 Z:  393.492
   W: -89.925 P:    0.073 R:    0.000

 UFrame XYZ Change:       0.217 mm
 UFrame Angular Change:   0.030 deg
 Mean Fit Error:          0.000 mm
 Maximum Deviation:       0.163 mm
```

2. Press F2, UPDATE, to update the group 1, user frame 1.

Note User frame is not automatically updated even its schedule has auto update set to TRUE. This is different from previous version of *i*RCalibration Frame Shift.

15.7.5 Using *i*RCalibration Frame Shift with Dynamic-Uframe

Dynamic-Uframe is a feature of coordinated motion where a specified Uframe of the follower group is attached to the Leader Frame of the leader group. When the leader group moves, the follower group's Uframe changes dynamically.

*i*RCalibration Frame Shift can work with a Dynamic-Uframe similarly to the normal static Uframe. For the case of Dynamic-Uframe, *i*RCalibration Frame Shift updates the Leader frame which is attached to the faceplate or positioner top of the leader group. This allows positioning errors of parts held by the leader to be compensated with respect to the leader faceplate or positioner top.

Once the Dynamic-Uframe is set up and enabled, *i*RCalibration Frame Shift programs can move both leader and follower groups. The detection instructions move only the follower group. However, both groups can move between detection instructions.

The Dynamic-Uframe *i*RCalibration Frame Shift program can also be a single group program that contains only the follower group. The *i*RCalibration Frame Shift program will use the current Dynamic-Uframe.

Refer to the *Coordinated Motion Setup and Operations* manual for more information on how to set up and use a Dynamic-Uframe.

15.7.6 Displaying the UFrame Log Screen

The UFrame log is a buffer that holds previous user frame or tool frame information. The buffer can hold up to 20 items. When the buffer is full, it will replace the oldest data in the buffer with the new logged data. If you want to view frame information, you can display the Frame Log screen. To display the frame log screen, use Procedure 15-36 .

Procedure 15-36 Displaying the UFrame Log Screen

Steps

1. Press MENU.
2. Select UTILITIES.
3. Select iRCalibration.
4. Select *i*RCalibration Frame Shift. You will see a screen similar to the following.

```
iRCalibration Frame Shift
        1 Robot: Group [1]  Device: Sensor
          Calibration Type: UFrame Recovery
        2 Setup             (Not Done)
        3 Teach Points      (Not Done)
        4 Create Program    (Not Done)
        5 Run: CCAL_G1      (Not Done)
        6 Update UFrame[ 0](Not Done)
        7 View Result
```

5. Move the cursor to View Result and press ENTER.

```
 Frame Calibration Result

  UFrame Offset:[G:1, Num:  ]
    X: 988.805 Y:   -3.351 Z:  393.492
    W: -89.925 P:    0.073 R:    0.000

  UFrame XYZ Change:       0.217 mm
  UFrame Angular Change:   0.030 deg
  Mean Fit Error:          0.000 mm
  Maximum Deviation:       0.163 mm
```

6. To display the UFrame Log, press F4, VIEW LOG. You will see a screen similar to the following

```
User Frame Log
                                        13/40
   Date     Time     X(W)     Y(P)     Z(R)
01-JAN-XX   16:57    989.3      -3.3   394.1
 UFrame Num[ 1]      -89.9        .0      .5
01-JAN-XX   16:57    989.3      -3.3   394.1
 UFrame Num[ 1]      -89.9        .0      .5
No XYZ Data
No WPR Data
No XYZ Data
No WPR Data
No XYZ Data
```

Each entry spans two lines. The first line lists the date and time of the saved entry as well as x, y, and z information. The second line contains the frame number and w, p, and r information.

If entries are not used or if they have been cleared, you will see the following lines:

NO XYZ DATA

NO WPR DATA

7. Select one of the following:

   - **If you want to save the entire log to the file "UFRM_LOG.DT,"** press F2, SAVE.
   - **If you want to restore a previous UFrame value,** move the cursor to the line that has the values you want to restore and press F3, APPLY. You will see a prompt box similar to the following.

```
User Frame Log
     Restore old Frame?
           YES        NO
```

   Move the cursor to YES and press ENTER.

   - **If you want to clear the log and remove all data entries,** press F4, CLEAR. You will see a screen similar to the following.

```
User Frame Log
     Clear All Logged Data?
           YES        NO
```

   Move the cursor to YES and press ENTER.

15.7.7 Troubleshooting *i*RCalibration Frame Shift

The following errors might occur during the search motion. Refer to Table 15–29 .

Table 15–29. Troubleshooting *i*RCalibration Frame Shift

Error	Cause	Remedy
Robot does not make contact with the part.	Search starts too far from part.	• Increase the search distance if the robot is moving along a correct search direction • Re-teach the search start position so it is closer to the part
	Search direction is not correct.	• Change the search direction in the FIND instruction to make sure the robot is moving to approach the part • Re-orient the tool so a TOOL search is in the right direction
Excessive vibration when robot is returning from the contact position back to the search start position.	Motion speed is too high.	• Reduce the return speed.
Frame correction is inconsistent from search to search (menu or automatic).	Search points too close.	Make sure that search points with the same search direction are far apart from each other.
	Irregular part or part with changing dimensions.	• Use more search points • Search part from opposing sides if possible • Make sure contact surface is as smooth as possible in the vicinity of the contact point • Make sure that the same surface is contacted that was contacted when mastering was done
Frame correction is inconsistent from search to search (Menu operation).	Convergence tolerance is too high	Lower the convergence tolerance, be sure to allow for part deviation if applicable.

Table 15–29. Troubleshooting *i*RCalibration Frame Shift (Cont'd)

Error	Cause	Remedy
Frame correction is inconsistent from search to search (automatic program operation)	Search iteration is not done.	• Call CELLFIND multiple times • Use register defined in schedule to check for convergence tolerance and repeat CELLFIND call until register value is within tolerance
	Program master had one or more bad searches.	• Make sure during mastering that good contact is made with the proper surface • Make sure no false detections occurred during mastering
	TCP has changed since program master was done.	• Highly recommend that TCPMate is used in conjunction with *i*RCalibration Frame Shift to ensure TCP is properly corrected • Run TCPMate before running *i*RCalibration Frame Shift • Make sure TCPMate is mastered before or at the same time *i*RCalibration Frame Shift is mastered.
	Contact surfaces have changed dimensions since program mastering.	• Increase convergence tolerance • Choose contact surfaces that do not change dimension with respect to each other • Use more search points and include opposing surfaces

Chapter 16

INTERFERENCE CHECKING

Contents

Chapter 16 INTERFERENCE CHECKING 16–1
16.1 OVERVIEW 16–3
16.1.1 Overview 16–3
16.1.2 Collision Avoidance 16–3
16.1.3 Multiple Controller Interference Checking 16–5
16.1.4 DualARM Single Controller Interference Checking 16–6
16.2 MODELING 16–7
16.2.1 Overview 16–7
16.2.2 Robot Arm Modeling 16–13
16.2.3 User Tool Modeling 16–18
16.2.4 Obstacle Modeling 16–26
16.2.5 How to Teach a Plane Obstacle Effectively 16–32
16.2.6 How to Teach a Box Obstacle Effectively 16–35
16.3 SETTING UP COLLISION CHECKING BETWEEN ROBOT ARM AND STATIC OBSTACLES 16–36
16.3.1 Overview 16–36
16.3.2 Setting Up Combination Pairs 16–37
16.4 SETTING UP INTERFERENCE CHECKING FOR MULTIPLE ROBOT APPLICATIONS 16–39
16.4.1 Overview 16–39
16.4.2 Host Robot and Priority List 16–39
16.4.3 Combination Check Pairs 16–43
16.4.4 Automatic Setup of Default Interference Zone Between Robot Pairs 16–47
16.5 SETTING UP DUALARM SINGLE CONTROLLER INTERFERENCE CHECKING 16–57
16.6 INTERFERENCE CHECK TP INSTRUCTIONS 16–58
16.7 STATUS AND RECOVERY 16–59
16.7.1 Overview 16–59

16.7.2 Recovery and Interference Checking 16–60
16.8 ROBOT PARAMETERS 16–61

16.1 OVERVIEW

16.1.1 Overview

The three main features of the Interference Checking option include:

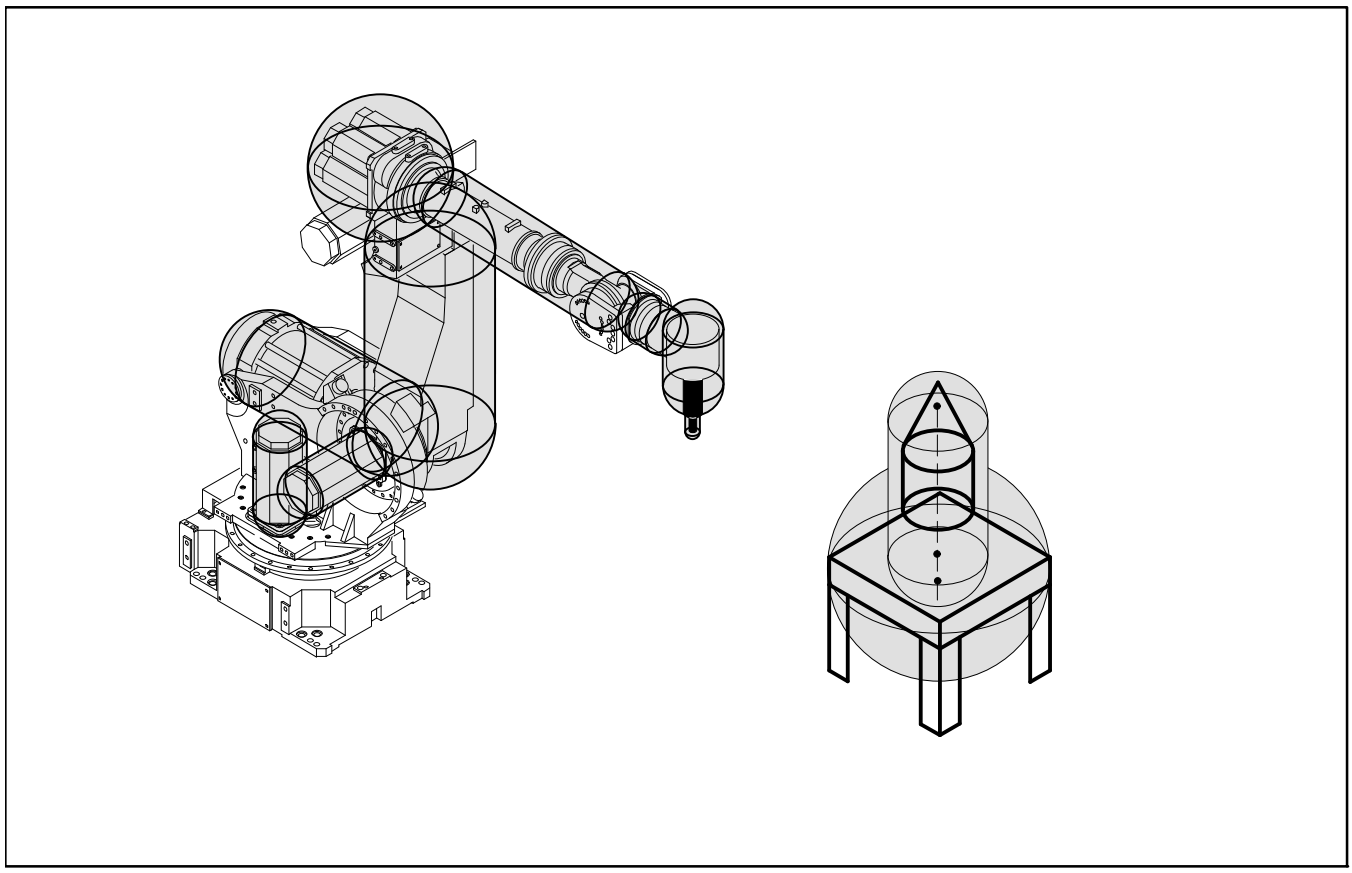
- Collision Avoidance
- Multiple Controller Interference Checking
- DualARM Single Controller Interference Checking

16.1.2 Collision Avoidance

The purpose of Collision Avoidance is to protect the robot arm (not just the TCP but the whole arm) and the user's tooling (attached to the robot) from colliding into a stationary fixture.

This is shown in Figure 16–1 .

Figure 16–1. Example of Collision Avoidance



After the *robot model* (which covers the robot), the *tool model* (which covers the user's tooling) and the *obstacle model* (which covers the static obstacles) are defined and enabled, the system will dynamically monitor potential collisions between the robot and tool models and the obstacle model(s). If interference is detected, the system will stop the robot in the same manner as a decelerated stop, and an alarm message will be posted.

All robot, tool, and obstacle models are defined as a combination of simple elements. The four kinds of simple elements include:

- Sphere element
- Cylinder element
- Plane element (Infinite and Finite Planes)
- Box element (Protect last element or TCP only)

The default robot model is set up automatically by the system. It covers the mechanical links of the robot up to the faceplate. Generally, the number of simple elements used to define the default robot model is close to the number of axes of the robot. In cases where there is additional tooling added

to the robot, additional model elements can be set up to protect this tooling from colliding with the obstacle object.

During setup, you must define the tool model that covers your tooling using a combination of up to five simple elements consisting of spheres and cylinders. Up to a total of ten models can be defined, one for each User tool frame. As part of the setup process, you need to associate each tool model with a tool frame number, so that when the tool frame number is changed, the corresponding tool model will be used by the system.

A total of ten static obstacles can be defined. Each obstacle consists of a combination of up to two simple elements.

The next step in the process is to define on which pairs of robot and tool, versus obstacles to check for a collision. After the combination pairs for checking are defined, you have the option to enable or disable the checking for each pair.



Caution

Be aware that too many models and combination settings will require more CPU time. If the computation time exceeds the system's capability, an alarm will be posted.

16.1.3 Multiple Controller Interference Checking

For applications where multiple robots are mounted closely together in a tight workcell, their workspace overlaps and the potential for collision among robots exists. By using the models (robot, tool, and obstacle models) you can set up interference zones for I/O interlocking.

For an application with more than two robots, you can define different pairs of robots that interfere. During setup, you can create an interference zone for each pair.

An interference zone is defined as an area that falls within the work envelope of a robot, in which there is the potential for the robot motion to coincide with the motion of another robot or machine, and for a collision to occur. Also, the idea behind zone interlocking is to allow one robot in the critical zone at one time. For example, if any part of Robot #2 is within the zone and Robot #1 attempts to intrude, Robot #1 will be held outside the zone until Robot #2 has released that zone.

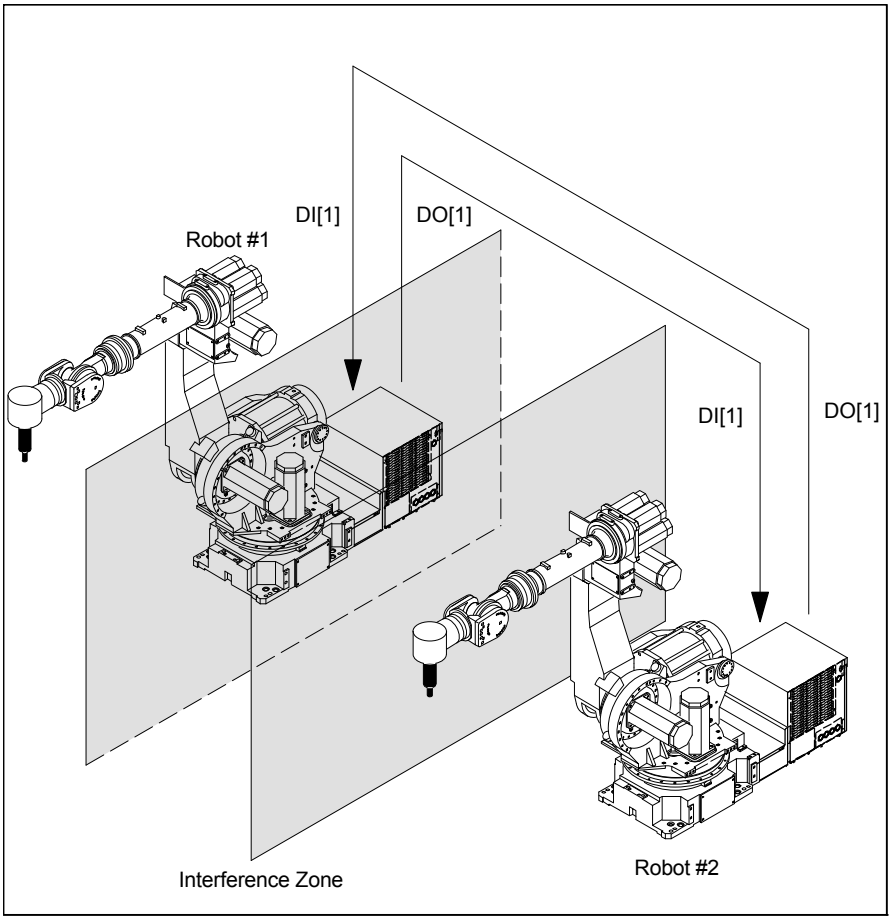
During setup, you must calibrate each pair of robots together. Robot #2 is calibrated with Robot #1 and Robot #1 is calibrated with Robot #2. As a result, Robot #2 has information about the position of the base of Robot #1, and Robot #1 has information about the position of the base of Robot #2.

The calibration data is used in the next step of setup, to automatically create a default interference zone for a given pair of robots. The default interference zone is calculated based on the working envelope of each robot in a pair, as well as the position of each robot relative to one another.

The size of the default interference zone depends directly on how close the two robots are mounted and the size of their respective working envelopes. If the robots are mounted very closely and the default interference zone is large, the cycle time will be increased by the robots repeatedly waiting for one another. You might need to alter the default interference zone to improve the performance of the system.

The last step of setup for multiple controller Interference Checking involves setting up the I/O communication for each pair. See Figure 16–2 for more information.

Figure 16–2. Robots on Different Controllers with Interference Zone and I/O



16.1.4 DualARM Single Controller Interference Checking

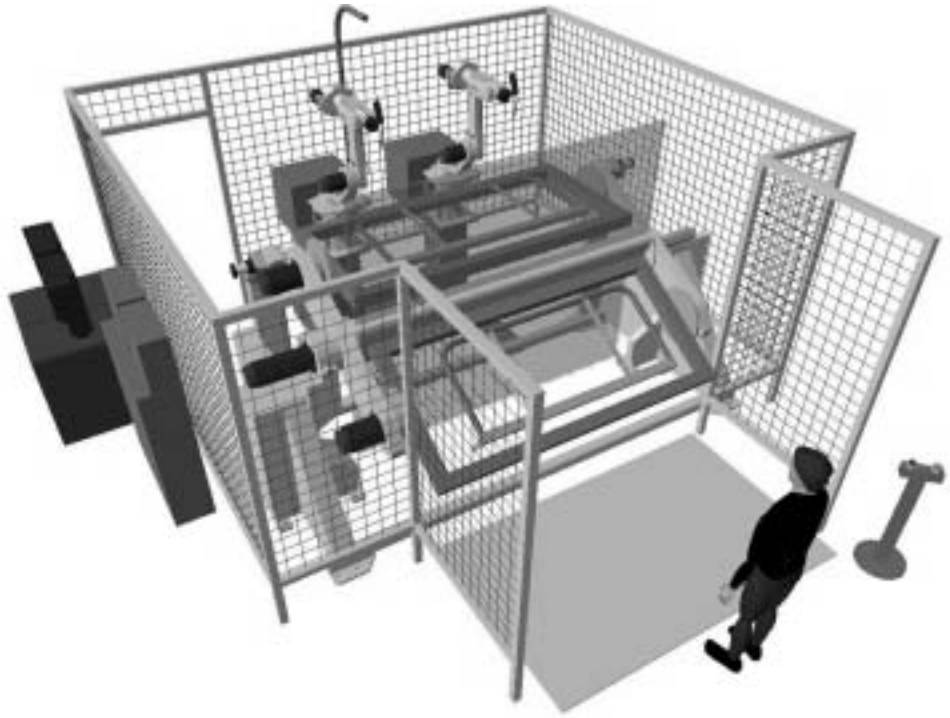
In a DualARM single controller system, Interference Checking can be used with the following features:

- Each robot arm can define its own set of tool models and obstacles.

- A fixture that is defined as a critical zone can be used to ensure that only one robot arm can enter the critical zone at any point of time.
- Each robot arm can be defined as a dynamic obstacle to the other robot arm, and vice versa.

The following figure illustrates a typical DualARM system with Ferris Wheel positioner:

Figure 16–3. Typical DualARM System with Ferris Wheel Positioner



Refer to Section 16.5 for more specific information about how to set up Interference Checking in a DualARM system.

16.2 MODELING

16.2.1 Overview

This section describes how to construct the following model types with simple elements:

- Robot model
- User tooling model
- Obstacle model

Robot models and User tooling consist of the following elements:

- Sphere
- Cylinder

The Obstacle model consists of a combination of the following three basic elements that you can set up:

- Sphere - is defined by its center point and its radius.
- Cylinder - is defined by its two end points and its radius.
- Plane - the finite plane element and the infinite plane element are both defined by three points: Pos 1, Pos 2, and Pos 3.
- Box - the rectangular box is defined by four points: Pos 1, Pos 2, Pos 3, and Pos 4.

See Figure 16–4 through Figure 16–6 for more information on each element Use Procedure 16-1 to set element data.

Figure 16–4. Sphere Element

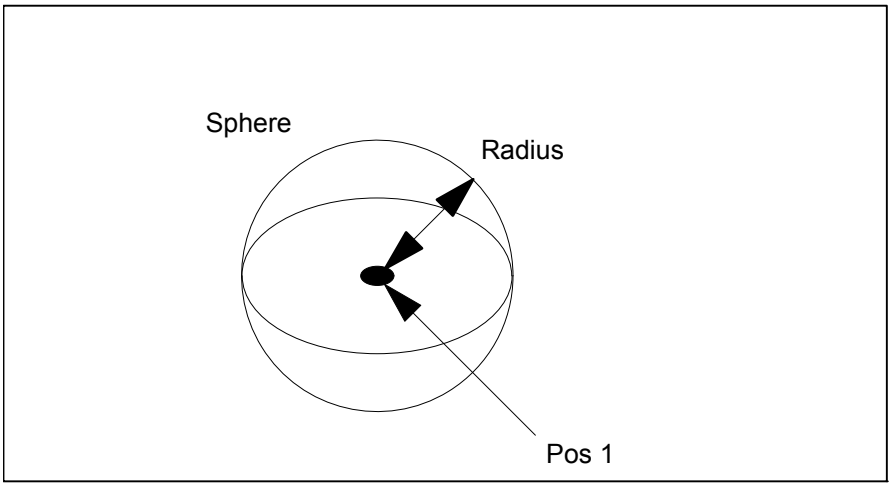


Figure 16–5. Cylinder Element

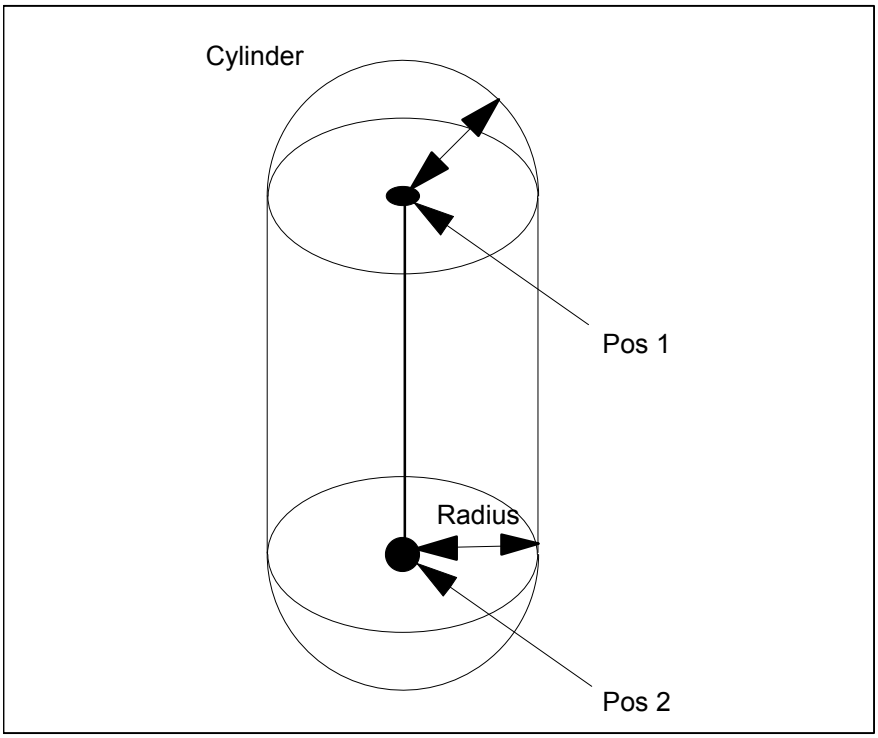
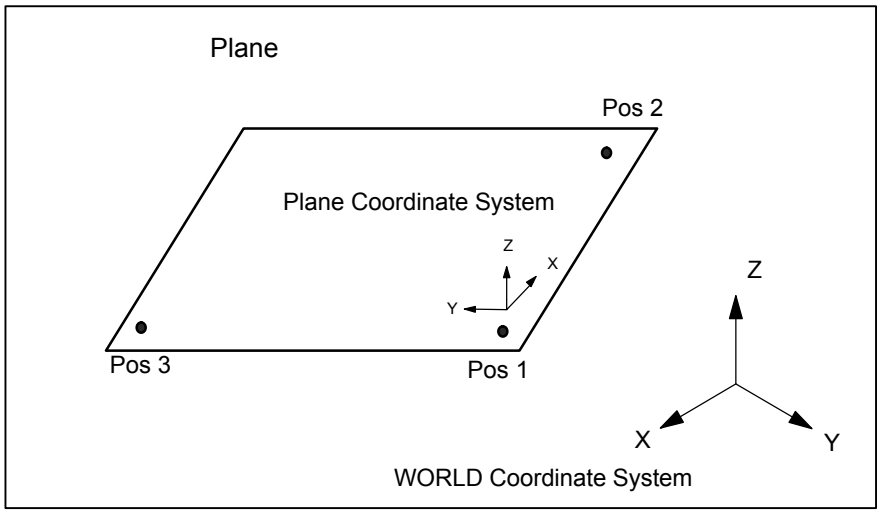


Figure 16–6. Plane Element



Definition of Elements

Models consist of elements that have the following parameters:

Table 16–1. Interference Check Element Data Screen Items

ITEM	DESCRIPTION
ENABLE/DISABLE status	This item defines the status of the robot model element.
Link no	This item defines the link number on the Interference Check Element Data screen.
Link [TYPE]	This item defines the link type.
Shape	This item defines the shape of the model element and can only be changed from the list screen.
Size (mm)	This item defines the size (mm) of the model element.
Pos 1 on specified link frame	For a sphere element, Pos 1 (x,y, z) defines the center of the sphere element. If the model element's shape is cylinder, then Pos1 (x,y,z) defines end point 1 of the cylinder element.
Pos 2 on specified link frame	This item is not used for a sphere model element. For a cylinder model element, Pos 2 (x,y, z) defines end point 2 of the cylinder element.

- **ENABLE/DISABLE -** If this parameter is ENABLED then this element composes the specified model. The default value is DISABLED (except for some pre-defined elements of the robot model).
- **Link No** : The link number of the robot.
- **Link type** : The link type is set by the system.
- **Shape** includes the following:
  - — Sphere
  - — Cylinder
  - — Plane
  - — Box
- **Size** includes the following:
  - — Sphere - The size represents the radius of the sphere.
  - — Cylinder - The size represents the radius of the cylinder
- **Pos 1/Pos 2/Pos 3/Pos 4 -** For the sphere element Pos 1 defines the center point of the sphere, and Pos 2 is not used. For the cylinder element, Pos 1 and Pos 2 are used to describe the two end points of the cylinder base. For the plane element, Pos 1, Pos 2, and Pos 3 define the plane coordinate system. The plane coordinate system is used to define which side of the plane is inside the plane obstacle and which side is outside of the plane obstacle. The positive z-direction of the plane coordinate system is the outside of the plane obstacle. For the box element, Pos 1, Pos 2, and Pos 3 define the plane coordinate system. Pos 4 is used to define the height of the box.

Note The order of Pos 1 and Pos 2 for the cylinder element is arbitrary. When the box element is defined as an obstacle or critical zone, the interference checking is done only between the last element of the Robot+Tool model and the box fixture.

Defining Element Data

In order to define the elements of each obstacle model it is not necessary to define the link number and link type. The position data of the elements are based on the WORLD frame of the specified group. The positions of Pos 1, Pos 2, or Pos 3 can be recorded by teaching the specified robot at the DETAIL screen. See Figure 16–7 through Figure 16–9 .

Figure 16–7. Teaching An Obstacle of a Sphere Type

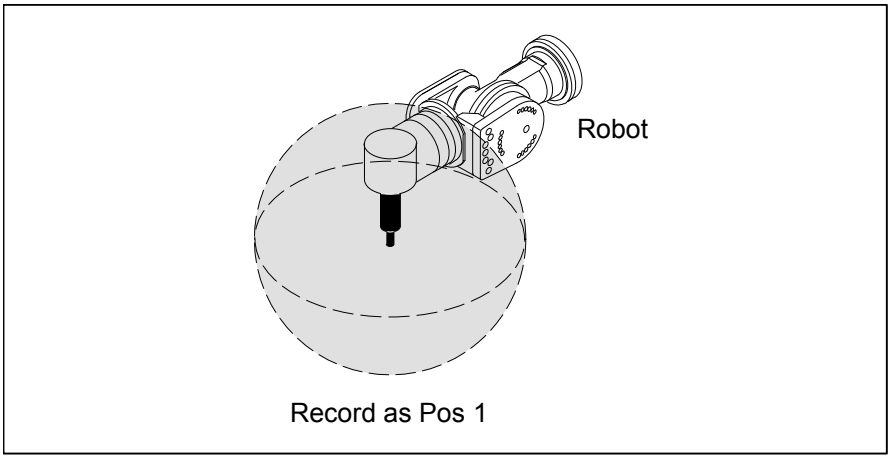


Figure 16–8. Teaching An Obstacle of a Cylinder Type

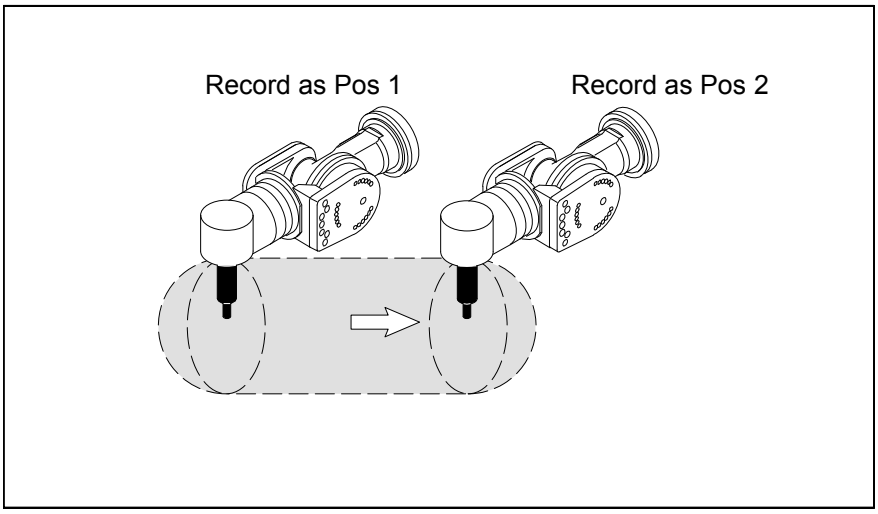


Figure 16–9. Teaching An Obstacle of a Plane Type

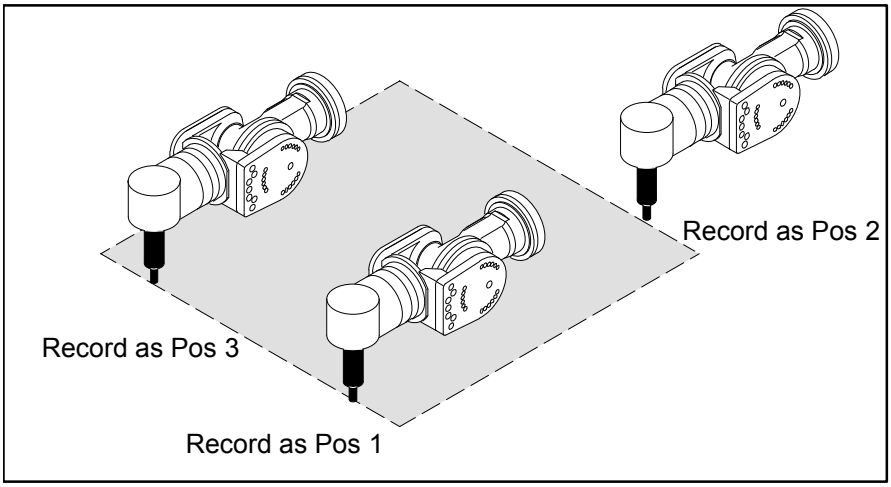
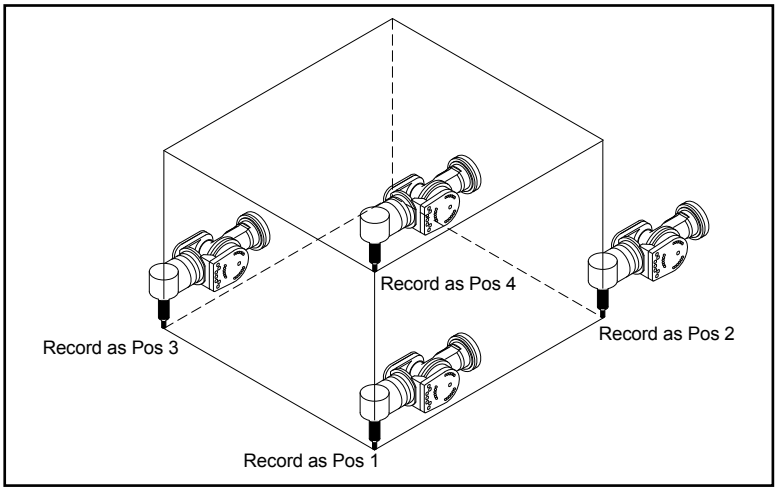


Figure 16–10. Teaching an Obstacle of a Box Type



Procedure 16-1 Setting Element Data

Steps

1. Move the cursor to Pos 1 on WORLD frame.
2. Move the robot (which is specified in the Taught Group), so that the TCP of the robot is pointing to the desired position.
3. Press the SHIFT key and F5, RECORD simultaneously. If the data is recorded successfully the message *Position was recorded* is displayed and the data on the screen is updated. See the following screen for an example.

```
Interference Check
Element data
Model:1 [Obstacle 1]
Element:1
Taught GP:1
  1  ENABLED/DISABLED          ENABLED
  2  Shape:                    Sphere
  3  Size(mm)                  230
Pos 1 on WORLD frame           <RECORD>
 6  X: 1250.000    Y: -550.000    Z: 0.000
Pos 2 on WORLD frame           <RECORD>
 7  X:  0.000      Y:  0.000      Z:  0.000
Position was recorded.
```

4. Confirm that the updated data on the screen is the position data from the specified group WORLD frame.
5. Repeat Step 1 through Step 4 for Pos 2 if the element shape is Cylinder. Repeat Step 1 through Step 4 for Pos 2 and Pos 3 if the element shape is Plane. Repeat Step 1 through Step 4 for Pos 2, Pos 3, and Pos 4 if the element shape is Box.

   Note After you have finished, all the data of the elements are initialized and the old data disappears.

16.2.2 Robot Arm Modeling

You can use this section to define your robot model. The default robot model elements are automatically defined during software installation. The content of default robot model elements are write-protected and the number of default robot elements varies with each robot model. To optimize the CPU usage, specific individual robot model elements can be disabled if the specified robot model element protection is not required. For example, the first robot model element for J1 protection can be disabled if the J1 protection is not required.

For example, for an R-2000i floor mount robot, 5 default robot model elements are set up automatically during loading. In this case, 5 elements are left for robot dressout modeling. Section 16.8 describes the robot parameters in detail for each robot that supports the Interference Checking function. Use Section 16.8 when you modify dressout for your application.

Use Procedure 16-2 to set up the robot model.

Table 16–2. Interference Check Element Data Screen Items

ITEM	DESCRIPTION
Robot setup	This item allows you to display the Robot List and Priority Table screen.
Model setup	This item allows you to display the Model Setup screen.
Check combination setup	This item allows you to display the Check Combination Setup screen.
Multiple controller setup	This item allows you to display the Multiple Controller Setup screen.

Table 16–3. Interference Check Model Setup Screen Items

ITEM	DESCRIPTION
Robot model setup	This item allows you to set up models for the robot and dressout.
Tool model setup	This item allows you to set up model(s) for the end of arm tooling.
Obstacle model setup	This item allows you to set up model(s) for obstacles.

Table 16–4. Interference Check Robot Model Setup Screen Items

ITEM	DESCRIPTION
Robot type	This item displays what robots are loaded on the system.

Procedure 16-2 Setting Up The Robot Model

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Interference. You will see a screen similar to the following.

```
Interference Check
Setup list of Interference Checking
 1 Model setup                 <*DETAIL*>
 2 Check combination setup     <*DETAIL*>
 3 Multiple controller setup   <*DETAIL*>
```

5. Move the cursor to Model Setup. You will see a screen similar to the following.

```
Interference Check
Setup list of models
1 Tool model setup              <*DETAIL*>
2 Obstacle model setup          <*DETAIL*>
3 Robot model setup             <*DETAIL*>
```

6. Move the cursor to Robot Model Setup, and press ENTER to display the robot list. You will see a screen similar to the following.

```
Interference Check
Robot model list
      Robot
  1   R2K/165F
```

Note If you press F2, DETAIL, the current robot DETAIL screen will be displayed. The robot list screen cannot be edited.

```
Interference Check
Model data
Robot:1 [R2K/165F]
                  Shape       Size
1     DISABLED    Cylinder    424
2     DISABLED    Cylinder    170
3     DISABLED    Cylinder    180
4     ENABLED     Cylinder    190
5     ENABLED     Cylinder    118
6     ENABLED     Sphere      508
7     ENABLED     Cylinder    161
8     DISABLED    Sphere      0
9     DISABLED    Sphere      0
10    DISABLED    Sphere      0
```

7. In the screen listed above, there are two kinds of model elements:

- Elements 1 through 7 are already defined at shipment and the content of elements cannot be modified. To optimize the CPU usage, specific individual robot model element can be disabled if the specified robot model element protection is not required. For example, the first robot model element for J1 protection can be disabled if the J1 protection is not required. Moving the cursor to the first robot model element and pressing F5 will disable the J1 protection..
- Elements 8 through 10 can be modified.

The cursor is positioned on the side of the element item number. The cursor can be positioned on any of the element parameters. Press F2, DETAIL, to refer to the detailed data of each element.

8. To add or edit one of the user-definable robot model elements, move the cursor to the desired robot model element. You can change the ENABLED/DISABLED field, the Shape field, and the Size field from the list screen, or from the DETAIL screen.

The ENABLED/DISABLED parameter, Shape parameter, and Size parameters can all be changed from the list screen or from the element DETAIL screen. Move the cursor to the desired robot model element and press F2 to display the DETAIL screen.

a. **ENABLE or DISABLE the Parameter** - To select this parameter move the cursor on ENABLE/DISABLE of the desired element. Press F3, ENABLED, or F4, DISABLED, to change the setting.

b. **Set the Parameter Shape** - Move the cursor to the Shape of the desired element and press F4, [CHOICE], to display the select screen. Select Sphere or Cylinder.

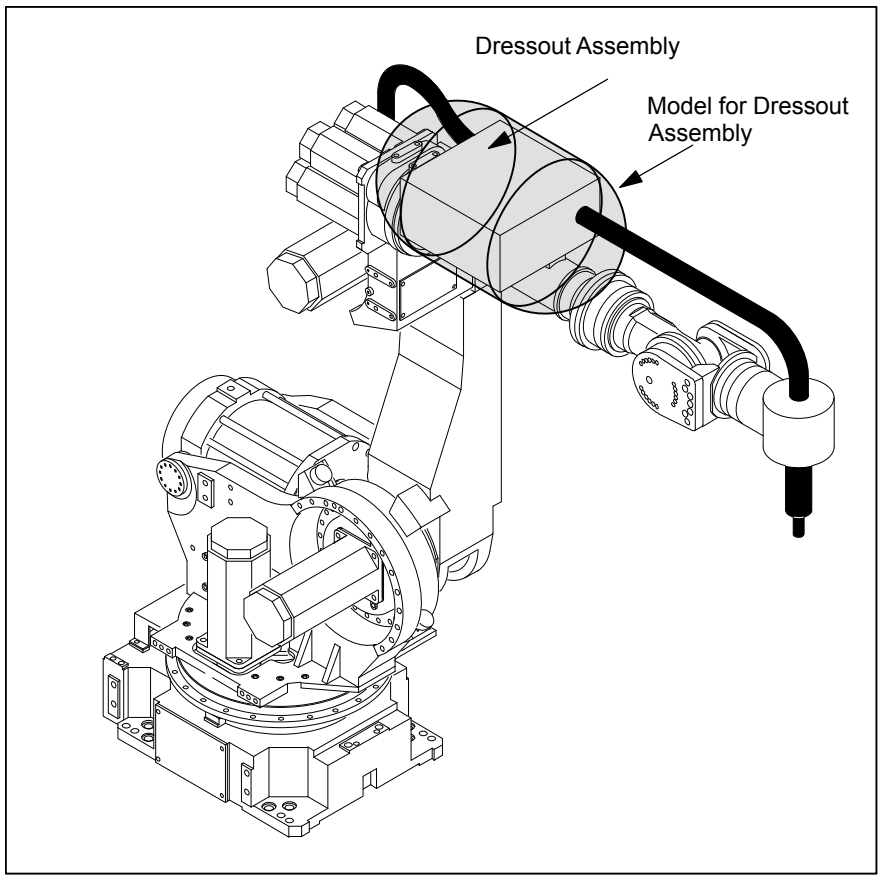
```
1 Sphere
2 Cylinder
Interference Check
Model data
  6   ENABLED     Sphere       508
  7   ENABLED     Cylinder     161
  8   DISABLED    Sphere       0
  9   DISABLED    Sphere       0
 10   DISABLED    Sphere       0
```

c. **Set the Size of the Parameter** - Move the cursor to the size and input the value directly.

```
Interference Check
Model data
Robot:1 [R2K/165F]
                  Shape       Size
1     DISABLED    Cylinder    424
2     DISABLED    Cylinder    170
3     DISABLED    Cylinder    180
4     ENABLED     Cylinder    190
5     ENABLED     Cylinder    118
6     ENABLED     Sphere      508
7     ENABLED     Cylinder    161
8     DISABLED    Sphere      0
9     DISABLED    Sphere      0
10    DISABLED    Sphere      0
```

Note The element data screen describes how to display the detailed data of each element. The data of the pre-defined elements can be viewed but not modified while the other elements can be modified. See Figure 16–11 for an example of a user-defined robot-model element.

Figure 16–11. Element Data



Note The dressout assembly can be modeled by a cylinder element attached to link No. 3.

```
Interference Check
Element data
Robot:1 [R2K/165F]
Element:8
  1 ENABLE/DISABLE:            ENABLED
  2 Link No:                   3
  3 Link type:                 Normal
  4 Shape:                     Cylinder
  5 Size(mm):                  300
  Pos 1 on specified link frame
   6  X:+0.000    Y:0.000      Z:-400.000
  Pos 2 on specified link frame
   7  X:+0.000    Y:0.000      Z:-850.000
```

9. To enable or disable element data move the cursor to ENABLED or DISABLED and press F3, ENABLED, or F4, DISABLED.

10. To define the link number and link type you want, move the cursor to Link No. and Link type and press ENTER.

11. Move the cursor to the data of the desired element and input the value directly.

Note When the selected shape is Sphere, type the position data of the center point of the sphere to Pos 1. When the selected shape is Cylinder, input the position data of two center points of the base of the cylinder to Pos 1 and Pos 2. The order of Pos 1 and Pos 2 is arbitrary. The positions are one from the specified link number and frame. Determine the position data by referring to the plan.

Caution

After Procedure 16-2 is finished, all the data will be initialized and the old data will disappear.

12. It is possible to reset the data of the element which is modified at once. At the detailed element screen press F3, CLEAR, and the following message will be displayed: *Clear this element? [Y=1/N=else]* To reset the data press 1; otherwise, press any other key.

Note For detailed information on robot dimensions and interference check parameters.

16.2.3 User Tool Modeling

The method used to create the end-of-arm tooling model is described in this section. One example of the method is shown in Figure 16–12 and Table 16–5 . From a 2-D side view of the wrist of the robot and the tooling, the model elements are used to protect the entire tooling, and in this case a

welding torch. Three elements are used to model the tooling in this example. See Figure 16–12 and Table 16–5 for more details.

Definition of Tool Models

The tool model is a combination of the following elements:

- **Comment** - The name or comment of each tool that is recorded. Although, you are able to leave the field blank, it is recommended to register a comment to classify each tool.
- **Definition of Elements** - You must organize the tool models by defining each element. However, if the total number of elements (robot models, tool models and obstacle models) are combined, an alarm might occur during run-time.
- **Correspondence of User tool frame number to tool number** - You must set the table to match to the tool model number and the user tool frame number for each group. The Interference Checking function will judge the tool which the robot has from the current user tool frame number. When the tool number is set to 0, the Interference Checking function works under the assumption that the robot has no tool.

Figure 16–12. Tool Model Example

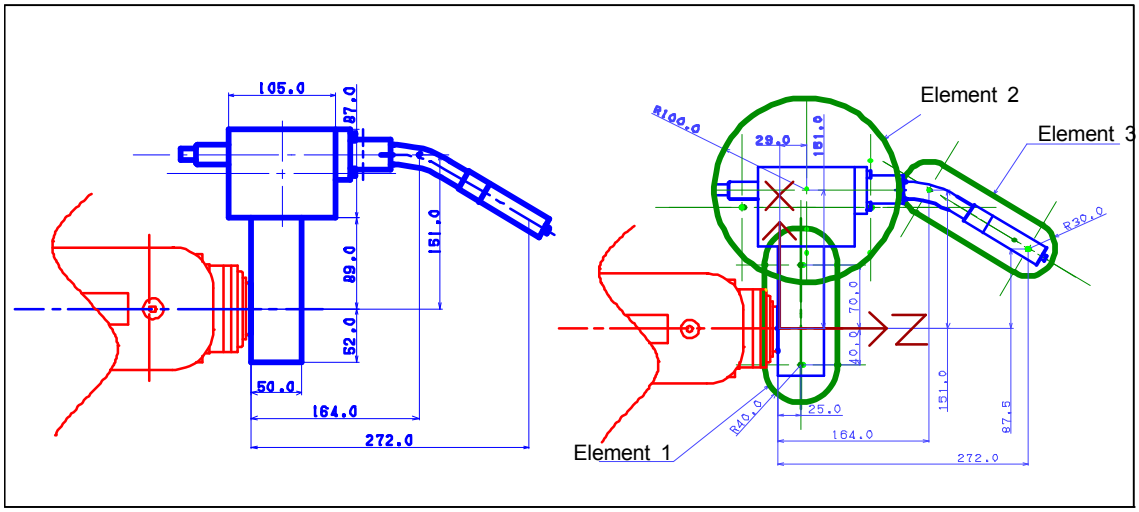


Table 16–5 lists and describes the element parameters of the tool model example shown in Figure 16–12 .

Table 16–5. Element Parameters

ITEM/ELEMENT	ELEMENT 1	ELEMENT 2	ELEMENT 3
ENABLED/DISABLED	ENABLED	ENABLED	ENABLED
Shape	Cylinder	Sphere	Cylinder
Size	40.0	100.0	30.0
Pos 1 X	-40.0	151.0	151.0
Pos 1 Y	0.0	0.0	0
Pos 1 Z	25.0	29.0	164.0
Pos 2 X	70.0	-	87.5
Pos 2 Y	0.0	-	0.0
Pos 2 Z	25.0	-	272.0

Defining the Tool Model

Use Procedure 16-3 to define the tool model using Figure 16–12 and Table 16–5 as an example.

Table 16–6. Interference Check Element Data Screen Items

ITEM	DESCRIPTION
Tool	This item indicates which tool model is being edited.
Element	This item indicates which element of the tool model is being edited.
ENABLE/DISABLE status	This item defines the status of the element data.
Shape	This item defines the shape of the model element and can only be changed from the list screen.
Size (mm)	This item defines the size (mm) of the model element.
Pos 1 on specified link frame	For a sphere element, Pos 1 (x,y, z) defines the center of the sphere element. If the model element's shape is cylinder, then Pos1 (x,y,z) defines end point 1 of the cylinder element.
Pos 2 on specified link frame	This item is not used for a sphere model element. For a cylinder model element, Pos 2 (x,y, z) defines end point 2 of the cylinder element.

Table 16–7. Interference Check Tool Model Setup Screen Items

ITEM	DESCRIPTION
Tool model	This item allows you to set up a list of tool models.

Table 16–8. Interference Check Tool Model List Screen Items

ITEM	DESCRIPTION
Comment	This item allows you to name each tool model.

Table 16–9. Interference Check Tool Model Element List Screen Items

ITEM	DESCRIPTION
ENABLED/DISABLED status	This item allows you to enable or disable the tool model element.
Shape	This item defines the shape of the tool model element.
Size	This item defines the size (mm) of the tool model element.

Procedure 16-3 Defining the Tool Model

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Interference. You will see a screen similar to the following.

```
Interference Check
Setup list of Interference Checking
1 Model setup                    <*DETAIL*>
2 Check combination setup        <*DETAIL*>
3 Multiple controller setup      <*DETAIL*>
```

5. Move the cursor to Model setup, and press ENTER. You will see a screen similar to the following.

```
Interference Check
Setup list of Models
1 Tool models setup              <*DETAIL*>
2 Obstacle models setup          <*DETAIL*>
3 Robot models setup             <*DETAIL*>
```

6. Move the cursor to Tool Models setup, and press ENTER. You will see a screen similar to the following.

```
Interference Check
Tool model list
   Comment
 1  Tool1
 2
 3
 4
 5
 6
 7
 8
 9
 10
```

7. Press ENTER to display the following screen will be displayed. It is possible to edit the comment by pressing F1 and F5. The maximum length of the comment is 13 characters.

```
1 Upper Case
2 Lower Case
3 Punctuation
4 Options
Interference Check
Comment
1 Tool1
2
3
```

8. Display the tool model list. You will see a screen similar to the following.

```
Interference Check
Tool model list
   Comment
 1 Tool1
 2
 3
 4
 5
 6
 7
 8
 9
 10
```

9. Move the cursor to the desired tool and press F2, DETAIL. You will see a screen a similar to the following.

```
Interference Check
Model data
Tool: 1  [Tool No.1]
                   Shape       Size
1     ENABLED      Cylinder     40
2     ENABLED      Sphere       100
3     ENABLED      Cylinder     30
4     DISABLED     Sphere       0
5     DISABLED     Sphere       0
```

10. At this screen, the following parameters of each element are to be modified. Press F2, DETAIL, to edit the data of each element.

11. Move the cursor at ENABLED or DISABLED of the desired element. Press F3, ENABLED, or F4, DISABLED, to change the setting.

12. Move the cursor on SHAPE of the desired element. Press F4, [CHOICE], to select Sphere or Cylinder for each element.

```
1 Sphere
2 Cylinder
Interference Check
Model data
  1   ENABLED      Cylinder    40
  2   ENABLED      Sphere      100
  3   ENABLED      Cylinder    30
  4   DISABLED     Sphere      0
  5   DISABLED     Sphere      0
```

13. Move the cursor to the size of each element and type the data directly.

```
Interference Check
Model data
Tool: 1  [Tool No. 1]
Element:1
                   Shape      Size
  1   ENABLED      Cylinder    40
  2   ENABLED      Sphere      100
  3   ENABLED      Cylinder    30
  4   DISABLED     Sphere      0
  5   DISABLED     Sphere      0
Old Value: 40
```

14. Press F2, DETAIL, in the model data screen and the detailed data of the element will be displayed.

```
Interference Check
Model data
Element:1
1  ENABLED/DISABLED          ENABLED
2  Shape:                    Cylinder
3  Size                      40
Pos 1 on Flange Flame
 6  X:-40.000  Y:  0.000  Z:  25.000
Pos 2 on Flange Flame
 7  X:  70.000  Y:  0.000  Z:  25.000
```

15. Press F4, ENABLED, or F3, DISABLED, to make modifications.

16. To change the size of the element, move the cursor to Size and type the value of the size directly.

17. To change the positional data, move the cursor to the positional data and type the value directly. When the selected shape is a sphere, type the positional data of the center point of the sphere to Pos 1. When the selected shape is Cylinder, type the positional data of 2 center points of the base of the Cylinder to Pos 1 and Pos 2. Determine the positional data by referring to your mechanical drawing of the end of arm tooling. See Figure 16–12 for a mechanical drawing of an example tool. See the following screen for an example.

```
Interference Check
Element data
Tool: 1  [Tool No. 1]
1  ENABLED/DISABLED          ENABLED
2  Shape:                    Cylinder
3  Size(mm):                 40
Pos 1 on Flange Flame
 6  X:-40.000  Y:  0.000  Z:  25.000
Pos 2 on Flange Flame
 7  X:  70.000  Y:  0.000  Z:  25.000
Old Value:-40
```

18. Press F3, CLEAR, at the DETAIL screen to reset the data of one of the elements. The following message will be displayed: *Clear this element? [Y=1/N=else].* To initialize the element's data, type 1; otherwise, press any other key.

Note It is necessary to match each user tool frame you define with the corresponding tool model definition. If and when the tool changes, the user tool frame is updated, and the tool model that

the system uses is updated as well. Use Step 19 through Step 21 to match UTOOL(s) and Tool model(s). If the following steps are not completed, no tool model will be used.

19. Display the tool model list screen. You will see a screen similar to the following.

```
Interference Check
Tool model list
   Comment
 1    Tool No. 1
 2
 3
 4
 5
 6
 7
 8
 9
 10
```

20. Press F4, UT_DATA, and you will see a screen similar to the following.

```
Interference Check
UTOOL - Tool Model definition
Group:1
      Utool No.      Tool No
1     1     -        0
2     2     -        0
3     3     -        0
4     4     -        1
5     5     -        0
6     6     -        0
7     7     -        0
8     8     -        0
9     9     -        0
10    10    -        0
```

21. Confirm the group number and type the tool number to the corresponding user tool frame number (Utool No. on the screen). For example, in the screen listed above, user tool frame No. 4 corresponds to tool number 1. This means that if the current user tool frame is No 4 then the group 1 robot has the tool which is defined as No 1 in the tool model list screen. In this case, the other user tool frame has been selected and it assumes that group 1 robot does not have a tool. To assign the tool model to a different robot group if it is used in a DualARM system, press F2, GROUP.

Note It is important to clear out all of the data of one tool model including all elements of the model and the comment if it is no longer used.

22. Display the tool model list. You will see a screen similar to the following.

```
Interference Check
Tool model list
   Comment
 1    Tool No. 1
 2
 3
 4
 5
 6
 7
 8
 9
 10
```

23. If needed, you can delete a tool model by moving the cursor to the tool model you want to delete and by pressing F3, CLEAR. The following confirmation message will be displayed: *Clear this tool data ? (Y=1/N=else).*

24. To reset the data, type 1; otherwise, press any other key.

16.2.4 Obstacle Modeling

Each Obstacle Model can be defined by a maximum of two elements. The data defined includes the following:

- **Comment** - Includes the name or comment of each obstacle recorded. You need to register the comment so each obstacle is classified.
- **Taught group** - Includes specifying the group for which the position of each obstacle is defined.
- **Definition of elements** - Construct the models by defining the elements. For one obstacle, you can define a total of two elements. However, due to the number of defined elements, an alarm might occur while running. You must consider the total number of elements used in all of the models to determine what is needed.

Note You can initialize all of the data of one obstacle model or element at once.

Obstacle Model Example

In Figure 16–13 , the coordinate system used to determine the position of the fixture is the robot's WORLD frame. Refer to Section for more information about frame setup. The element parameters used to construct the obstacle model in Figure 16–13 are listed on Table 16–10 .

Figure 16–13. Obstacle Model Example

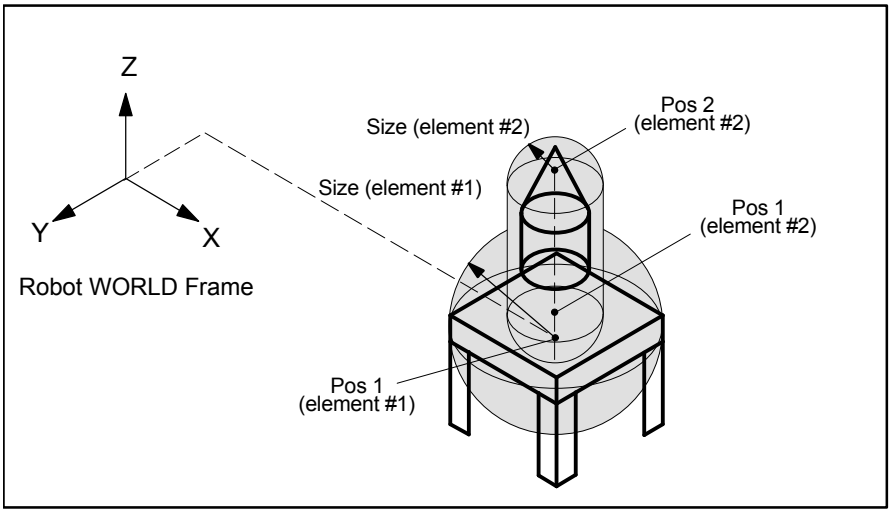


Table 16–10. Obstacle Model Parameters

ITEM/ELEMENT	ELEMENT 1	ELEMENT 2
ENABLED/DISABLED	ENABLED	ENABLED
Shape	Sphere	Cylinder
Size	650	230
Pos 1 X	1250	1250
Pos 1 Y	-550	-550
Pos 1 Z	0	200
Pos 2 X	-	1250
Pos 2 Y	-	-550
Pos 2 Z	-	1000

Note The position of the obstacle model in the robot's WORLD frame is known. You must obtain the position data by jogging the robot to the correct position and recording that position.

Use Procedure 16-4 to set up the obstacle model.

Table 16–11. Interference Check Obstacle Model Setup Screen Items

ITEM	DESCRIPTION
Obstacle model setup	This item allows you to set up a list of obstacle models.

Table 16–12. Interference Check Obstacle Model List Screen Items

ITEM	DESCRIPTION
Comment	This item allows you to name each obstacle model.
Taught GP	This item defines the group to which the obstacle model corresponds.

Table 16–13. Interference Check Obstacle Model Data Screen Items

ITEM	DESCRIPTION
Obstacle no	This item defines the obstacle number.
Taught GP no	This item defines the group number for the obstacle.
ENABLED/DISABLED status	This item allows you to enable or disable the obstacle model element.
Shape	This item defines the shape of the obstacle model element.
Size	This item defines the size (mm) of the obstacle model element.

Table 16–14. Interference Check Obstacle Element Data Screen Items

ITEM	DESCRIPTION
Obstacle no	This item defines the obstacle number.
Taught GP no	This item defines the group number for the obstacle.
ENABLED/DISABLED status	This item allows you to enable or disable the obstacle model element.
Shape	This item defines the shape of the obstacle model element.
Size	This item defines the size (mm) of the obstacle model element.
Pos 1 in World frame	For sphere elements, Pos 1 (x,y,z) defines the center of the sphere element. For a cylinder element, Pos 1 (x,y,z) defines end point 1 of the cylinder element. For a plane element, Pos 1 (x,y,z) defines the origin of the plane element.
Pos 2 in World frame	This item is not used for sphere model elements. For a cylinder model element, Pos 2 (x,y,z) defines end point 2 of the cylinder element. For a plane model element, Pos 2 (x,y,z) defines a point on plane element.
Pos 3 in World frame	This item is not used for sphere or cylinder model elements. For a plane model element, Pos 3 (x,y,z) defines a point on the plane element.

Procedure 16-4 Setting Up The Obstacle Model

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Interference. You will see a screen similar to the following.

```
Interference Check
Setup list of Interference Checking
1 Model setup                  <*DETAIL*>
2 Check combination setup      <*DETAIL*>
3 Multiple controller setup    <*DETAIL*>
```

5. Move the cursor on Model Setup and press ENTER to display the setup list of models. You will see a screen similar to the following.

```
Interference Check
Setup list of Models
1 Tool models setup            <*DETAIL*>
2 Obstacle models setup        <*DETAIL*>
3 Robot models setup           <*DETAIL*>
```

6. Move the cursor on Obstacle Model Setup, and press ENTER to display the static obstacle list screen. You will see a screen similar to the following.

```
Interference Check
Obstacle list Model
   Comment                     Taught Group
 1                             1
 2                             1
 3                             1
 4                             1
 5                             1
 6                             1
 7                             1
 8                             1
 9                             1
 10                            1
```

7. To edit the comment, press ENTER on the desired obstacle and the edit screen will be displayed. The maximum number of characters is 13.

```
Interference Check
1 Lower case
2
3 Punctuation
Interference Check
    Comment
1 Obstacle 1
2
3
```

Note You can display and edit the status of all composed elements of one obstacle model. Also, you can define the elements for one obstacle model on the screen.

8. Move the cursor to the desired obstacle and press F2, DETAIL, to display the obstacle model list. You will see a screen similar to the following.

```
Interference Check
Obstacle list Model
   Comment                    Taught GP
 1 Obstacle 1                 1
 2                            1
 3                            1
 4                            1
 5                            1
 6                            1
 7                            1
 8                            1
 9                            1
 10                           1
```

9. On the screen listed below, the parameters for each element must be modified. Press F2, DETAIL, and you also can edit the data in the detail screen of each element.

```
Interference Check
Model data
Model 1 [Obstacle 1]
Taught GP: 1
                   Shape       Size
      1 ENABLED    Sphere      650
      2 ENABLED    Cylinder    230
```

10. Move the cursor to the ENABLED or DISABLED field and press F3, ENABLED, or F4, DISABLED, to change the setting.

11. Move the cursor to the Shape field of the element and press F4, [CHOICE], to display the [CHOICE] screen. Select Sphere or Cylinder and you will see a screen similar to the following.

```
1 Sphere
2 Cylinder
Interference Check
Model data
  1   ENABLED      Sphere      650
  2   ENABLED      Cylinder    230
```

12. Move the cursor to the Size field and type the value directly if you need to change it. You will see a screen similar to the following.

```
Interference Check
Model data
Model 1 [Obstacle 1]
Taught GP: 1
                Shape       Size
  1   ENABLED   Sphere      650
  2   ENABLED   Cylinder    230
Old Value: 650
```

Note Use Step 13a through Step 13c to re-initialize all the data from one of the obstacle models including the elements of the model and their comments.

13. Re-initialization obstacle model data:

a. Display the obstacle model data screen. You will see a screen similar to the following.

```
Interference Check
Obstacle model list
   Comment         Taught GP
 1 Obstacle 1      1
 2                 1
 3                 1
 4                 1
 5                 1
 6                 1
 7                 1
 8                 1
 9                 1
 10                1
```

b. Move the cursor to the desired obstacle, and press F3, CLEAR. You will see the following message: *"Clear this tool data? [Y=1/N=else]:"*

c. When you are initializing the data, type 1; otherwise, press any other key.

16.2.5 How to Teach a Plane Obstacle Effectively

In order to create a plane obstacle, you need to teach three points to define the plane. The method you use to teach the three points (Pos 1, Pos 2 and Pos 3) determines which side of the plane is outside the plane obstacle.

If you think of the plane obstacle as a wall (virtual wall) separating the inside and the outside of a room, one side of the plane is outside the plane obstacle, and the other side is inside the plane obstacle. The outside of a plane obstacle is a zone where access is unrestricted, and the inside of the plane obstacle is a zone where access is restricted.

By convention, the outside of a plane obstacle is determined by the direction of the normal to the plane, which is determined by the three points: Pos 1, Pos 2 and Pos 3. You can teach the three points that define the desired plane by following the right hand rule.

Refer to Table 16–15 to set element data.

Table 16–15. Setting Element Data

ELEMENT	DEFINITION
ENABLED/DISABLED	Move the cursor to ENABLED or DISABLED and press F4, ENABLED, or F5, DISABLED, to modify.
Link Number and Link Type	It is unnecessary to define the parameters for an obstacle. Leave the parameters set to their default, the link number set to 0, and the link type set to NORMAL.
Shape	Move the cursor to Shape and press F4, [CHOICE]. The selection screens of Sphere and Cylinder are displayed and you can select the shape.

Table 16–15. Setting Element Data (Cont'd)

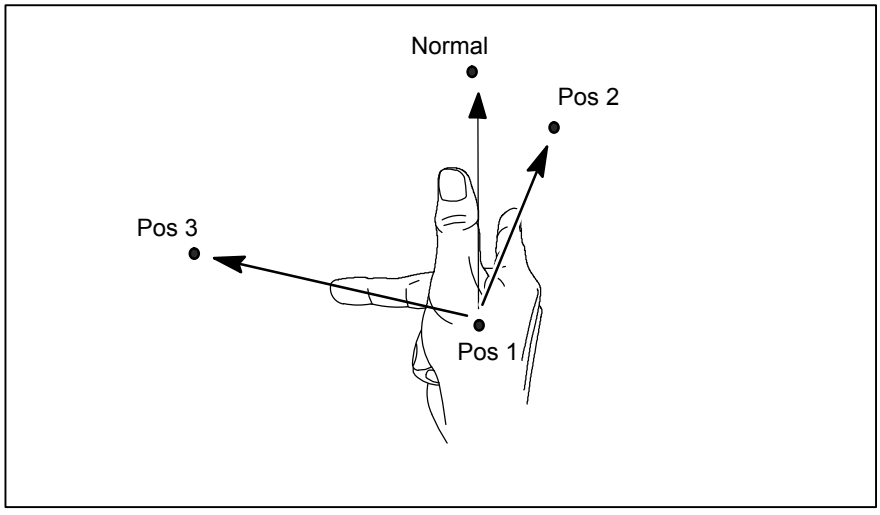
ELEMENT	DEFINITION
Size	Move the cursor to the data of the aimed element and edit the value directly.
Position	When the selected shape is Sphere, set the positional data of the center point of the Sphere to Pos 1. Do not select Pos 2 because it has invalid data. When the selected shape is Cylinder, set the Position data of 2 center points of the base of the cylinder to Pos 1 and Pos 2. The order of Pos 1 and Pos 2 is arbitrary. The positions mentioned are from one specified robot in WORLD frame. Determine the positional data by referring to the plan or teaching the robot.

Right Hand Rule

To determine the direction of the normal to the plane defined by Pos 1, Pos 2, and Pos 3 do the following. See Figure 16–14 .

- Point your index finger along the direction from Pos 1 to Pos 2.
- Point your middle finger along the direction from Pos 1 to Pos3.
- Hold your thumb perpendicular to the plane formed by both your index finger and middle finger.
- Your thumb will then point in the direction of the normal to the plane defined by Pos 1, Pos 2 and Pos 3.

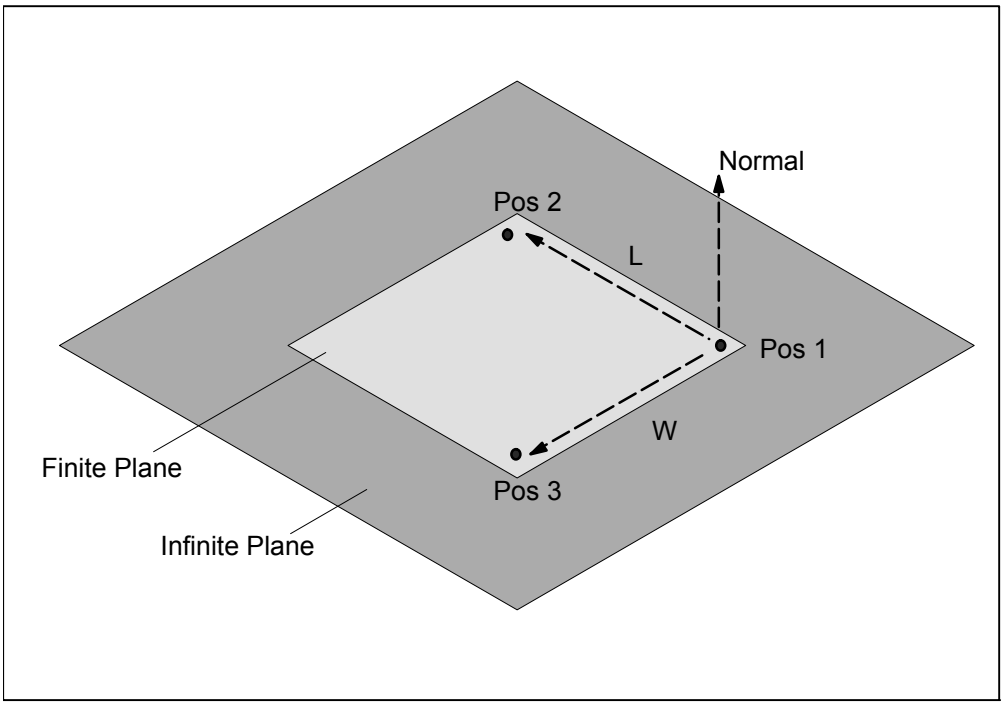
Figure 16–14. Right Hand Rule Example



Infinite Plane and Finite Plane

By definition, the infinite plane and finite plane differ by the fact that the finite plane has dimensions of the finite plane (width W and length L)are determined from the location of the three taught points: Pos 1, Pos 2 and Pos 3. The plane obstacles shown in Figure 16–15 are the respective infinite and finite planes derived from the same points: Pos 1, Pos 2 and Pos 3.

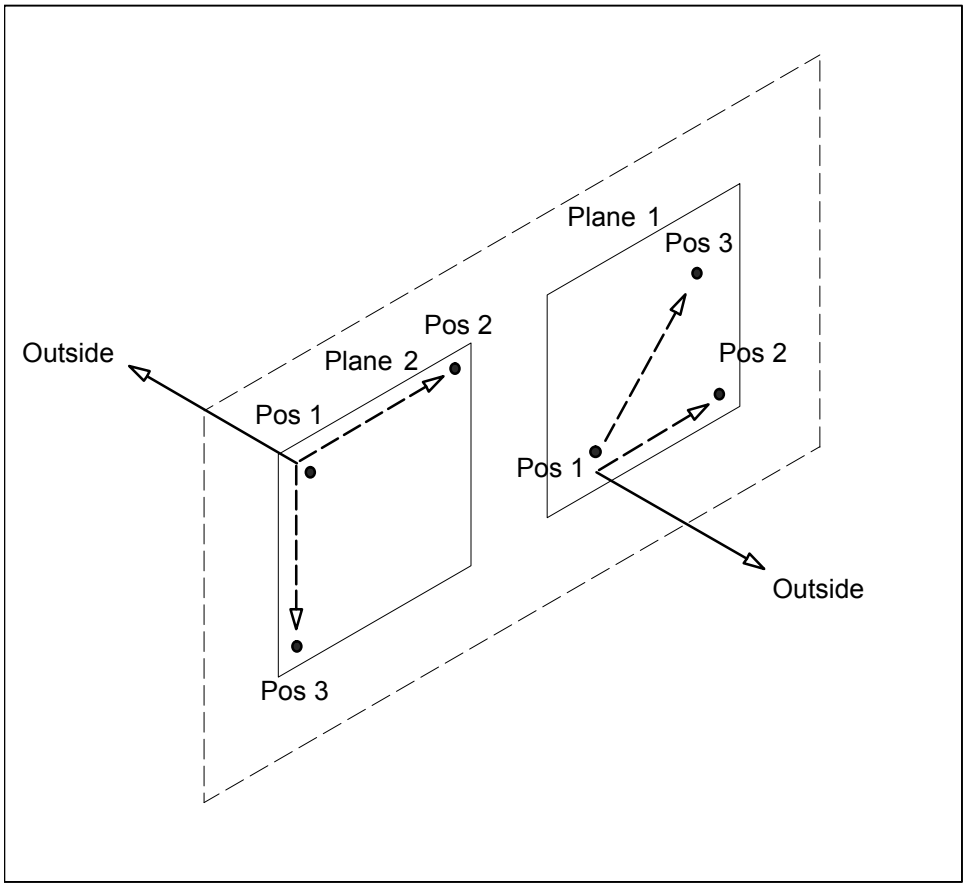
Figure 16–15. Infinite Plane and Finite Plane Examples



Plane Obstacle Example

Figure 16–16 shows two similar plane obstacles with different sides as the outside of the obstacle (opposite orientation). If plane 1 and plane 2 in this example are infinite planes (with infinite dimensions) they are the same plane, with opposite orientation.

Figure 16–16. Plane Obstacle Example



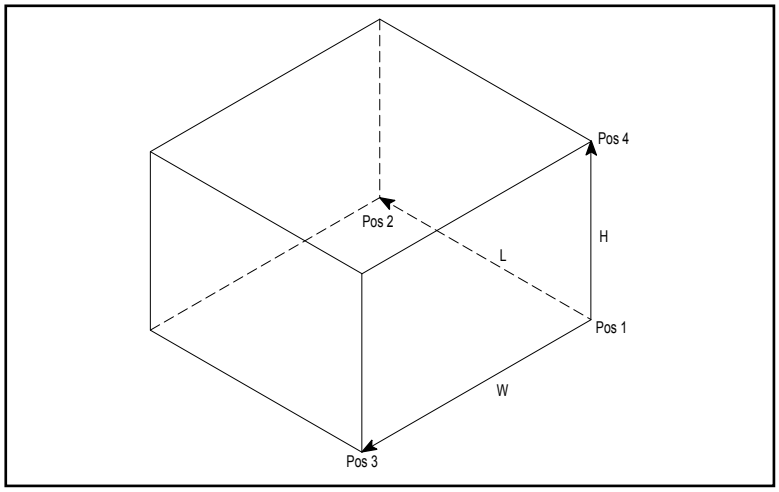
Plane 1 and plane 2 have opposite orientation because of the way Pos 1, Pos 2 and Pos 3 are arranged in the two cases. By the right hand rule, you can verify that the arrangements of the three taught points: Pos 1, Pos 2 and Pos 3 for plane 1 and plane 2 define two similar planes with opposite orientation.

16.2.6 How to Teach a Box Obstacle Effectively

In order to teach a box obstacle, you need to teach three points (Pos 1, Pos 2, and Pos 3) to define a rectangular plane, and then teach Pos 4 to define the height of the box. If Pos 4 is taught inside of the plane, an error message “Four points not form a box” will be posted on the teach pendant.

Length (L) and width (W) are calculated by the rectangular plane formed by Pos 1, Pos 2, and Pos 3. Height (H) is defined by Pos 4 and the plane. The following figure illustrates the L, W, and H:

Figure 16–17. Box Obstacle Example



Caution

Note that when a box is defined for a fixture, the interference checking is done only between the last element of the Robot+Tool model and the box, and it only protects the TCP of the robot. The system configuration must meet either one of the following two requirements to protect the robot TCP:

- The TCP is included in the last element of the tool model.
- A tiny sphere with the TCP as the center is defined as the last element of the tool model.

If neither condition is met, the robot TCP could be damaged.

16.3 SETTING UP COLLISION CHECKING BETWEEN ROBOT ARM AND STATIC OBSTACLES

16.3.1 Overview

After the robot models (robot arm and dressout), the tooling models, and the static obstacle models are defined, you must set up checking combination pairs.

16.3.2 Setting Up Combination Pairs

There are ten checking combination pairs available. This last step of setup lets you define the robot model and static obstacle pair you want to check. You can also enable or disable a checking pair at any time. The robot model number refers to the robot group number and the obstacle number refers to the obstacle number in the obstacle set up list. Use Procedure 16-5 to specify the combination pairs to be checked.

Table 16–16. Interference Check Combination Setup Screen Items

ITEM	DESCRIPTION
Check combination setup	This item allows you to specify the combination pairs to be checked.

Table 16–17. Interference Combination to Check Screen Items

ITEM	DESCRIPTION
Parameters: Rbt +Tool and Obstacle	This item allows you to define Rbt +Tool or Obstacle parameters.
ENABLED/DISABLED status	This item allows you to define enabled or disabled status for the check combination screen.

Table 16–18. Interference Combination to Check DETAIL Screen Items

ITEM	DESCRIPTION
Combination no.	This item is the specific combination that you selected from the Combination to Check screen.
Input	This item is the input I/O point data used for interference zone checking and is used only if stop on boundary is set to 0.
Output	This item is the output I/O point data used for the interference zone checking and is used only if stop on boundary is set to 0.
Stop (on boundary)	This item defines if combination pair is used for obstacle checking (1) or interference checking (0).
Other Robot No.	This item defines the robot number of the other robot used for interference zone checking. Enter 0 for multiple robots on the same controller.

Procedure 16-5 Specifying Combination Pairs

Steps

1. Press MENUS.
2. Select SETUP.

3. Press F1, [TYPE].
4. Select Interference. You will see a screen similar to the following.

```
Interference Check
Setup list of Interference Checking
1 Model setup                    <*DETAIL*>
2 Check combination setup        <*DETAIL*>
3 Multiple controller setup      <*DETAIL*>
```

5. Move the cursor to Check Combination Setup. You will see a screen similar to the following.

```
Interference Check
Combination to check
      Item 1    --    Item 2
 1  Rbt+Tool 1      Obstacle 1  DISABLED
 2  Rbt+Tool 1      Obstacle 1  DISABLED
 3  Rbt+Tool 1      Obstacle 1  DISABLED
 4  Rbt+Tool 1      Obstacle 1  DISABLED
 5  Rbt+Tool 1      Obstacle 1  DISABLED
 6  Rbt+Tool 1      Obstacle 1  DISABLED
 7  Rbt+Tool 1      Obstacle 1  DISABLED
 8  Rbt+Tool 1      Obstacle 1  DISABLED
 9  Rbt+Tool 1      Obstacle 1  DISABLED
 10 Rbt+Tool 1      Obstacle 1  DISABLED
```

6. Move the cursor to the desired combination and define the following parameters: *Rbt+Tool* or *Obstacle* .
7. Move the cursor to Rbt+Tool (or Obstacle) and press F3, [CHOICE], to change the model. You will see a screen similar to the following and you can select the model type. You must also specify Rbt+Tool for the robot and model tool case, or Obstacle, for the obstacle case.

```
1  Rbt+Tool
2  Obstacle
Interference Check
Combination to check
   Item1     -      Item2
1 Rbt+Tool  1 -   Obstacle 1      DISABLED
2 Rbt+Tool  1 -   Obstacle 1      DISABLED
3 Rbt+Tool  1 -   Obstacle 1      DISABLED
```

8. Move the cursor to definition of model number and type the number directly with the cursor on the number.

```
Interference Check
Combination to check
     Item 1     -       Item 2
1   Rbt+Tool 1  -      Obstacle 1  DISABLED
2   Rbt+Tool 1  -      Obstacle 1  DISABLED
Old Value: 1
```

9. Move the cursor to ENABLED or DISABLED and press F3, ENABLED, or F4, DISABLED, to change the setting. When the definition is disabled, the Interference Checking function is invalid to this combination pair.

16.4 SETTING UP INTERFERENCE CHECKING FOR MULTIPLE ROBOT APPLICATIONS

16.4.1 Overview

When using the Interference Check function with multiple controllers, setup is required to define what types of robots are used and where each robot base is located with respect to all other robot bases. You can set up a total of five robots together, each on a separate controller.

Each robot and controller must be calibrated with respect to every other robot and controller in the cell. For a three controller three robot application, Robot 1, Robot 2 and Robot 3 are interfering together. Robot 1 needs calibration data for Robot 2 and Robot 3. Robot 1 is a host robot to both remote robots 2 and 3. Also, Robot 2 needs calibration data for Robot 1 and Robot 3. Robot 2 is a host robot to both remote robots 1 and 3. Finally, Robot 3 needs calibration data for Robot 1 and Robot 2. Robot 3 is a host robot to both remote robots 1 and 2.

By defining interference pairs (host-remote) each host robot controller can derive a default interference zone for each one of its remote robots, based on the calibration data and the working envelope data for the pair.

16.4.2 Host Robot and Priority List

In a multiple robot application, each robot is on a different controller. Interference Checking allows you to use up to 5 robots on different controllers. Since the robots used may all be different, it is required to number each robot, as well as define priority levels for each robot. By defining priority levels, you can avoid gridlock situations, when two or more robots are waiting for each other.

Use Procedure 16-6 when defining the Host Robot and Priority List.

Table 16–19. Interference Check Robot No. and Priority Table Screen Items

ITEM	DESCRIPTION
Current robot no.	This item allows you to enter the current robot number.
Robot No. - Priority	This item allows you to define the priority level of robot numbers 1-5.

Table 16–20. Interference Check Remote Robot Setup Screen Items

ITEM	DESCRIPTION
Host robot number	This item defines the robot number of the host robot.
Robot number	This item defines the robot number of each robot in the multiple robot system.
Type	This item defines the robot type of each robot in the multiple robot system.
Calib done	This item defines the status of calibration for each robot of the multiple robot system.

Procedure 16-6 Specification of Host Robot and Priority List

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Interference.
5. Move the cursor to Multiple Controller Setup and press ENTER.

```
Interference Check
Multiple Controller Setup
1 Remote robot setup              <*DETAIL*>
2 Interference pair setup         <*DETAIL*>
3 Robot priority setup            <*DETAIL*>
```

6. Move the cursor to Robot Priority Setup, and press ENTER to display the Robot List and Priority Table screen.
7. Move the cursor on Current Robot # and type the number host robot directly.

```
Interference Check
Robot List and Priority Table
1  Host Robot No.:1
2  Robot 1    -  Priority     1
3  Robot 2    -  Priority     2
4  Robot 3    -  Priority     3
5  Robot 4    -  Priority     4
6  Robot 5    -  Priority     5
```

8. Move the cursor to the desired remote robot priority and type a value for the priority level directly.

```
Interference Check
Robot List and Priority Table
1  Host Robot No.:1
2  Robot 1    -  Priority     1
3  Robot 2    -  Priority     2
4  Robot 3    -  Priority     3
5  Robot 4    -  Priority     4
6  Robot 5    -  Priority     5
```

9. After defining the host robot number and the priority levels of all the robots in the cell, press F3, DONE to validate your selection.



Caution

The priority list must be identical in all the controllers that you set up for the Interference Checking function. Otherwise, Interference Checking will not operate properly.

Note Perform Step 10 through Step 16 to set up remote robots. If you are using three robots on separate controllers and the host is robot #2, repeat Step 10 through Step 16 for robots 1 and 3.

10. Press MENUS.

11. Select SETUP.

12. Press F1, [TYPE].

13. Select Interference. You will see a screen similar to the following.

```
Interference Check
Setup list of Interference Checking
1 Model setup                    <*DETAIL*>
2 Check combination setup        <*DETAIL*>
3 Multiple controller setup      <*DETAIL*>
```

14. Move the cursor to Multiple Controller Setup, and press ENTER to display the Multiple Controller Setup screen.

```
Interference Check
Multiple Controller Setup
1 Remote robot setup             <*DETAIL*>
2 Interference pair setup        <*DETAIL*>
3 Robot priority setup           <*DETAIL*>
```

15. Move the cursor on Remote Robot Setup, and press ENTER to display the Remote Robot Setup screen.

```
Interference Setup
  Host Robot #: 1
  Robot #          Type        Calib done
1 Robot 1        R-2000i         FALSE
2 Robot 2        R-2000i         FALSE
3 Robot 3        R-2000i         FALSE
4 Robot 4        R-2000i         FALSE
5 Robot 5        R-2000i         FALSE
```

16. To define the robot type, move the cursor to the desired remote robot and press F4, [CHOICE] to display the list of supported robots.

```
Interference Setup
  Host Robot #: 1
  Robot #          Type        Calib done
1 Robot 1          R-2000i     FALSE
2 Robot 2          R-2000i     FALSE
3 Robot 3          R-2000i     FALSE
4 Robot 4          R-2000i     FALSE
5 Robot 5          R-2000i     FALSE
```

a. Select the robot type from the list of supported robots and press ENTER.

```
1  R-2000i
2
3
4
Interference Check
2 Robot 2          R-2000i        FALSE
3 Robot 3          R-2000i        FALSE
4 Robot 4          R-2000i        FALSE
5 Robot 5          R-2000i        FALSE
```

b. To validate your selection press F2, CHANGE. This step also sets the radius of the working envelope automatically based on your selection of robot type. If you do not complete this step, the working envelope of the remote robot will not be defined correctly.

```
Interference Setup
  Host Robot #: 1
  Robot #          Type        Calib done
1 Robot 1           R-2000i    FALSE
2 Robot 2           R-2000i    FALSE
3 Robot 3           R-2000i    FALSE
4 Robot 4           R-2000i    FALSE
5 Robot 5           R-2000i    FALSE
                  Envelope radius was set
```

16.4.3 Combination Check Pairs

There are ten checking combination pairs available. You can define the robot model and static obstacle pair you want to check. You can also enable or disable a checking pair at any time. The robot model number refers to the robot group number and the obstacle number refers to the obstacle number in the obstacle set up list. Procedure 16-7 to setup a combination pair for interference zone checking between multiple robots.

Procedure 16-7 Setting Up a Combination Pair for Interference Zone Checking between Multiple Robots

Steps

1. From the combination pair list screen, select the combination pair you wish to setup. You will see a screen similar to the following.

```
Interference Check
Combination to check
      Item 1    --    Item 2
 1  Rbt+Tool 1      Obstacle 1  DISABLED
 2  Rbt+Tool 1      Obstacle 1  DISABLED
 3  Rbt+Tool 1      Obstacle 1  DISABLED
 4  Rbt+Tool 1      Obstacle 1  DISABLED
 5  Rbt+Tool 1      Obstacle 1  DISABLED
 6  Rbt+Tool 1      Obstacle 1  DISABLED
 7  Rbt+Tool 1      Obstacle 1  DISABLED
 8  Rbt+Tool 1      Obstacle 1  DISABLED
 9  Rbt+Tool 1      Obstacle 1  DISABLED
```

2. Press F2, DETAIL to access the Combination DETAIL screen

```
Interference Check
Combination to check
 Combination[ 1]
 1  Input:                       DI   [  0]
 2  Output:                      DO   [  0]
 3  Combination Type:                    1
      Enter 0 for Critical Zone
            1 for Obstacle
            2 for Virtual Fence
 4  Other Robot No.:                     0
```

3. Press F4, [CHOICE] to select the type of input/output signals you wish to use for the controller communication.

```
Interference Check
1 DI
2 RI
3 WI
4
Interference Check
 Combination[  1]
 1  Input:                       DI   [  0]
 2  Output:                      DO   [  0]
 3  Combination Type:                    1
      Enter 0 for Critical Zone
            1 for Obstacle
            2 for Virtual Fence
 4  Other Robot No.:                     0
```

4. Enter the index you want for the input and output signals.

```
Interference Check
Combination to check
Combination [  1]
 1  Input:                        DI   [  2]
 2  Output:                       DO   [  2]
 3  Combination Type:                     1
      Enter 0 for Critical Zone
            1 for Obstacle
            2 for Virtual Fence
 4  Other Robot No.:                      0
Old Value: 0
```

5. Set the Combination Type to 0. You will see a screen similar to the following.

```
Interference Check
Combination to check
Combination [  1]
 1  Input:                        DI   [   2]
 2  Output:                       DO   [   2]
 3  Combination Type:                      0
      Enter 0 for Critical Zone
            1 for Obstacle
            2 for Virtual Fence
 4  Other Robot No.:                      0
Old Value: 1
```

Note If the Combination Type is set to 1, the interference check function will ignore the input and output signals and will treat the interference zone as an obstacle. If the Combination Type is set to 0, the interference check function uses the input and output signals to lock and release the interference zone. If the Combination Type is set to 2, the interference check function will ignore the input and output signals and will treat the interference zone as a virtual fence, the robot will be confined in a user defined virtual fence (enclosure), an alarm will be posted if any portion of the robot and the tooling is outside the user defined enclosure. Virtual fence supports the following model shapes: Sphere, Cylinder, Infinite Plane, and Box.

Warning

When Constant Path option is enabled, 100% general override is required during Critical Zone (Combination Type 0) checking with multiple robots, i.e. 100% general override required only during motion entering and leaving the Critical Zone. There is no general override limitation for Critical Zone checking when Constant Path option is disabled.

6. You must set the other robot number to determine which robot in the work cell is associated with the pair. Set the other robot number to 0 for multiple robots on the same controller.

Warning

You must set this parameter correctly for the function to work properly. If the other robot parameter is not set properly, handling of priority conflicts will not work. This could result in a deadlock situation or a collision between two robots.

7. Press F3, DONE to validate your entries. You will see a screen similar to the following.

```
Interference Check
Combination to check
Combination [  1]
 1  Input:                        DI  [   2]
 2  Output:                       DO  [   2]
 3  Combination Type:                     0
      Enter 0 for Critical Zone
            1 for Obstacle
            2 for Virtual Fence
 4  Other Robot No.:                      0
```

8. Move your cursor to ENABLED or DISABLED and press F3, ENABLED or F4, DISABLED, to change the setting. When the combination pair is disabled, the Interference Checking function will not work for this combination pair. You will see a screen similar to the following.

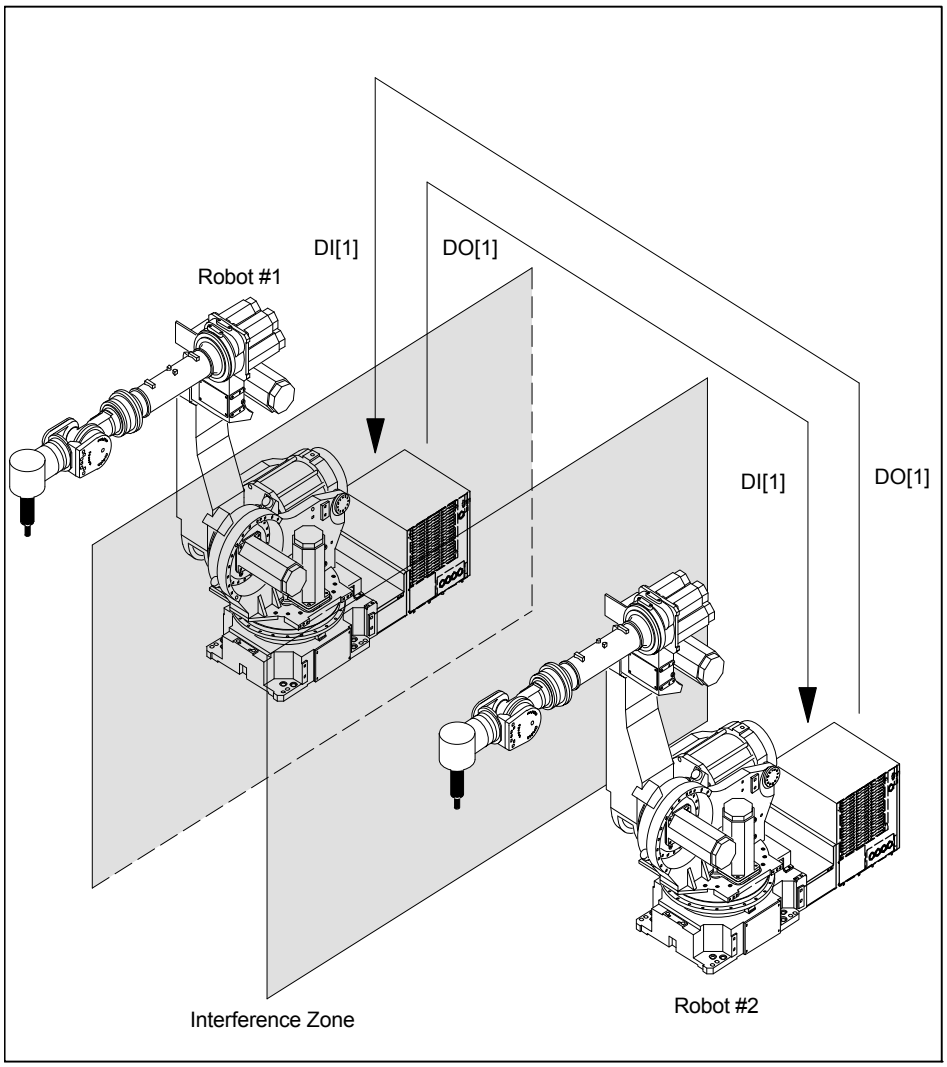
```
Interference Check
Combination to check
      Item 1    --    Item 2
 1  Rbt+Tool 1      Obstacle 1  ENABLED
 2  Rbt+Tool 1      Obstacle 1  DISABLED
 3  Rbt+Tool 1      Obstacle 1  DISABLED
 4  Rbt+Tool 1      Obstacle 1  DISABLED
 5  Rbt+Tool 1      Obstacle 1  DISABLED
 6  Rbt+Tool 1      Obstacle 1  DISABLED
 7  Rbt+Tool 1      Obstacle 1  DISABLED
 8  Rbt+Tool 1      Obstacle 1  DISABLED
 9  Rbt+Tool 1      Obstacle 1  DISABLED
```

16.4.4 Automatic Setup of Default Interference Zone Between Robot Pairs

Instead of creating interference zones manually using the obstacle setup procedures and defining combination pairs, it is possible to create a set of default interference zones and combination pairs for a given pair of robots.

Based on calibration data and the reach of each of the undressed robots, the interference zone checking function will create two infinite planes representing the overlap of the two robots workspaces. The corresponding combination pairs will also be created and ENABLED.

Figure 16–18. Robots on Different Controllers with Interference Zone and I/O



Direct Entry Calibration

Calibration remote robots with respect to the host robot process of determining the location and orientation of the remote robot's world frame with respect to the host robot's world frame.

Use Procedure 16-8 , the direct entry method, to calibrate the remote robots

Table 16–21. Interference Check Multiple Controller Setup Screen Items

ITEM	DESCRIPTION
Multiple Controller Setup	This item allows you to view Remote Robot Setup or Interference Pair Setup.

Table 16–22. Interference Check Multiple Controller Setup Screen Items

ITEM	DESCRIPTION
Remote Robot Setup	This item allows you to display the Remote Robot Setup screen.
Interference Pair Setup	This item allows you to display the Interference Pair Setup screen.

Note This procedure is only for remote robots. If you are using three robots on separate controllers and the host robot is robot #2, repeat Procedure 16-6 for robots 1 and 3.

Procedure 16-8 Direct Entry Calibration

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Interference. You will see a screen similar to the following.

```
Interference Check
Setup list of Interference Checking
1 Model setup                    <*DETAIL*>
2 Check combination setup        <*DETAIL*>
3 Multiple controller setup      <*DETAIL*>
```

5. Move the cursor to Multiple Controller Setup, and press ENTER to display the Multiple Controller Setup screen.

```
Interference Check
Multiple Controller Setup
1 Remote robot setup             <*DETAIL*>
2 Interference pair setup        <*DETAIL*>
3 Robot priority setup           <*DETAIL*>
```

6. Move the cursor to Remote Robot Setup, and press ENTER to display the Remote Robot Setup screen.

```
Interference Setup
  Host Robot #: 1
  Robot #          Type        Calib done
1 Robot 1           R-2000i    FALSE
2 Robot 2           R-2000i    FALSE
3 Robot 3           R-2000i    FALSE
4 Robot 4           R-2000i    FALSE
5 Robot 5           R-2000i    FALSE
```

7. Move the cursor to the desired remote robot and press F3, CALIB to display the calibration screen.

```
Interference Check
Calibration Direct Entry
Remote Robot #: 2
1 X:              0.000
2 Y:              0.000
3 Z:              0.000
4 W:              0.000
5 P:              0.000
6 R:              0.000
```

8. Press F3, [METHOD] and select Direct Entry.
9. Press F4, CHANGE to edit the calibration data. The cursor is now positioned on the first line of the calibration data. Move the cursor to the desired field and type the correct value.
10. After editing the calibration data, press F4, COMPLETE to validate your changes.

Calibration Using The Three Point Method

This method of calibration consists of teaching three points common to the host robot and remote robot. You must set up the Tool Center Point of all the robots before you start calibration using the Three Point Method.

Note The precision of the TCP affects the accuracy of the calibration.

The position of the three points chosen for the calibration also affects the accuracy of the calibration procedure. For example, do not teach positions in which the robot is close to a singularity or stroke limit. If the three points are too close to each other, or almost aligned on a straight line, the system will not be able to calculate good calibration data. The taught points must be at least 400 mm away from each other, and in a location easy to reach for both robots.

For multi-controller systems, it is impossible to record the position of the taught point for the remote robot. You can only record the position of the host robot. You must enter the position of the remote robot manually.

When calibrating the pair Robot 2 and Robot 3, you might calibrate Robot 3 (remote) with respect to Robot 2 (host) at the same time you calibrate Robot 2 (remote) with respect to Robot 3 (host). This allows you to teach the three points only once for both calibrations.

Note This procedure is only for remote robots. If you are using three robots on separate controllers and the host robot is robot #2, repeat Procedure 16-6 for robots 1 and 3.



Warning

The Interference Checking function is not active during setup and during calibration. Be careful not to collide the two robots into each other. If the two robots collide into each other, you might cause serious injury to personnel, and damage equipment.

Use Procedure 16-9 to calibrate the robot using the three point method.

Procedure 16-9 Calibration Using the Three Point Method

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Interference. You will see a screen similar to the following.

```
Interference Check
Setup list of Interference Checking
1 Model setup                    <*DETAIL*>
2 Check combination setup        <*DETAIL*>
3 Multiple controller setup      <*DETAIL*>
```

5. Move the cursor to Multiple Controller Setup, and press ENTER to display the Multiple Controller Setup screen.

```
Interference Check
Multiple Controller Setup
1 Remote robot setup             <*DETAIL*>
2 Interference pair setup        <*DETAIL*>
3 Robot priority setup           <*DETAIL*>
```

6. Move the cursor to Remote Robot Setup, and press ENTER to display the Remote Robot Setup screen.

```
Interference Setup
Remote Robot Setup
  Host Robot #: 1
  Robot #          Type        Calib done
1 Robot 1           R-2000i    FALSE
2 Robot 2           R-2000i    FALSE
3 Robot 3           R-2000i    FALSE
4 Robot 4           R-2000i    FALSE
5 Robot 5           R-2000i    FALSE
```

7. Move the cursor to the desired remote robot, and press F3, CALIB to display the calibration screen.

```
Interference Check
Calibration Direct Entry
Remote Robot #: 2
1 X:              0.000
2 Y:              0.000
3 Z:              0.000
4 W:              0.000
5 P:              0.000
6 R:              0.000
```

8. After inside the calibration screen, press F3, [METHOD] and select Three Point.
9. Press F4, CHANGE to change the calibration data. You will see a screen similar to the following.

```
Interference Check
 Calibration 3 Point Entry
 Remote Robot #: 2
 X:00000.000  Y:00000.000  Z:00000.000
 W:00000.000  P:00000.000  R:00000.000
            Host          Remote
1 Pos 1:    UNINIT        UNINIT
2 Pos 2:    UNINIT        UNINIT
3 Pos 3:    UNINIT        UNINIT
```

10. Move the cursor on Pos 1. Jog each robot alternatively so that the TCP of both robots are on the same point as Pos 1.

11. Record the position of the host robot by pressing SHIFT and F4, RECORD at the same time. When the position of the host robot is recorded successfully, the message *Position was Recorded* is displayed at the bottom of the screen. The display for Pos 1 of the host robot is also changed from UNINIT to RECORDED.

```
Interference Check
 Calibration 3 Point Entry
 Remote Robot #: 2
 X:00000.000  Y:00000.000  Z:00000.000
 W:00000.000  P:00000.000  R:00000.000
            Host         Remote
1 Pos 1:    RECORDED     UNINIT
2 Pos 2:    UNINIT       UNINIT
3 Pos 3:    UNINIT       UNINIT
                  Position was recorded
```

12. Leaving the cursor on Pos 1, press F4, EDIT. Since it is impossible to record the position for the remote robot, you must edit it manually.

13. On the Position Edit screen, type the values of x, y, z, w, p and r that are displayed on the position screen of the remote robot's teach pendant. The TCP remote robot must still be at Pos 1, where the host robot TCP is located. Press F4, COMPLETE to validate the Pos 1 data.

```
Interference Check
Calibration 3 Point Entry
Remote Robot #: 2
1 X:              350.000
2 Y:              -1030.000
3 Z:              389.000
4 W:              0.000
5 P:              56.000
6 R:              0.000
                  Position was recorded
```

14. The display of position status for Pos 1 of the remote robot will be changed from UNINIT to RECORDED.

```
Interference Check
 Calibration 3 Point Entry
 Remote Robot #: 2
 X:00000.000  Y:00000.000  Z:00000.000
 W:00000.000  P:00000.000  R:00000.000
            Host         Remote
1 Pos 1:    RECORDED     RECORDED
2 Pos 2:    UNINIT       UNINIT
3 Pos 3:    UNINIT       UNINIT
                  Position was recorded
```

15. Repeat Step 10 through Step 14 for Pos 2 and Pos 3. After all three points have been recorded for both the host robot and the remote robot, the following screen will be displayed. All position status will be changed from UNINIT to RECORDED. Press F2, COMPLETE to finish the calibration for the remote robot #2.

Interference Pairs And Default Interference Zones

Before you can setup interference pairs and calculate the default interference zone for a pair of robots, the following conditions must be satisfied.

- Priority level is defined for the host and remote robots
- Remote robot is calibrated with respect to the host robot

To set up interference zones automatically for two robots, you must first define interference pairs. By defining which pairs of robots in your system actually interfere with each other, it is possible to compute a default interference zone for each pair.

The default interference zone is calculated based on the position of the remote robot base (where it is mounted) relative to the host robot base, and based on the size of the envelope of each robot that makes up the interference pair. You must define an input signal and an output signal for each interference pair. This I/O communication will allow each controller to know if the interference zone is occupied or released.

Note If two robots with a sizeable working envelope are mounted very close to each other, the default interference zone will take up most of the robots working envelopes.

Use Procedure 16-10 when you want to define an interference pair.

Table 16–23. Interference Check Zone Setup Screen Items

ITEM	DESCRIPTION
Interf. Pair no.	This item defines the interference pair number between 1 and 5.

Table 16–23. Interference Check Zone Setup Screen Items (Cont'd)

Host Rbt	This item defines the robot number for the host robot.
Remote Rbt	This item defines the robot number for the remote robot.

Procedure 16-10 Defining An Interference Pair

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Interference. You will see a screen similar to the following.

```
Interference Check
Setup list of Interference Checking
1 Model setup                    <*DETAIL*>
2 Check combination setup        <*DETAIL*>
3 Multiple controller setup      <*DETAIL*>
```

5. Move the cursor to Multiple Controller Setup, and press ENTER to display the Multiple Controller Setup screen. You will see a screen similar to the following.

```
Interference Check
Multiple Controller Setup
1 Remote robot setup             <*DETAIL*>
2 Interference pair setup        <*DETAIL*>
3 Robot priority setup           <*DETAIL*>
```

6. Move the cursor to Interference Pair Setup, and press ENTER to display the Multi-Robot Setup screen. You will see a screen similar to the following.

```
Interference Check
Multi-Robot Zone Setup
                       Host gp    Remote Rbt
1 Interference pair 1  1            1
2 Interference pair 2  1            1
3 Interference pair 3  1            1
4 Interference pair 4  1            1
5 Interference pair 5  1            1
```

7. To define an interference pair between the host robot (group 1 of the host controller), move the cursor to the pair you are defining to define the remote robot number.
8. To access the DETAIL screen for one of the five interference pairs, move the cursor to the desired pair, and press F2, DETAIL. You will see a screen similar to the following.

```
Interference Check
Multi-Robot Zone Setup
Interference pair #:            1
1 Host robot #:                 2
2 Remote Robot #:               3
3 Zone input signal:            DI[0]
4 Zone output signal:           DO[0]
```

9. Move the cursor to the zone input signal line to select type and press ENTER.
10. Press F4, [CHOICE], to make your selection.
11. Type the input signal number directly. If this value is 0, then the assigned input signal is always OFF.
12. Move the cursor on the zone output signal line to select the signal type.
13. Press F4, [CHOICE] to make your selection.
14. Type the output signal directly. If this value is 0, then the assigned output signal is always OFF.
15. Repeat Step 1 through Step 14 to access the Multi-Robot zone setup DETAIL screen.
16. From the Multi-Robot zone setup DETAIL screen, verify that all the information for the interference pairs is correct. If it is incorrect, repeat Step 1 through Step 14 to change the settings for the interference pair. If it is correct, press F3, CALC_ZON to compute the default interference zone. See Figure 16–18 .

```
Interference Check
Multi-Robot Zone Setup
Interference pair #:            1
1 Host robot #:                 2
2 Remote Robot #:               3
3 Zone input signal:            DI[1]
4 Zone output signal:           DO[0]
```

Table 16–24. Interference Check Multi-Robot Zone Setup Screen Items

ITEM	DESCRIPTION
Interf. Pair no.	This item defines the interference pair number between 1 and 5.
Host Rbt	This item defines the robot number for the host robot.
Remote Rbt	This item defines the robot number for the remote robot.
Zone Input Signal	This item is the input I/O point data used for interference zone checking and is used only if stop on boundary is set to 0.
Zone Output Signal	This item is the output I/O point data used for the interference zone checking and is used only if stop on boundary is set to 0.

16.5 SETTING UP DUALARM SINGLE CONTROLLER INTERFERENCE CHECKING

In a DualARM single controller system, Interference Checking can be used with the following features:

- Each robot arm can define its own set of tool models and obstacles.
- A fixture that is defined as a critical zone can be used to ensure that only one robot arm can enter the critical zone at any point of time.
- Each robot arm can be defined as a dynamic obstacle to the other robot arm, and vice versa.

For robot to robot calibration, refer to Procedure 16-8 for Direct Entry Calibration, or Procedure 16-9 for Calibration Using the Three Point Method. In the Robot Model setup menu, you can view or add additional elements for each robot arm. In the Tool Model list menu, you can assign each tool model to the associated robot arm. In the Obstacle Model list menu, each obstacle can be taught by one of the two robot arms.

When a fixture is used as a critical zone as defined in the Check Combination Setup Menu, it can be used to ensure that only one arm can enter the specified critical area to avoid any potential collision between the two robot arms. Proper I/O signal setup is required to indicate that the critical zone is clear to enter or is occupied by another robot arm.

Each robot arm can be defined as a dynamic obstacle to another robot arm. Proper setup in the Check Combination Setup Menu is required, for example:

```
Item 1   --  Item 2
1 Rbt+Tool 1  Rbt+Tool 2    ENABLED
```

Note You cannot define any robot arm as a critical zone to another robot arm. CPU usage is very intensive when a dynamic obstacle is defined for robot arm to robot arm interference checking. If the computation time exceeds the system's capability, an alarm will be posted.

To optimize the CPU usage, specific individual robot model elements can be disabled if the specified robot model element protection is not required. For example, the first robot model element for J1 protection can be disabled if the J1 protection is not required. To disable the individual robot model element, refer to Step 7 in Procedure 16-2 , Setting Up The Robot Model .

16.6 INTERFERENCE CHECK TP INSTRUCTIONS

There are three Interference Check TP instructions:

- IC Comb Pair [] Enable
- IC Comb Pair [] Disable
- IC Zone [] Shift PR[]

IC Comb Pair [] Enable

This instruction enables specified combination pairs. For example "IC Comb Pair[1] Enable" will enable the combination pair #1 configured in the Check Combination Setup menu.

IC Comb Pair [] Disable

This instruction disables specified combination pairs. For example "IC Comb Pair[2] Disable" will disable the combination pair #2 configured in the Check Combination Setup menu.

IC Zone [] Shift PR[]

This instruction shifts specified critical zones with translation offsets specified in the position register. For example "IC Zone [3] Shift PR[9]" will shift the critical zone #3 with x, y, and z translation offset specified in PR[9].

- The specified critical zone is configured in the Obstacle Model Setup menu.
- x, y, and z translation offset is specified in the base frame group of the position register. For example, $ic_setup.$bfrmgrp is set to 1 by default; the x, y, and z translation offset is specified in the group 1 of the specified position register.
- w, p, and r rotation offset in the specified position register has no effect in this instruction.
- Each Zone Shift instruction will shift the original defined critical zone with x, y, and z offset.

16.7 STATUS AND RECOVERY

16.7.1 Overview

The Interference Checking function stops robot motion when the distance between the robot and the obstacle becomes too short and unsafe. When the robot stops because of an imminent collision with an obstacle or interference zone, one of the following errors will be posted.

- SSPC-201 - Interference Detected (G:1)
- SSPC-202 - Obstacle Detected (G:1)
- SSPC-202 - Obstacle Detected (G:1): The robot stops in the same manner as an E-STOP.
- SSPC-211 - Obstacle Detected (G:1): The robot does not stop in the same manner as an E-STOP, it decelerates to a stop.

If a collision or interference is detected, you can consult the interference check status screen to determine which robot or tool model element collided with a particular obstacle model element.

Confirmation of Halted Status

Use Procedure 16-11 to get information on the collision status.

Table 16–25. Interference Check Status Screen Items

ITEM	DESCRIPTION
Item 1	This item displays the number of the robot and number of the model element that collided with the element displayed in Item 2. For example, Robot 1 - 1 means robot 1, element 1, collided with the obstacle.
Item 2	This item displays the number of the obstacle and number of the model element that collided with the robot displayed in Item 1. For example, Obstacle 2 - 1 means obstacle 2, element 1, collided with the robot.

Procedure 16-11 Information on Collision Status

Steps

1. Press MENUS.
2. Select STATUS.
3. Press F1, [TYPE].
4. Select Interference Checking. You will see a screen similar to the following.

```
Interference Check
Interference Status
   Item 1     -             Item 2
1  Robot 1 Ele 5 -         Obstacle 1  1 Ele 1
```

Note This screen shows that element 5 of robot 1 has collided with element 1 of obstacle 1, and element 2 of tool 1 has collided with element 1 of obstacle 1.

16.7.2 Recovery and Interference Checking

You must use Obstacle Detection and Interference Zone Detection when Recovery and Interference Checking are used.

In case where the robot actually intruded the obstacle, the robot will stop in the same way as a E-STOP. When that happens, the only way to move the robot is to use the SHIFT-RESET mechanism to override the Interference Check function.

When a collision with an obstacle is detected, one of the following alarms is posted:

- SSPC-211: Obstacle Detected (G:1). The robot decelerates to a stop before actually colliding with the obstacle. You can reset this alarm and jog the robot away from the obstacle, or use backward execution when the collision occurs during program execution.
- SSPC-201: Obstacle Detected (G:1). The robot intruded the obstacle because a decelerated stop was not possible. The robot will stop in the same way as an E-STOP. When this happens, the only way to move the robot is to use the SHIFT-RESET mechanism to override the Interference Check function.

Warning

When SHIFT-RESET is used, the Interference Check function is disabled. When the Interference Check function is disabled, the robot will not be stopped if a collision or interference is detected. You could injure personnel or damage equipment.

Interference Zone Detection

When an interference with a zone is detected, the alarm SSPC-201 Interference Detected (G:1) is posted by the system. During jogging motion, this alarm will cause motion to be cancelled since another robot is inside the zone. In that situation, SHIFT-RESET is the only way to move the robot away from the interference.

Warning

When SHIFT-RESET is used, the Interference Check function is disabled. When the Interference Check function is disabled, the robot will not be stopped if a collision or interference is detected. You could injure personnel or damage equipment.

During program motion, the alarm SSPC-201 Interference Detected (G:1), will cause robot motion to hold. At this point, program execution is not cancelled, and the busy running signal is still on. Robot motion will resume when the interference zone is released and the robot has the correct level of priority.

16.8 ROBOT PARAMETERS

The default robot parameters and link dimensions for each supported robot are included in Figure 16–19 through Figure 16–22 . Each link of the robot is defined with respect to a coordinate frame. Also, each robot model (and dressout model) is defined with respect to one of the links of the robot.

Figure 16–19. Side view of the R-2000i/165F Robot and DH Frame Assignment and Dimensions

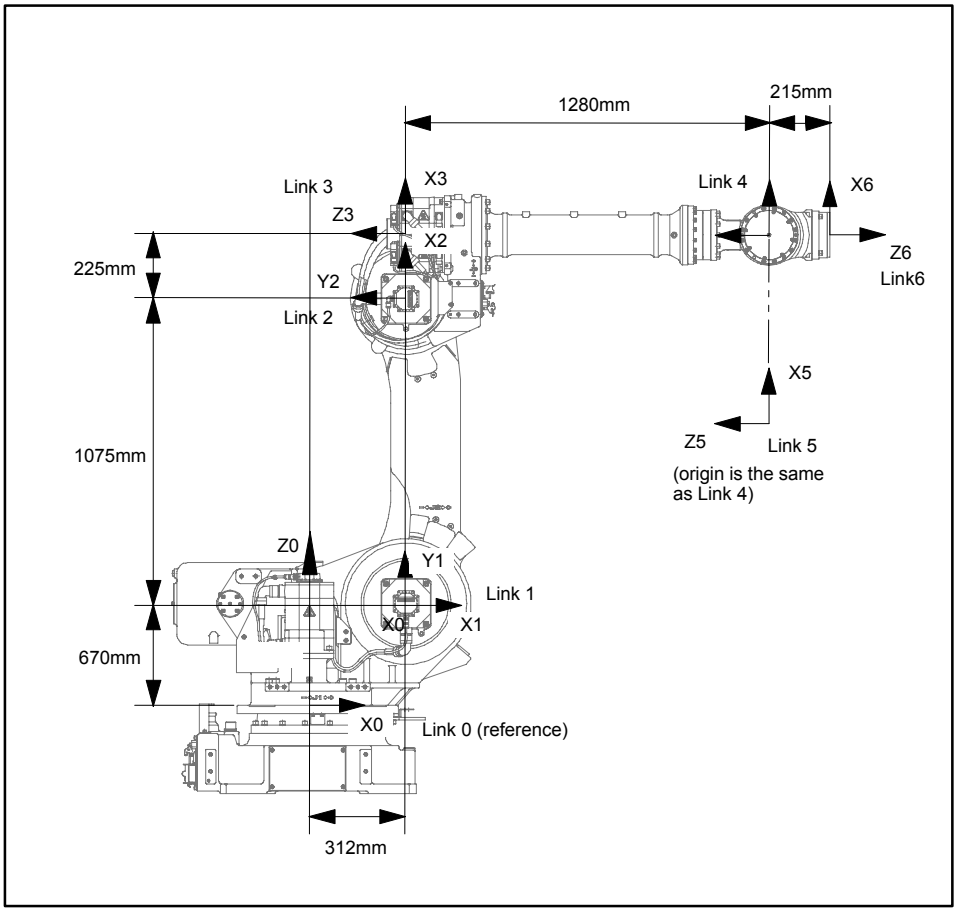


Figure 16–20. Isometric view of R-2000i/165F Robot and DH Frame Assignment

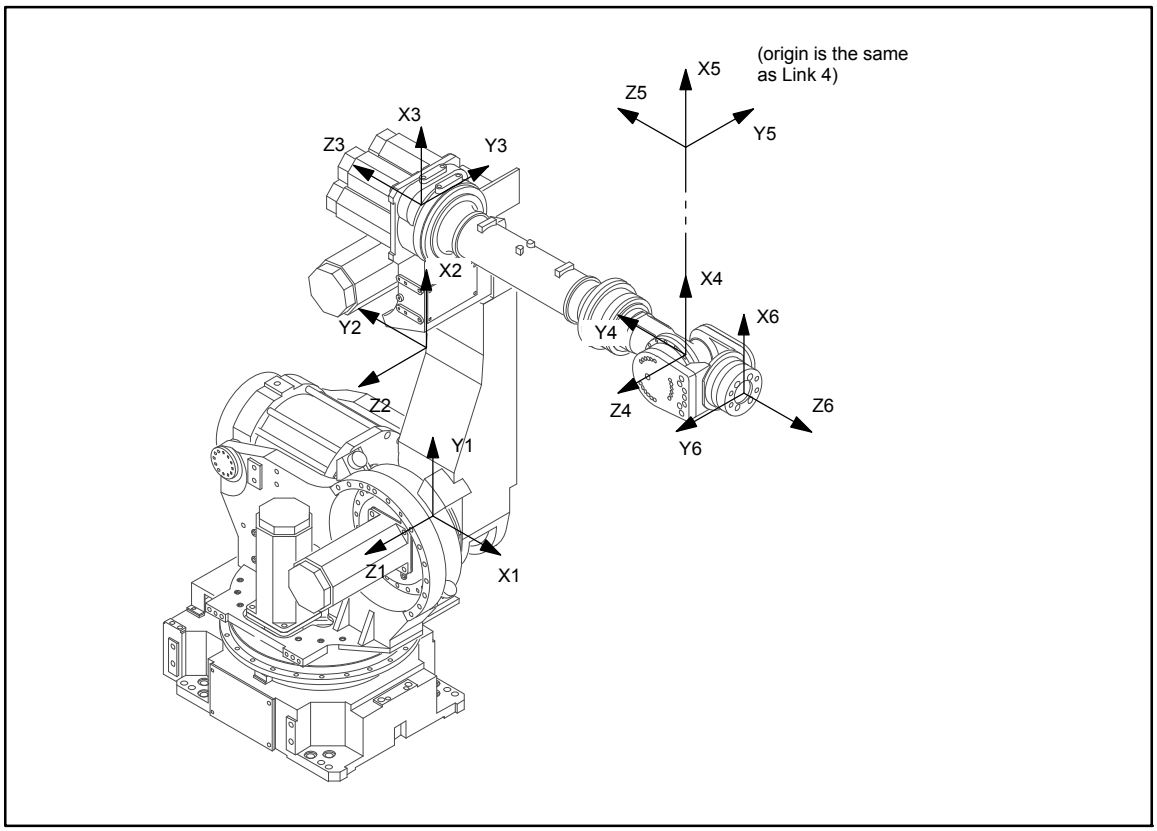


Figure 16–21. Side View of R-2000i/165F Robot with default robot model elements

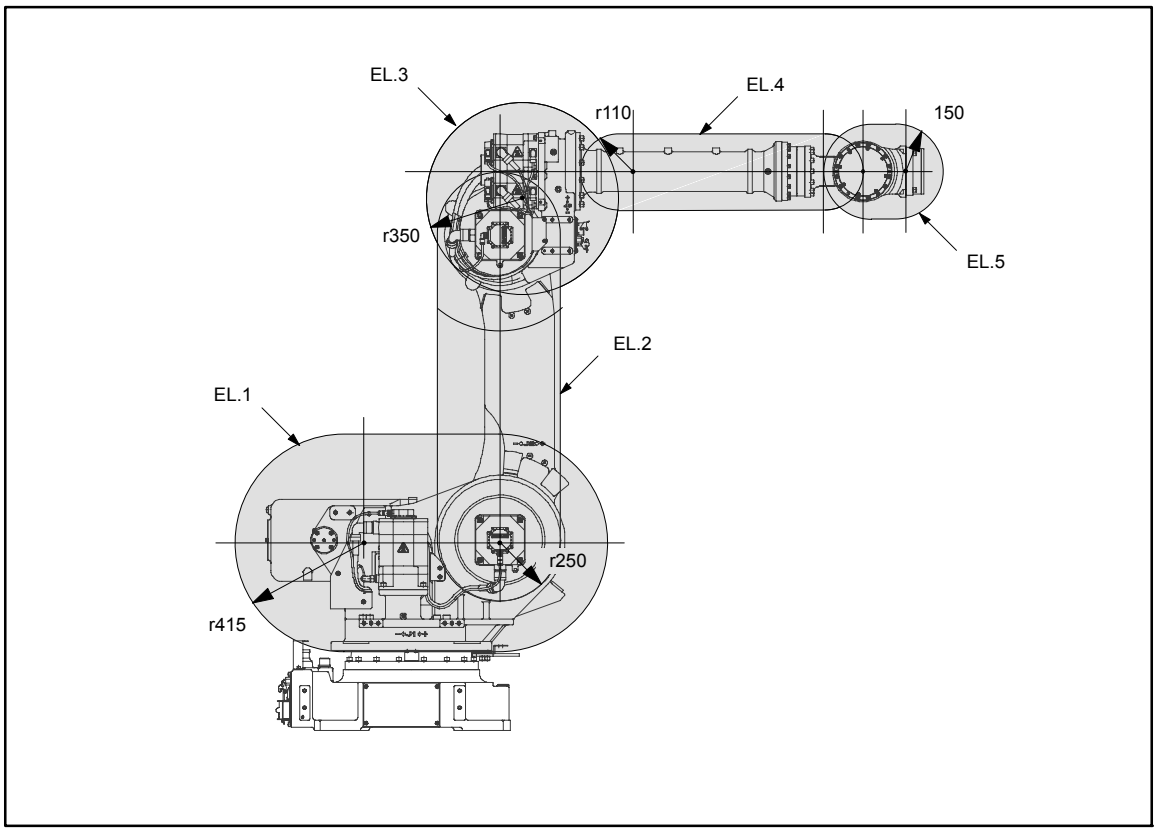


Figure 16–22. Front view of R-2000i/165F robot with default robot model elements

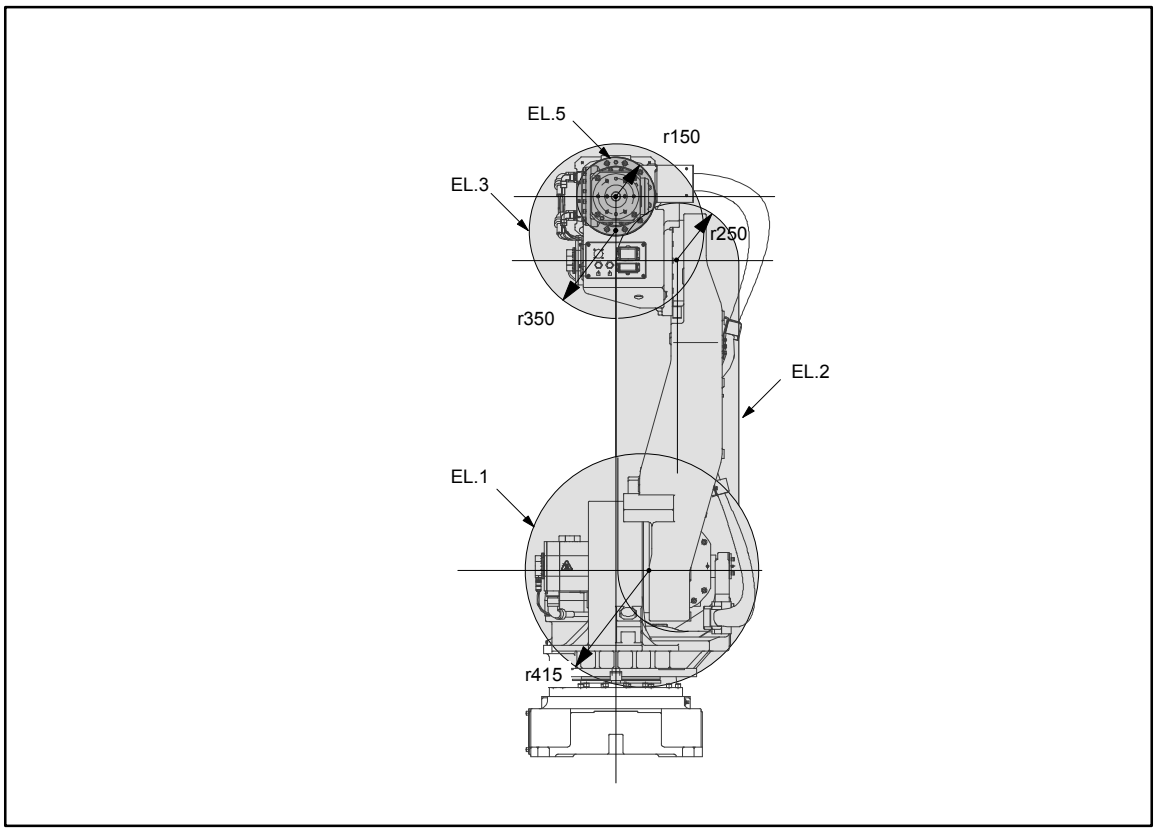


Table 16–26. R-2000I/165F Parameters

ITEM/EL	ELEMENT 1	ELEMENT 2	ELEMENT 3	ELEMENT 4	ELEMENT 5
Enabled/Disabled	ENABLED	ENABLED	ENABLED	ENABLED	ENABLED
Link number	1	2	3	3	5
Link type	NORMAL	NORMAL	NORMAL	NORMAL	NORMAL
Shape	Cylinder	Cylinder	Sphere	Cylinder	Cylinder
Size	415.0	250	350	110.0	150.0
Pos1 X	-482.0	0.0	-110.0	0.0	0.0
Pos1 Y	0.0	2.0	40.0	0.0	-15.0
Pos1 Z	-80.0	-250.0	-88.0	-418.0	12.0
Pos2 X	-102.0	-1075.0	0.0	0.0	0.0
Pos2 Y	0.0	2.0	0.0	0.0	-15.0
Pos2 Z	-80.0	-250.0	0.0	-1148.0	-113.0

Chapter 17

INTERFERENCE AVOIDANCE

Contents

Chapter 17 INTERFERENCE AVOIDANCE 17–1
17.1 Overview 17–2
17.2 Define Robot and Tool Model 17–3
17.3 Setting Up the AutoZone Space 17–5
17.4 Set Up the AutoZone Manager Member List and Calibrate Each Member 17–8
17.5 Set Up the AutoZone Schedule 17–10
17.6 Master Each Program Which is to be Protected from Collision Example 17–12
17.7 AutoZone Status 17–14
17.8 AutoZone Graphical Status 17–15
17.9 Interference Avoidance Limitations 17–20

17.1 Overview

The Interference Avoidance option uses a method of collision avoidance that maps each motion line in a user program to an *AutoZone*. The AutoZone approximates the total volume that is occupied at any time during the execution of the motion line. This volume can be visualized as a cavern-like shape that is carved out of a defined AutoZone space. The AutoZone space is a rectangular box aligned with a user frame coordinate system's major axes.

Up to 16 robots can be protected from collision with each other in the common AutoZone space. The robots can be on the same controller, or on multiple controllers.

AutoZone setup requires a two step process. After initial setup, a program is executed with AutoZone Mastering turned on. During this execution, the AutoZone software monitors the space occupied by the robot and tool model. This occupied space is digitized and efficiently stored in a file. Mastering is turned off and the robot will be protected from collision in the defined AutoZone space during execution with other similarly defined programs in other robot groups or other robot controllers.

The Interference Avoidance option is built on top of the Interference Check (IC) software. The IC robot model and tool model for the robots on each controller are used by the Interference Avoidance option. The other IC setup is not required for Interference Avoidance to function, but the IC functionality is fully compatible with the Interference Avoidance option. Therefore, in addition to AutoZones, the Interference Check software can protect against arm-to-arm collisions within the same controller and against entry to an occupied critical zone and against collision with a defined obstacle.

During program execution, the data representing the space occupied during a single motion statement execution is sent to the AutoZone manager. The AutoZone manager checks this 3-dimensional occupied space against the spaces already reserved and occupied by other robots. If there is no intersection between the spaces then the requesting program is allowed to execute the motion line. If there is an intersection then the requesting program motion will not be started until the robot responsible for the intersecting space requests a new space that has no intersections. Multiple motion statements can be grouped together to form a process zone or Interference Avoidance space. The robot is not allowed to enter this AutoZone Region until there is no intersection with the entire region with any other robot occupied space. The AutoZone setup requires the following steps:

1. Define the robot and tool model.
2. Set up the AutoZone space.
3. Set up the AutoZone manager member list and calibrate each member.
4. Set up the AutoZone schedule.
5. Master each program which is to be protected from collision.

17.2 Define Robot and Tool Model

Refer to Table 17–1 and Table 17–2 for information on defining the robot and tool model. Use Procedure 17-1 to define the robot and tool model.

Table 17–1. Interference Check Setup Items

ITEM	DESCRIPTION
Model Setup	This item allows you to display the Model Setup screen.
Check Combination Setup	This item allows you to display the Check Combination Setup screen.
Multiple Controller Setup	This item allows you to display the Multiple Controller Setup screen.
AutoZone Setup	This item allows you to display the AutoZone Setup screen.

Table 17–2. Interference Check Model Setup Items

ITEM	DESCRIPTION
Robot Model Setup	This item allows you to set up models for the robot and dressout.
Tool Model Setup	This item allows you to set up model(s) for the end of arm tooling.
Obstacle Model Setup	This item allows you to set up model(s) for obstacles.

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Interference. The existing Interference Check menus are used to define and enable the robot and tool models. You will see a screen similar to the following.

```
Interference Check    G1      JOINT 100%
Setup list of Interference Check
  1 Model setup               <*DETAIL*>
  2 Check combination setup   <*DETAIL*>
  3 Multiple controller setup <*DETAIL*>
  4 AutoZone setup            <*DETAIL*>
```

5. Select Model Setup, and press ENTER. You will see a screen similar to the following.

```
Interference Check    G1      JOINT 100%
                                     1/3
Setup list of Models
  1 Robot model setup         <*DETAIL*>
  2 Tool model setup          <*DETAIL*>
  3 Obstacle model setup      <*DETAIL*>
```

6. Robot models are already set up by the system and cannot be changed. Each link model can be enabled or disabled. You can add models for arm dressout from this menu if desired. Select Tool model setup, and press ENTER. You will see a screen similar to the following.

```
Interference Check    G1      JOINT 100 %
Tool Model list                      1/10
          Comment
1  AZ TOOL
2
3
4
5
6
7
8
9
```

7. Press F3, CLEAR to reset the tool data.

8. Press F2, DETAIL. Pressing DETAIL on the Tool Model List screen allows you to define the tool model for the selected tool. The tool model must be defined to protect the tooling mounted on the faceplate. You will see a screen similar to the following.

```
Interference Check    G1      JOINT 100 %
Model data                           1/5
Tool:  1 [AZ TOOL        ]
                Shape    Size
  1 ENABLED     Sphere   50
  2 DISABLED    Sphere   0
  3 DISABLED    Sphere   0
  4 DISABLED    Sphere   0
  5 DISABLED    Sphere   0
```

9. Each group utool can share a common tool model or they can reference different tool models. Press F4, UT_DATA. You will see a screen similar to the following.

```
Interference Check    G1      JOINT 100 %
UTOOL - Tool Model definition
GROUP : 1
        Utool No.      Tool No.
  1         1       -      1
  2         2       -      0
  3         3       -      0
  4         4       -      0
  5         5       -      0
  6         6       -      0
  7         7       -      0
  8         8       -      0
```

10. Press F2, GROUP to change the group number.

17.3 Setting Up the AutoZone Space

The AutoZone space defines the entire region that is protected from collision by the Interference Avoidance option. Portions of robots outside this space are not protected from collision by the Interference Avoidance option.

The space is aligned with respect to the specified Uframe of group[1] of the zone manager controller. The origin or Zone Reference of the AutoZone space is at the specified location with respect to Uframe. The Space Side Length can be specified as positive or negative numbers and indicate whether the zone edge is along the positive or negative direction from the Zone Reference along the Uframe axis. The space is represented by volumetric pixels, or *voxels*. The maximum number of voxels is fixed at 7040 for the AutoZone space. The default number along the X axis is 32, along the Y axis is 44, and along the Z axis is 5. The voxels can have different resolution or side length along each axis. The default side length is 50mm for the X and Y sides and 200mm for the Z side. This constructs the default Space Side Length to be 1600mm X, 2200mm Y, and 1000mm Z. You can increase or decrease the voxel side length or the space side length to change the number of voxels along each axis, adjust the length along the X axis first, Y axis second, then Z axis, the minimum number of voxels along each axis is 1.

Each voxel is represented by 1 bit in memory. The memory required for each voxelized space is 880 bytes plus some additional space for a total of 1 kilobyte. This is stored in compressed form in the file, FR5:**az**progname.bin occupying about 125 bytes of FROM. Additional data is stored in FR5:**az**progname.dat. However, the DRAM consumption of each voxelized motion statement is about 1KB per enabled space.

Refer to Table 17–3 and Table 17–4 for information on the AutoZone Space setup items. Use Procedure 17-2 to set up the AutoZone Space.

Table 17–3. AutoZone Space Setup Items

ITEM	DESCRIPTION
AutoZone Space Setup	This item allows you to display the AutoZone Space Setup Screen.
AutoZone Schedule	This item allows you to display the AutoZone Schedule Setup Screen.
AutoZone Manager Setup	This item allows you to display the AutoZone Manager Setup Screen.
AutoZone Test-Run Setup	This item allows you to display the AutoZone Test-Run Setup Screen.

Table 17–4. AutoZone Space DETAIL Setup Items

ITEM	DESCRIPTION
Space Id	This item indicates the number of the Space Id.
Enabled/Disabled	This item indicates the space is enabled or disabled.
Taught Member	This item indicates the member that taught the space.
Host Name	This item indicates the host name of the taught member.
Group	This item indicates the motion group number of the taught member.
Uframe	This item indicates the user frame number of the taught group with which the space is aligned.
Zone Reference in Uframe	This item indicates the origin corner of the taught space with respect to the specified Uframe.

Table 17–4. AutoZone Space DETAIL Setup Items (Cont'd)

ITEM	DESCRIPTION
Space Side Length	This item indicates the side length of the taught space. Either positive or negative number can be used to indicate the space edge is along the positive or negative direction of the specified Uframe. You can adjust the space side length to change the number of voxels along each axis, adjust along the X axis first, Y axis second, then Z axis.
Voxel Side Length	This item indicates the side length of the voxel. The default side length is 50mm for the X and Y sides and 200mm for the Z side. You can adjust the voxel side length to change the number of voxels along each axis, adjust along the X axis first, Y axis second, then Z axis.
Voxel Side Margin	This item indicates the margin which system can tolerate without re-mastering after a point has been re-touched up.
Number of Elements	The default number of voxels along the X axis is 32, along the Y axis is 44, and along the Z axis is 5. The minimum number of voxels along each axis is 1.

Procedure 17-2 Setting Up the AutoZone Space

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Interference. You will see a screen similar to the following.

```
Interference Check    G1      JOINT 100%
                                    1/4
Setup list of Interference Check
  1 Model setup               <*DETAIL*>
  2 Check combination setup   <*DETAIL*>
  3 Multiple controller setup <*DETAIL*>
  4 AutoZone setup            <*DETAIL*>
```

5. Select AutoZone setup. You will see a screen similar to the following.

```
AutoZone Setup         F1      JOINT 100%
                                     1/3
  1 AutoZone space setup       <*DETAIL*>
  2 AutoZone schedule          <*DETAIL*>
  3 AutoZone manager setup     <*DETAIL*>
  4 AutoZone test-run setup    <*DETAIL*>
```

6. Select AutoZone Space Setup. You will see a screen similar to the following.

```
AutoZone Space Setup
                                        1/7
   1 Space Id: 1  [******************]
   2 Enabled/Disabled:      DISABLED
     Taught Member:
      Host Name: ROBOT
      Group:     1
   3  Uframe:    3
   Zone Reference in Uframe: <*RECORD*>
   5  X:      0mm  Y:      0mm  Z:      0mm
   Space Side Length:
   6  X:  1600mm  Y:  2200mm  Z:  2000mm
   Voxel Side Length:
   7  X:  50.0mm  Y:  50.0mm  Z: 400.0mm
   Voxel Side Margin:
   8  X:   0.0mm  Y:   0.0mm  Z:   0.0mm
   Number of elements:
      X:    32    Y:    44    Z:     5
   Memory required: 0.88 KB
```

17.4 Set Up the AutoZone Manager Member List and Calibrate Each Member

Use Procedure 17-3 to set up and calibrate the AutoZone Manager.

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Interference.
5. Select AutoZone Setup.
6. Select AutoZone Manager Setup. You will see a screen similar to the following.

```
Zone Manager Setup    G1    JOINT 100%
                                  1/17
              Host Name         Grp Done
  1 Managr ROBOT
  2 Member ROBOT                  1 TRUE
  3 Member ROBOT                  2 TRUE
  4 Member ROBOT                  3 TRUE
  5 Member ROBOT                  4 TRUE
  6 Member                        0 FALSE
  7 Member                        0 FALSE
  8 Member                        0 FALSE
  9 Member                        0 FALSE
```

7. The Interference Avoidance Manager setup includes defining the controllers and robots that will be included in the Interference Avoidance function. If multiple controllers are used, RIPE must be already set up. For this case, press F4, [CHOICE] to display a list of robots included in the RIPE setup, the rosipcfg.xml file.

8. The Zone Manager Host Name should be the first robot listed in rosipcfg.xml. The first Member should be the same Host Name as the Manager. The first Member group should be 1. The first member is the Reference Group for future calibration menus. All the calibration frame go from the first Member world frame to the other Calibration Member world frame. See the following screen for an example.

```
Interference Check    G1    JOINT 100%
Calibration Direct entry
    Ref Grp:1                 ROBOT
    Cal Grp:2                 ROBOT
     X:                          0.000
     Y:                       1000.000
     Z:                          0.000
     W:                          0.000
     P:                          0.000
     R:                          0.000
```

9. Press F3, CALIB, softkey to display the Calibration Direct Entry screen. This screen displays the reference group and the calibration group. The reference group is the same host as the zone manager and is group 1. The calibration group is the group that is being calibrated. If you want to directly enter the calibration numbers then press F4, [CHANGE].

```
Interference Check    G1     JOINT 100 %
Calibration Direct entry           1/6
    Ref Grp:1                   ROBOT
    Cal Grp:2                   ROBOT
  1  X:                            0.000
  2  Y:                         1000.000
  3  Z:                            0.000
  4  W:                            0.000
  5  P:                            0.000
  6  R:                            0.000
```

10. Type in the calibration numbers, and press F4, COMPLETE when finished. This calibration frame is from the reference group world frame to the calibration group world frame.

11. From the Calibration Direct Entry screen, press F2, METHOD, and select the Three Point calibration method. You will see a screen similar to the following.

```
Interference Check   G1      JOINT 100%
                                   1/3
Calibration 3 Point entry
    Ref Grp:1                   ROBOT
    Cal Grp:2                   ROBOT
    X:    0.000 Y: 1000.000 Z:    0.000
    W:    0.000 P:    0.000 R:    0.000
       Ref Robot    -   Cal Robot
  1  Pos1: UNINIT    Pos1: UNINIT
  2  Pos2: UNINIT    Pos2: UNINIT
```

12. The Reference group and calibration group current calibration data is displayed. If you want to record the calibration points, press F4, [CHANGE].

When you press F5, RECORD, the current position of the selected robot will be recorded to the selected position. This record will work across controllers or on the same controller. You should touch three points which are far apart and encompass as much of the shared workspace as possible. The three points should not be co-linear. You can record the points by touching the TCP of robots to each other and record both robots. You need to cursor to one robot, press SHIFT-RECORD then cursor to the other robot and SHIFT-RECORD. You can also record them independently by touching points in the workspace first with one robot then with the other. Press F2, COMPLETE, when finished to calculate the calibration data. Press F3, CLEAR, to clear any existing data. Press F4, EDIT, to edit the point without having to move the robot if the target point coordinates are known.

17.5 Set Up the AutoZone Schedule

Refer to Table 17–5 . Use Procedure 17-4 to set up an AutoZone Schedule.

Table 17–5. AutoZone Schedule Setup Items

ITEM	DESCRIPTION
AutoZone Sch	The program AutoZone schedule number is specified in the program header. A different AutoZone schedule number can be specified in each AutoZone Start instruction.
AutoZone Recording	
Recording Interval	The recording interval governs how often the robot position data is voxelized during the motion when the Master Flag is ON. If the controller has heavy CPU loading because of many options or for other reasons, you can increase this interval to speed up the voxelization process.
Priority	Not Available. This item is the priority to resolve deadlock.
Progressive Flag Default: FALSE	The Progressive flag is used in a Process Zone or AutoZone Region section of a program. This is between the AutoZone Start and AutoZone End instruction. A Process Zone allows a robot to be guaranteed to complete a process without being halted by another robot. For most process applications, the robot should remain running along the path without interruption until the process (welding, cutting, routing, grinding, etc) is completed. FALSE indicates that one voxelized space will be used to represent the whole region. No other robot will be able to enter any portion of the region until the robot currently inside the region exits the region. Using the Progressive flag = FALSE also saves memory since only one voxelized space is used to represent all the motion lines between the AutoZone Start and AutoZone End instruction. If the Progressive flag is TRUE then as each motion statement executes between the AutoZone Start and AutoZone End instruction, only the space occupied by the remainder of the Process Zone will be reserved. This allows another robot to enter a region of the Process Zone after the first robot has passed the area and if it will not return to that area inside the Process Zone.

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Interference.

   The default values of the AutoZone schedule are usually acceptable.

5. Select AutoZone Setup.
6. Select AutoZone Schedule Setup. You will see a screen similar to the following.

```
AutoZone Schedule    G1     JOINT 100%
                                  1/4
  1 AutoZone Sch:  1 [AutoZone Sched1  ]
  AutoZone Recording:
  2 Recording Interval: every  3 ITP

  3 Priority                   1
  4 Progressive flag:          FALSE
```

17.6 Master Each Program Which is to be Protected from Collision Example

```
PRG1
                                                  1/13
    1:J  P[1] 100% FINE
    2:L  P[2] R[1:G1 SPD]mm/sec CNT100
     :
    3:  AutoZone Start [1]
    4:L  P[3] R[1:G1 SPD]mm/sec CNT100
     :
    5:L @P[4] R[1:G1 SPD]mm/sec CNT100
     :
    6:L  P[5] R[1:G1 SPD]mm/sec CNT100
     :
    7:L  P[6] R[1:G1 SPD]mm/sec CNT100
     :
    8:L  P[7] R[1:G1 SPD]mm/sec CNT100
     :
    9:L  P[8] R[1:G1 SPD]mm/sec CNT100
     :
   10:  AutoZone End
   11:L  P[9] R[1:G1 SPD]mm/sec CNT100
     :
   12:L  P[1] R[1:G1 SPD]mm/sec CNT100
```

This is an example of a program with AutoZone Start and AutoZone End. This region is known as a *Process Zone* or *AutoZone Region*. After the robot is granted permission to move to P[3] in Line 4, the whole space occupied by the robot until reaching P[8] on Line 9 is reserved. This guarantees that any welding or other process can be completed without interruption. If the Progressive flag in the AutoZone schedule is TRUE then when P[4], Line 5 is started the space occupied between P3 and P4 is released (unless the remainder of the motions re-enter the area).

```
Detail                G1      JOINT 100%
  AutoZone program data detail      1/2
  1  AutoZone Enabled:              Yes
  2  Global AutoZone Schedule ID:   1
```

The application data in the program header (Press DETAIL from the SELECT menu) allows AutoZone protection to be enabled or disabled for the program. You must turn on the global master flag in AutoZone test-run setup menu before the program mastering, and turn off the global master flag after mastering the program. The Global AutoZone schedule number indicates which AutoZone schedule will be used outside of the AutoZone Start/End region. A Cold start is not required for the program mastering to be effective. The program mastering is effective immediately after mastering and when the global master flag is turned OFF.

```
AutoZone test-run setup
                                 1/3
1 Global master override     TRUE
2 Master append mode         FALSE
3 AutoZone jogging enabled   FALSE
```

Table 17–6. AutoZone Test-Run Setup Items

ITEM	DESCRIPTION
Global master override	This item indicates global master flag. Set the global master flag to TRUE before the program mastering, and set the global master flag to FALSE after mastering the program.

Table 17–6. AutoZone Test-Run Setup Items (Cont'd)

Master append mode	This item indicates master append mode. When the append mode is enabled, re-mastering process will not delete the previous voxel data file, instead it will append the new voxel data to the existing data file, it is useful to support multiple program execution paths with the branch instruction.
AutoZone jogging enabled	This item indicates AutoZone jogging enabled. When AutoZone jogging is enabled, the jogged robot will be checked against other robots on the same or different controllers.

17.7 AutoZone Status

From the Interference Check status page, press F4, AUTOZONE, to display the AutoZone execution status map. This AutoZone status screen displays the programs being executed, including subprograms that are called from a main program. The status can be idle, running, or waiting. If a status line has waiting status then the number in the index column indicates the object (robot or program) is the cause of the wait. If a program is waiting because of interference with a robot which is idle and not currently running a program then that robot hostname and group number will be displayed following the list of executing programs.

```
AutoZone Status      G1     JOINT 100%
                                 1/20
   Host     Program      Status  Index
 1 ROBOT R1_470MN        running 0
 2 ROBOT R2_470MN        running 0
 3 ROBOT R3_470MN        running 0
 4 ROBOT R4_470MN        running 0
 5 ROBOT R2_47F10        running 8
 6 ROBOT R4_47PER        waiting 7
 7 ROBOT R1_47PER        waiting 8
 8 ROBOT R3_47PER        waiting 5
 9
```

In this example of a four-group system with a running program for each group, the main program (lines 1..4) for each group has called a subprogram (lines 5..8). The status of the teach pendant program R4_47PER indicates it is waiting on index 7 which corresponds to the teach pendant program R1_47PER. R4_47PER will not continue execution until the interference with R1_47PER is removed. Note that a paused program still retains the ownership of the AutoZone associated with the paused line.

This example shows a deadlock condition. R2_47F10 and R1_47PER are both waiting on R3_47PER but R3_47PER is waiting on R2_47F10. This deadlock condition could be prevented by using AutoZone regions (using AutoZone Start / AutoZone End) to reserve space needed by future motions.

A better way of preventing deadlock conditions is to plan the program execution sequence carefully.

```
AutoZone Status    G1    JOINT 100%
                                1/20
   Host    Program    Status Index
 1 ROBOT   R3_470MN   running 0
 2 ROBOT   R3_47F01   waiting 3
 3 ROBOT   group:2       idle 0
 4
 5
 6
 7
 8
 9
```

This example shows a running program being blocked by an idle robot that is not running a program. The waiting program, R3_47F01, is blocked by group 2. The waiting program can be resumed once group 2 is moved outside the AutoZone space requested by R3_47F01.

17.8 AutoZone Graphical Status

From the Interference Check status menu the GRAPHIC, F3, softkey will bring up a graphical display of the AutoZone space on the *i*Pendant. By default the XY view of the AutoZone space is displayed. This view is the projection of all the active AutoZones for each robot onto the XY plane. Because the AutoZones are a 3-D representation some information is lost with the projection. For instance, you cannot differentiate between zones when there is overlap of XY values at different Z values.

If you press F3, VIEW , the display toggles to XZ and YZ views. These are projections of the AutoZones onto the XZ and YZ planes. Example *i*Pendant displays and associated ROBOGUIDE images for each of the AutoZone status views follow. The ROBOGUIDE images are a snapshot during the execution of a program. As is apparent from the AutoZone status images, some robot paths are longer and reserve more AutoZone space than others. See Figure 17–1 through Figure 17–5 .

Actual robot 3-D snapshot postures are displayed on the right hand side of Figure 17–1 through Figure 17–3 , 3-D AutoZone data of one motion segment within the specified space projected on the 2-D plane are displayed on the left hand side of the corresponding figures.

Figure 17–1. Interference Avoidance Display

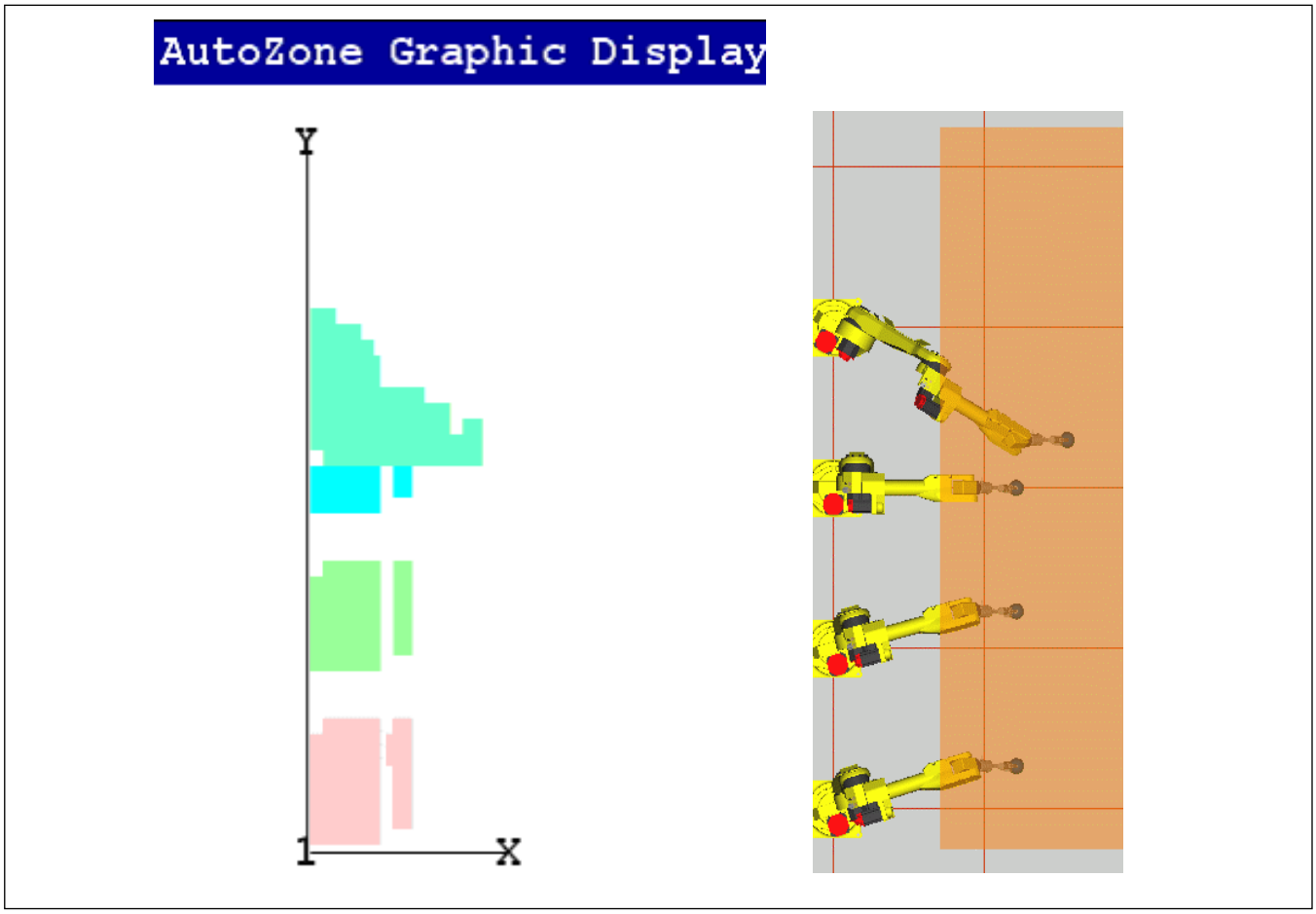


Figure 17–2. Interference Avoidance Display 2

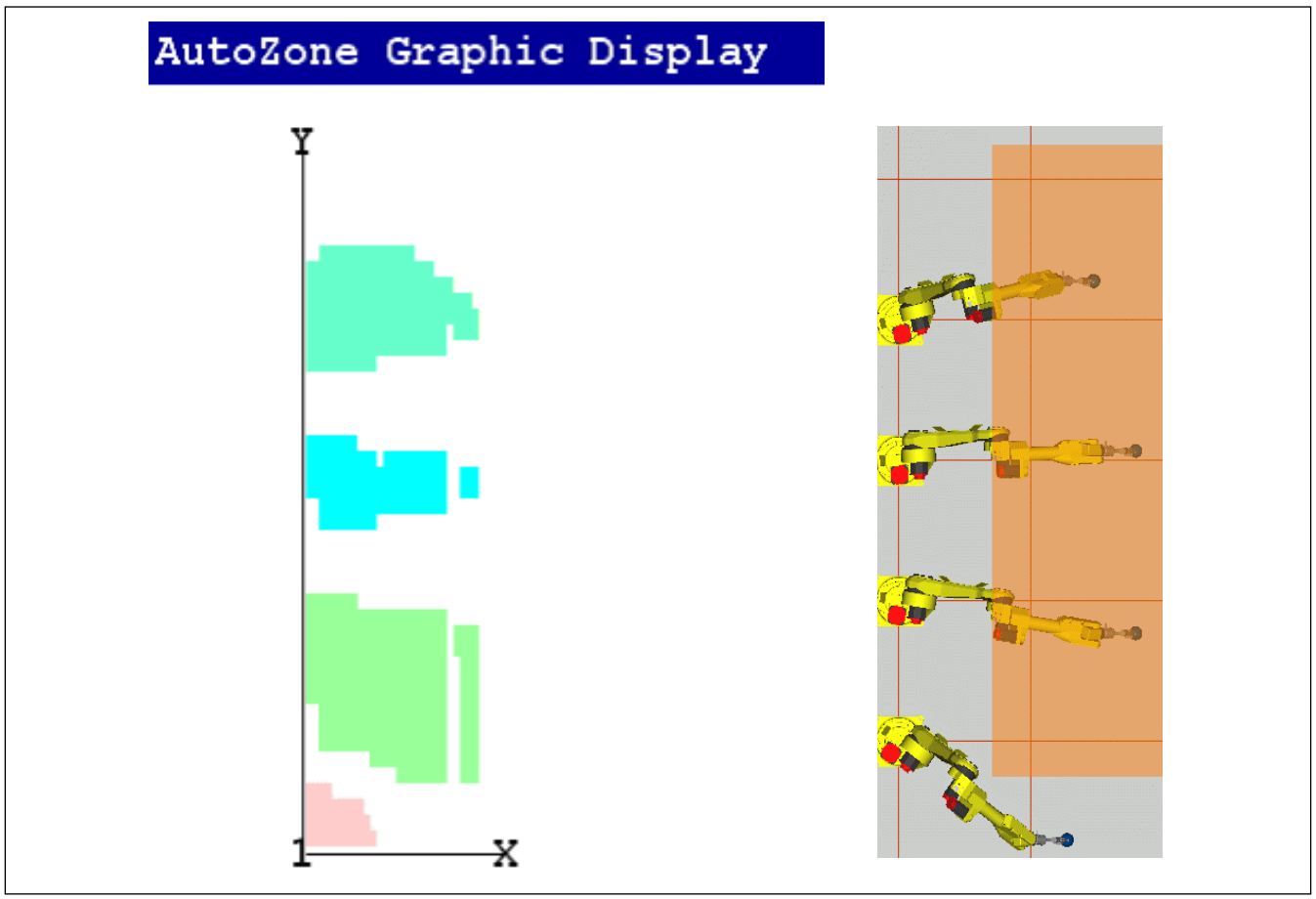


Figure 17–3. Interference Avoidance Display 3

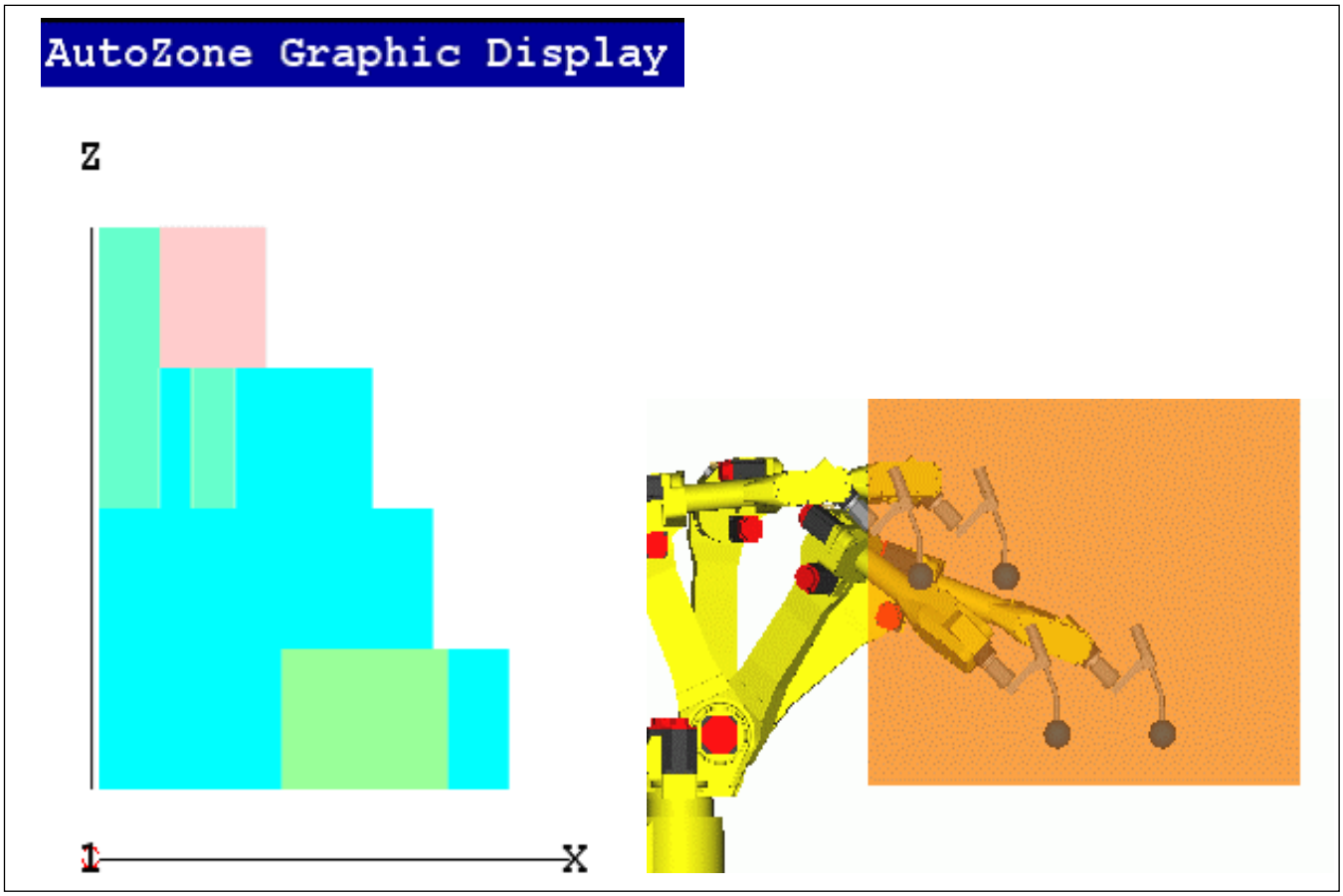


Figure 17–4 is a 2-D Y-Z plane projection of one motion segment associated with actual robot snapshot postures of Figure 17–5 .

Figure 17–4. Interference Avoidance Display 4

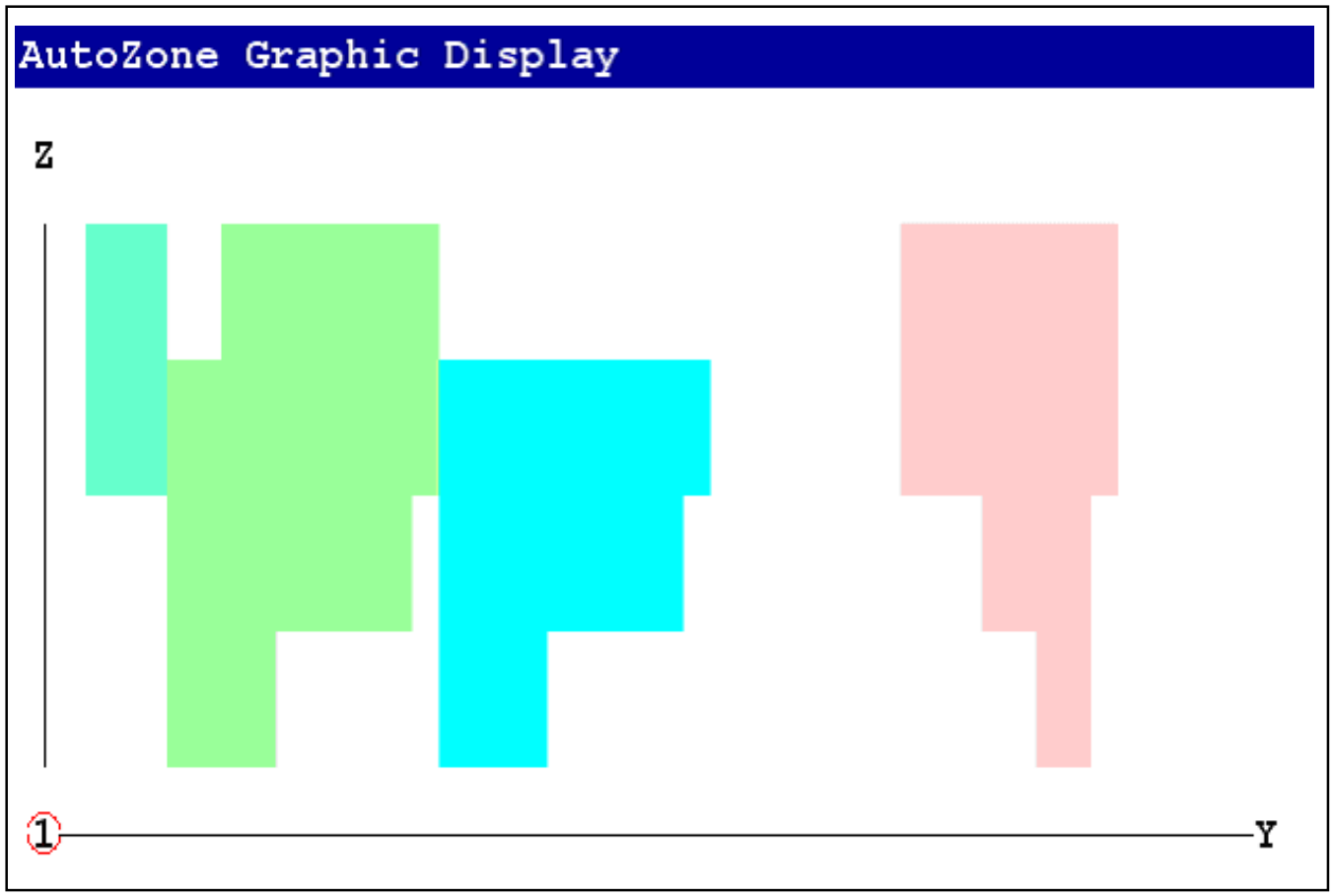
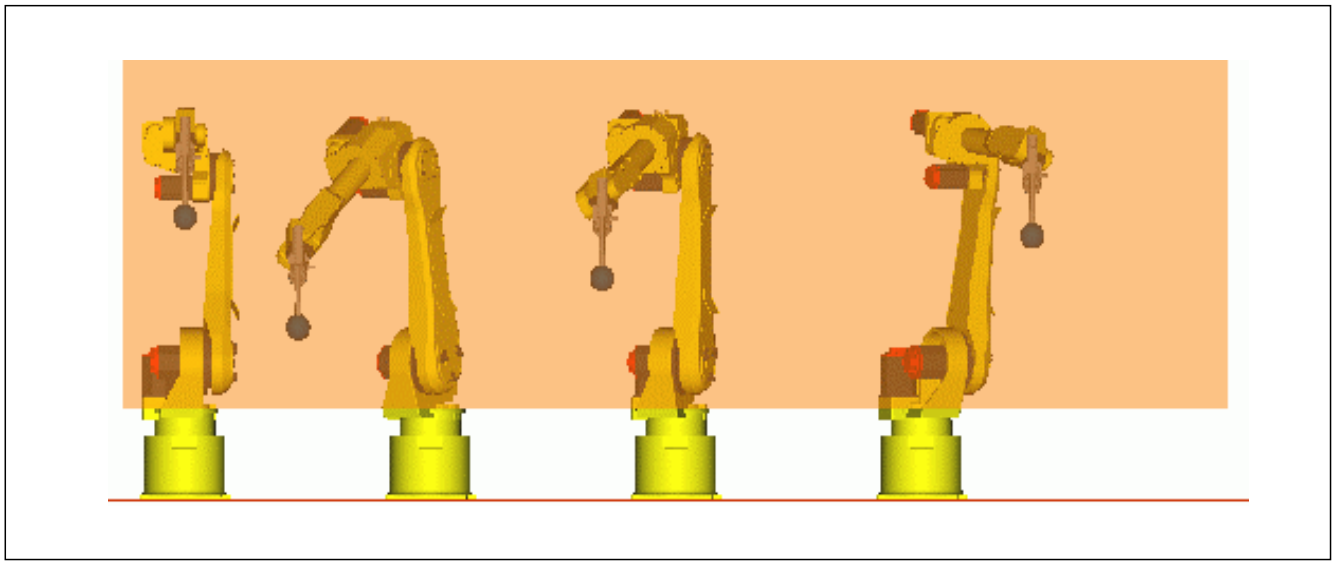


Figure 17–5. Interference Avoidance Display 5



17.9 Interference Avoidance Limitations

- Only segment based progressive Interference Avoidances are supported. Distance and time based progressive zones are not supported.
- Interference Avoidance does not protect the very first motion in TP program execution. The trajectory from the starting robot position to the first motion position is not included in the Interference Avoidance data. The first move in a CALLed program is protected by the Interference Avoidance data associated with the CALL statement. It is only the first motion in a new TP task that is not protected. ·
- When a program is paused and then jogged away:
  - — the motion from current position to paused position is not protected by the Interference Avoidance
  - — the motion from paused position to destination position is protected by the Interference Avoidance
- Skip line is not protected for example, move cursor from current line #2 to line #4 then shift-FWD is not protected. This is treated as the first motion in a program.
- "Skip, LBL[x]" motion option is not supported.
- Does not support dynamic motion path modifications that exceed the tolerance of the model setup and specified voxel side margin. The path used during mastering will be the path used for interference checking, regardless of difference in run-time offsets. The offsets themselves do not prohibit the use of Interference Avoidances. Large offsets for the following conditions are not supported:
  - — Offset from Position Register
  - — Vision Shift
  - — Dynamic RTCP
  - — Tracking softparts
    - – Thru Arc (TAST)
      - – Most path deviation of TAST is within the tolerance, i.e. the deviation is less than the voxel side margin.
    - – Line Tracking
    - – Robot Link motion of slave robots is not supported.
- Each enabled space is limited by maximum of 7040 voxels, requiring approximately 1Kb of DRAM memory.
- During the program mastering, user needs to execute all permutations of the branch instructions to collect complete voxel data.
- Only one zone manager is allowed. The zone manager handles a mazimum of 16 robots. All robots must be calibrated to the zone manager. If a robot and the zone manager cannot touch

common points with their TCP then the calibration data must be entered manually. The zone manager is the first robot listed in the RIPE setup, in the rosipcfg.xml file.

- Each motion line, each CALL statement, each JMP statement, and each conditional branch statement all require 1024 bytes of storage in DRAM and ~125 bytes FROM for each enabled space. There is some additional fixed overhead less than 100KB.
- If the program is not re-mastered after the program touchup, an alarm will be posted during the regular program execution with the global master flag turned off.
- Deadlock prevention is not supported. It is up to the user to ensure that no deadlock can occur when executing the programs.
- By default, there can be a maximum of 20 motion statements in an Interference Avoidance Start/End region if progressive mode is enabled. If progressive mode is disabled there is no limit to the number of statements in this region. There is no limit to the number of Interference Avoidance Start/End regions, except for the normal memory usage limitations. The number of motion statements allowed in an Interference Avoidance region can be increased by setting the system variable, $IC_AZ_CONF.$MAX_AZ_LINE.
- By default, there can be a maximum of 60 programs on a controller that are protected by Interference Avoidances. This number can be increased by setting the system variable, $IC_AZ_CONF.$MAX_AZ_PROG.
- Interference Check (R557) is required for the Interference Avoidance option
- Only those motion groups with full kinematics are supported. Positioners are not supported.
- AutoZone calibration must be done from the AutoZone menu. If Interference Check is used for critical zones or arm-to-arm interference checking then separate Interference Check calibration from the Interference Check menu is required.
- Hot start is not supported.

Chapter 18

TOUCH SENSING

Contents

Chapter 18	TOUCH SENSING	18–1
18.1	OVERVIEW	18–3
18.2	ASSIGNING TOUCH SENSING I/O	18–4
18.2.1	Touch Sensing I/O Overview	18–4
18.2.2	Touch Sensing Input Signal	18–5
18.2.3	Touch Sensing Enable/Disable Output Signal	18–5
18.2.4	Assigning the Touch Sensing Inputs and Outputs	18–6
18.3	SETTING UP TOUCH SENSING	18–7
18.3.1	Touch Sensing Setup Overview	18–7
18.3.2	Touch Frames	18–10
18.3.3	Search Pattern	18–16
18.3.4	Touch Schedule	18–22
18.4	TOUCH SENSING PROGRAMMING	18–28
18.4.1	Programming Overview	18–28
18.4.2	Touch Sensing Instructions	18–28
18.4.3	Touch Sensing Motion Option	18–29
18.4.4	Motion Instructions Used with Touch Sensing	18–30
18.4.5	Executing a Touch Sensing Program	18–30
18.4.6	Touch Sensing Robot Position Touchup	18–31
18.4.7	Programming Examples	18–32
18.5	TOUCH SENSING MASTERING	18–38
18.5.1	Mastering Overview	18–38
18.5.2	Mastering	18–38
18.5.3	Remastering	18–39
18.5.4	Offsets	18–40
18.5.5	Patterns	18–41
18.5.6	Master Flag	18–42
18.5.7	Touching Up Path Positions	18–44
18.5.8	Adding New Positions	18–46
18.5.9	Multiple Searches	18–47

18.5.10 Touching Up Search Start Positions 18–49
18.6 TOUCH SENSING HARDWARE 18–51
18.6.1 Hardware Overview 18–51
18.6.2 Touch Sensing Input Signal 18–51
18.6.3 Touch Sensing Enable/Disable Output Signal 18–51
18.6.4 Simple Low Voltage Touch Sense Detection Circuit 18–52
18.7 LASER ANALOG INPUT NON-CONTACT TOUCH SENSING 18–54
18.7.1 Overview 18–54
18.7.2 System Setup 18–54
18.7.3 Analog Input Touch I/O Setup 18–56
18.7.4 Analog Input Search Direction 18–57

18.1 OVERVIEW

Touch sensing (an optional feature) allows the robot to change a path automatically to compensate for object displacement.

Note For Multi-Arm Touch Sensing, refer to the *Multi-Arm Controller Option Manual.*

Touch sensing consists of:

- Moving the robot tool center point (TCP) toward the object using pre-defined robot motion, speed, and direction.
- Using an input signal to indicate that the robot has come into contact with the object.
- Storing the found location of the object, or *position offset information*, in position registers.
- Using the stored position to move the robot to the stored position, or using the stored position offset information to shift one or more positions in your welding program.

To use touch sensing you must:

- Set up the robot Tool Center Point (TCP) properly. Refer to Section 18.3 to set up the tool frame.
- Set up touch sensing hardware. The hardware monitors an input signal to determine when the robot comes into contact with the object.
- Assign I/O to enable and use the electrical interface circuit.
- Set up how the robot moves to the object and the type of position offset information that is stored.
- Set up a coordinated motion pair for coordinated motion touch sensing
- Create a touch sensing program.

See Figure 18–1 for an example of a program that includes touch sensing.

Figure 18–1. Example Program Including Touch Sensing Routine

INSTRUCTION	DESCRIPTION
`1:  J P[1] 100% Fine`	Teach a point in space
`2:    Search Start[3] PR[3]`	Use touch schedule 3. Store offset in pos. reg. 3
`3:  J P[2] 100% Fine`	Teach a search starting position
`4:  J P[3] 100% Fine Search [Y]`	Do a search motion in Y direction
`5:  J P[4] 100% Fine`	Teach a search start position
`6:  J P[5] 100% Fine Search [X]`	Do a search motion in X direction
`7:     Search End`	End of the search
`8:  J P[6] 100% Fine`	Teach an intermediate point (optional)
`9:    Touch Offset PR[3]`	The following points will be offset by PR[3]
`10: J P[7] 100% Fine`	P[7] is offset by PR[3]
`11:   Arc Start [1]`	Begin welding
`12: L P[8] 30IPM Fine`	P[8] is offset by PR[3]
`13:   Arc End [1]`	End welding
`14:   Touch Offset End`	End of offsetting positions

18.2 ASSIGNING TOUCH SENSING I/O

18.2.1 Touch Sensing I/O Overview

To use touch sensing you must assign the

- Input signal that the touch sensing circuit monitors to indicate when the robot has reached the object.
- Output signal that enables and disables the touch sensing circuit.

Note You must wire the necessary connections for the input and output signals to be used for touch sensing. The wire stick detection circuit on the process I/O board also can be used for touch sensing.

The controller supports numerous I/O options. If you decide to use an I/O point other than the standard, (such as a modular I/O), then the controller must be wired and configured correctly.

Note Some welding power supplies, such as the Lincoln Electric PowerWave models and the Miller Auto Invision provide internal touch sensing circuitry. These power supplies can be set up for the appropriate inputs and outputs when software configuration is performed.

18.2.2 Touch Sensing Input Signal

The touch sensing input signal indicating contact with a part is monitored by the touch sensing circuit. When the input is received, the current robot position is stored in a position register. Refer to Section 18.6 for more information about the touch sensing circuit.

Any of the following can be used as the touch sensing input signal:

- Welding Digital Inputs (WI) 1-8, found on the CRW1 Connector of the process I/O board.
- Digital Inputs (DI) 1-22, found on the CRM2A and CRM2B connectors of the process I/O board, or any Digital Inputs.
- Wire stick detection circuit input WSI (WDI+, WDI-), found on the CRW1 connector of the process I/O board.
- Robot Digital Inputs (RI) 1-8, found on the Axis Control PCB.
- Optional Digital I/O, (such as a Modular I/O or DeviceNet I/O).

You can also set up touch sensing to monitor the condition of any RO or DO signal as an input signal. When the selected output turns on during a touch sensing routine, the controller reads this as a received input signal.

18.2.3 Touch Sensing Enable/Disable Output Signal

Any of the following can be used to enable the touch sensing circuit:

- Robot Digital Outputs (RO).
- Digital Outputs (DO).
- Welding Digital Outputs (WO) 1-8 found on the CRW1 Connector of the Process I/O Board.
- Wire stick detection circuit enable WSE is an internal output on the process I/O board that enables the detection circuit and allows it to be used for touch sensing.



Caution

If a WO is assigned as the touch sensing input signal, the dedicated function it performs must be disabled. Refer to Section 18.2.4 .

Note To use touch sensing, the weld interface cable must be installed. If you are using a weld power supply for the touch sense circuit, the power source must be turned on for touch sensing to work.

Refer to the Connections section of the *Maintenance and Connection Manual* for connector location, pin configurations, and I/O signal specifications.

18.2.4 Assigning the Touch Sensing Inputs and Outputs

You must assign touch sensing inputs and outputs to match the hardware interface at your site. This involves assigning both input and output type and port number.

Touch sensing inputs are shown as **sensor ports** on the Touch Sense I/O SETUP screen. Touch sensing outputs are shown as **circuit ports** on the Touch Sense I/O Setup screen. Use Procedure 18-1 to assign touch sensing I/O.

Note After you have decided what I/O to use for touch sensing, you should add a comment to the selected I/O indicating that the I/O has been assigned to touch sensing. This is done using the SETUP menu. Refer to Chapter 3 *GENERAL SETUP* .

Procedure 18-1 Assigning Touch-Sensing Inputs and Outputs

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE]
4. Select Touch I/O. You will see a screen similar to the following.

```
Touch I/o Setup
      NAME                   VALUE
 Sensor port type              RDI
 Sensor port number            1
 Circuit port type             RDO
 Circuit port number           1
```

5. Assign Sensor (input) and Circuit (output) types as follows:
   a. Move the cursor to the line you want to assign.
   b. Press [F4], CHOICE.
   c. Move the cursor to the desired input/output type.
   d. Press ENTER.

   Note The allowable input range for the sensor and circuit ports is from 1 to 1024. The software checks the validity of the port type and port number when running your program that includes touch sensing. If the port type or number is invalid, the system displays an I/O invalid error message.

6. Assign Sensor and Circuit number:
   a. Move the cursor to the line you want to assign.
   b. Type the value and press ENTER.

   Note After the input signal has been wired and assigned, perform a test to verify that it is connected properly. The input signal condition can be monitored from the I/O Menu.

18.3 SETTING UP TOUCH SENSING

18.3.1 Touch Sensing Setup Overview

Search motions locate an object and store the found location, or *position offset information* , of the object in a position register. Search motions use

- Touch frames
- Touch patterns
- Touch schedules

A *touch frame* determines the direction of the search motion. The search motion is actually a programmed move along the x, y, or z axis of a selected touch frame.

For touch sensing with coordinated motion, you can select the touch frame relative to the UFRAME of the robot (follower) or the coordinated frame of the reference group (leader). The reference group must be set for the leader group so that the search direction will be relative to that group.

Typically, only one search motion is used for each search direction. Some *search patterns* require two search motions in each of two search directions for the software to calculate an angular offset. Search patterns determine the type of information stored in the position register. The stored information is either the found position or position offset information depending on the search pattern used.

Up to five search motions in one search direction can be done to improve the accuracy of locating an object. When more than one search motion in a direction is used, the software calculates an average value of the searches and uses the average for the offset calculation except when using the search pattern *1D+Rotate, 2D +Rotate,* or *3D+Rotate.*

Also, you can include a maximum of 15 searches between the program instructions SEARCH START and SEARCH END. Refer to Section 18.3.3 .

Touch schedules allow you to set up the conditions that define the search motions. These conditions include the position register, touch frame and search pattern to use; the robot speed and motion type; and other conditions.

Figure 18–2 , Figure 18–3 , and Figure 18–4 represent how search motions are used in a program.

Touch Sense defaults to using position register 32.

Figure 18–2. Search Using Searches in One Direction

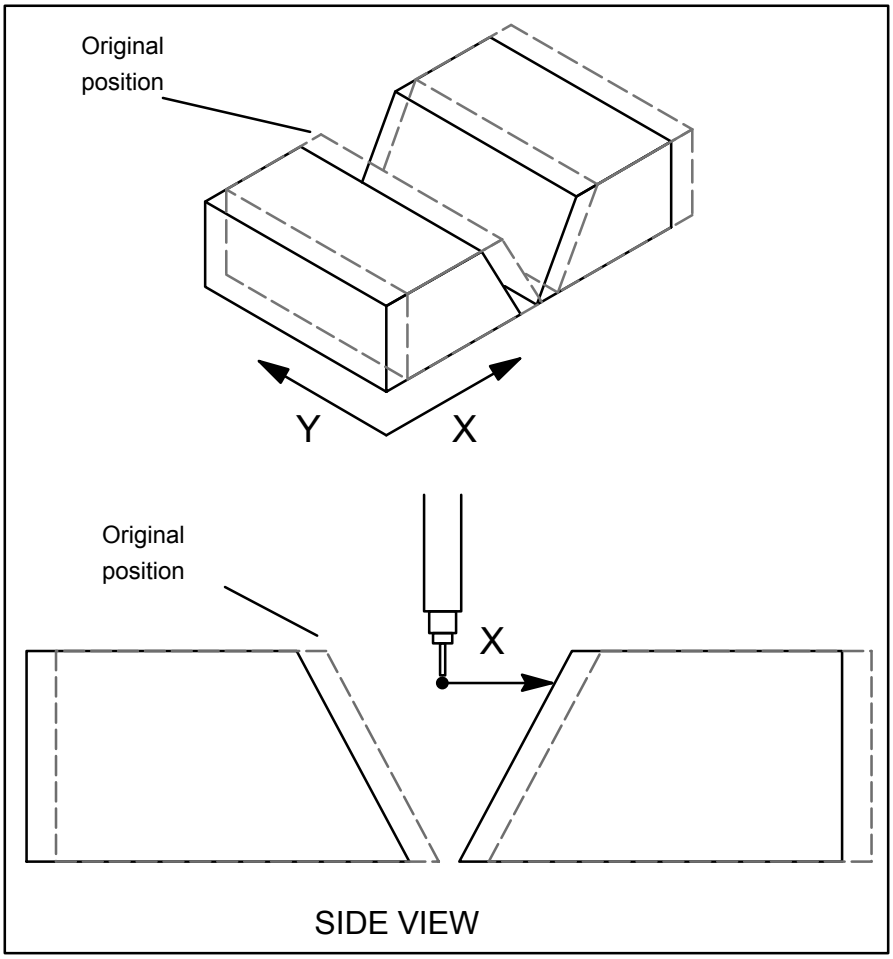


Figure 18–3. Search Using Offsets in Two Dimensions

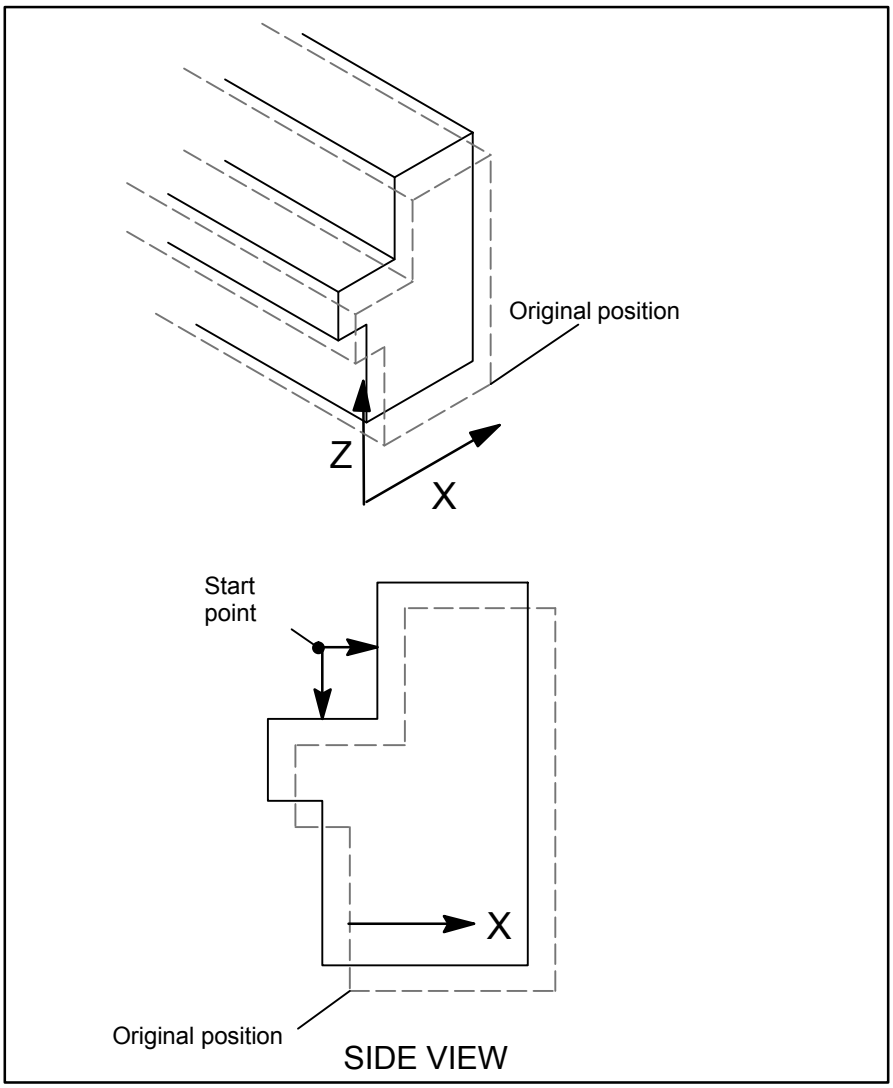
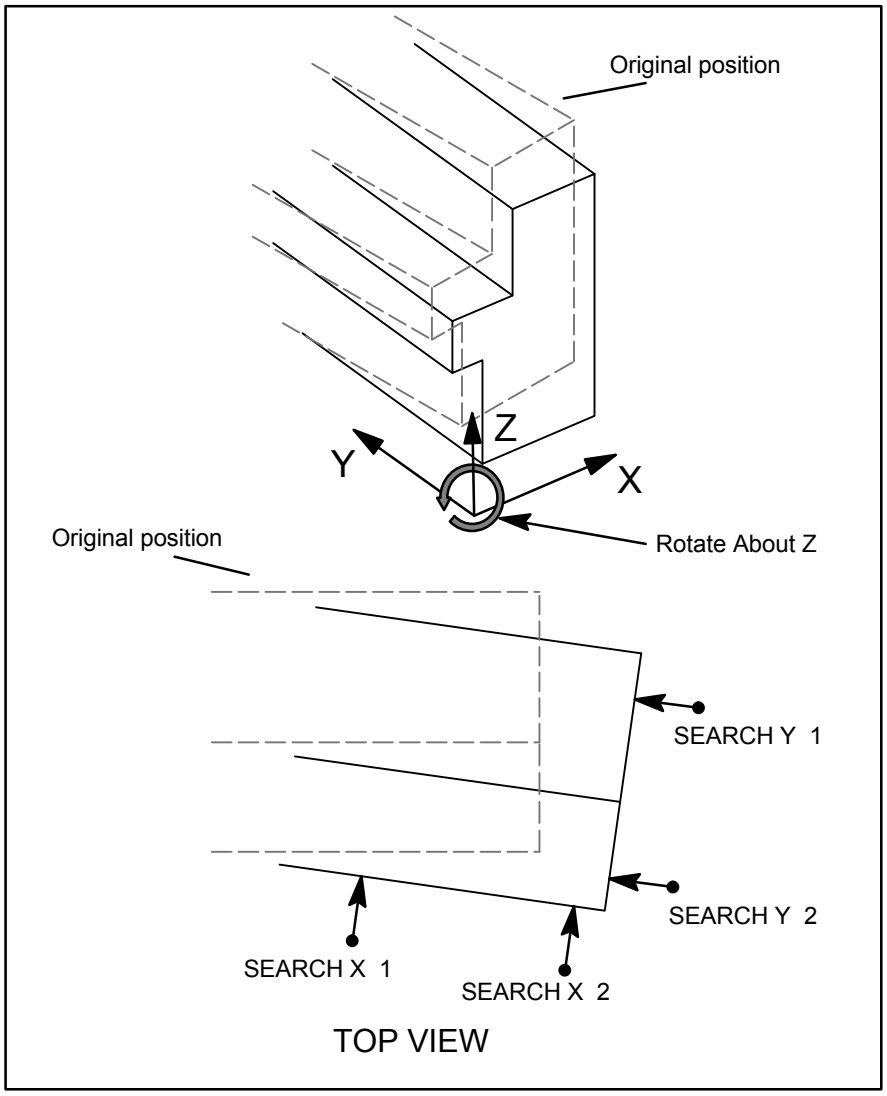


Figure 18–4. Search Using Two Search Motions in Two Different Directions to Obtain X and Y Offset and Rotation about Z



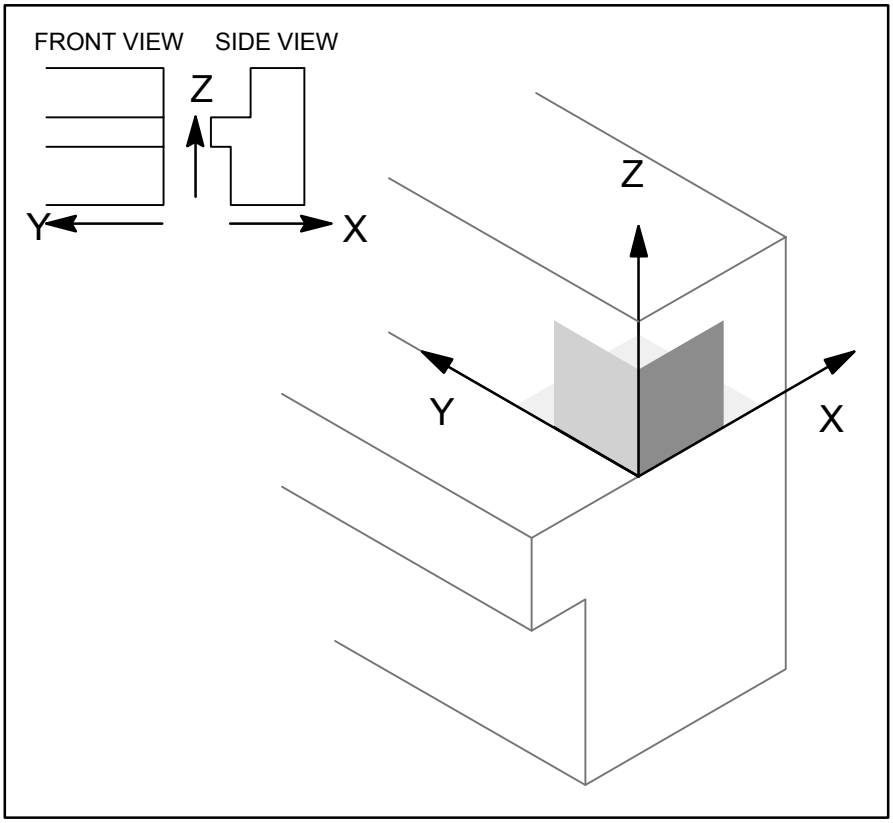
The characteristics of a search motion are controlled by variables set in touch schedules. The x, y, or z movements in a search motion are aligned with an object by using one of the touch frames.

18.3.2 Touch Frames

A touch frame determines the motion direction of the robot TCP. A touch frame is defined by three points. The first point defines the *origin* , or starting point. The second point defines the positive x direction of the touch frame. The third point defines the positive x-y plane.

Figure 18–5 shows a touch frame and how it is used in a touch sensing program. The orientation of the touch frame to the object is arbitrary in Figure 18–5 . The positive x axis could be aligned with the current z direction. This would re-define positive z to be in the opposite direction of the current positive x direction.

Figure 18–5. Touch Frame Used in a Program



Note You can set up a maximum of 32 touch frames. Touch frames are set up using the touch frame screen in the SETUP menu.

Note You must define a touch frame before you perform a search motion in a program.

There are two ways to define touch frames: the teaching method and the direct entry method. The teaching method defines the touch frame by recording three points. The direct entry method defines the touch frame by the rotation angle value you enter in the Touch Sense SETUP screen. For Direct Entry, use Procedure 18-3 . For the Teach Method, use Procedure 18-2 .

Table 18–1 lists and describes the items you must set to define the touch frame.

Table 18–1. Touch Frame Setup Items

ITEM	DESCRIPTION
Frame Number	This item specifies the number of the touch frame you want to define.
Reference Group	This item specifies the reference group to which the touch frame is relative: • **0**: There is no reference group and no leader group; touch frame is relative to the robot itself. • **1** : Touch frame is relative to the UFRAME of the robot (follower) • **2** : Touch frame is relative to the coordinated frame of robot group 2 (leader) ** • **3** : Touch frame is relative to the coordinated frame of robot group 3 (leader) **, *** • **4** : Touch frame is relative to the coordinated frame of robot group 4 (leader) ** • **5** : Touch frame is relative to the coordinated frame of robot group 5 (leader)** The coordinated frame is established between a positioner/robot (leader) and a robot (follower) motion group. Refer to the *Coordinated Motion Setup and Operations Manual.* *** In the case of DualARM systems with Touch Sensing, standard Touch Sensing for group 2 robot references group 2.
Robot Group	This item specifies the motion group performing the search move and to which the offset is applied. When using DualARM or Multi-Robot systems this can be group 1 or 2. For single group systems this will always be 1. When using multi-arm simultaneous touch sensing, each touch process robot can define its own touch frame within the same touch frame number.
Direct Entry	
Rotate about X	This item specifies the rotation about X for the touch frame.
Rotate about Y	This item specifies the rotation about Y for touch frame.
Rotate about Z	This item specifies the rotation about Z for touch frame.
Teach Method	
Origin	This item allows you to record the origin of the touch frame.
+X direction	This item allows you to define the +X direction of the touch frame.
+Y direction	This item allows you to define the +Y direction of the touch frame.

Note When Reference Group is not equal to 1 for robot #1 or 2 for robot #2, the touch frame changes with the coordinated motion frame, but the display of the Rotate about X, Y, and Z items remains unchanged.

Use Procedure 18-2 to define your touch frame by using the teaching method. Use Procedure 18-3 to define your touch frame by using the direct entry method.

Procedure 18-2 Setting Up a Touch Frame Using the Teaching Method

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Touch Frame. You will see a screen similar to the following.

```
Touch Frame Setup
Frame #: 7  Ref Grp: 2   Robot Grp: 1
  Direct Entry:
    Rotate about X :      11.732
    Rotate about Y :       2.274
    Rotate about Z :     -20.195
  Teach Method:
    Origin         :      USED
    +X direction   :      USED
    +Y direction   :      USED
```

5. Move the cursor to Frame Number. Type the number of the frame to define and press ENTER.
6. Move the cursor to Reference Grp. Type the number of the reference group and press ENTER.
7.



Caution

Searches and offsets for coordinated motion cannot be mixed with non-coordinated. That is, you cannot do searches for coordinated motion, then do searches for non-coordinated motion, and then use the offset from the coordinated motion search. You must use the offsets for coordinated motion searches before doing any non-coordinated searches. Similarly, you must use any offsets from non-coordinated motion searches before doing any coordinated motion searches.

Move to Robot Grp and select the appropriate group number.

Note The Robot Group refers to the robot that is making the search moves. The Reference Group is the group in which the touch sensing frame is established. In single group systems, these two groups will have a value of "1" and "1" respectively (the default values).

In the case of multi-group systems using touch sensing with coordinated motion, the reference group will be the leader group of your coordinated motion CD_pair, and the robot group will be the follower group of your CD_pair. A valid CD_pair calibration is required. This causes the offset calculated by the search routine to be calculated with respect to the coordinated motion frame. Refer to the *Coordinated Motion Setup and Operations Manual* for more details regarding coordinated motion setup and calibration.

In the case of multi-robot multi-group systems, touch sensing is supported on each robot, and group1 and group2 are supported. Standard touch sensing for group 1 will have a value of "1" and "1" respectively (the default values). Standard touch sensing for group 2 robot will have a value of "2" and "2" respectively. Coordinated motion touch sensing will have a positioner group and a robot group (for example, Ref Grp = 3, Robot Grp = 1).

8. Define the origin point of the Touch Frame.
   a. Move the cursor to Origin.
   b. Jog the Robot TCP to the desired starting point (origin).
   c. Press F2, RECORD.
9. Define the +X direction.
   a. Move the cursor to X.
   b. Jog the robot TCP to a point along the +X axis of the touch frame.
   c. Press F2, Record.
10. Define the +Y direction.
   a. Move the cursor to Y.
   b. Jog the robot in the +Y direction of the touch frame, to a point on the X-Y plane.
   c. Press F2, RECORD.
11. Press F5, DONE, to complete the definition of the frame.

Procedure 18-3 Setting Up a Touch Frame Using the Direct Entry Method

Steps

1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Touch Frame. You will see a screen similar to the following.

```
Touch Frame Setup
  Direct Entry:
    Rotate about X :      11.732
    Rotate about Y :       2.274
    Rotate about Z :     -20.195
  Teach Method:
    Origin         :      USED
    +X direction   :      USED
    +Y direction   :      USED
```

5. Move the cursor to Frame Number. Type the number of the frame to define and press ENTER.
6. Move the cursor to Reference Grp. Type the number of the reference group and press ENTER.
7. Move to Robot Grp and select the appropriate group number.

Note If you change the value of Reference Grp for an initialized frame, the following warning message will be displayed:

```
Frame data will be cleared!
                Yes     No
```

If you press F3, Yes, the frame data will be reinitialized.

If you set Reference Grp > 1, but no leader group matches the selected reference group, or it has not been calibrated for coordinated motion, the value of Reference Grp will not change and the following warning message will be displayed:

```
Referenced group does not exist
```

8. Define the rotation angle about X.
   a. Move the cursor to Rotate about X.
   b. Enter the value (in degrees).
9. Define the rotation angle about Y.
   a. Move the cursor to Rotate about Y.
   b. Enter the value (in degrees).
10. Define the rotation angle about Z.
   a. Move the cursor to Rotate about Z.

b. Enter the value (in degrees).

11. Press F5, DONE, to complete the definition of the frame.

18.3.3 Search Pattern

Search patterns determine the kind of information stored in the position register. The stored information is either the found position, or the position offset information depending on the search pattern used and the reference group specified in the touch schedule. Four types of search patterns are available:

- Simple search
- Fillet/lap search
- V-Groove search
- Outside/inside diameter search

Note You select the type of search pattern that is used when you set up the touch sensing schedule. See Section 18.3.4 .

Simple Search

For a **simple search** , a two-dimensional search is executed to find the actual location of one position on an object. **A simple search stores the found position (x, y, z, w, p, r) into a position register PR[].** Once completed, the robot is programmed to move to the position stored in that position register.



Caution

Do not use simple search when you use the multi-pass option with touch sensing because both the simple search and multi-pass features use position registers. Simple search stores the computed position in a position register. Multi-pass cannot use position registers to plan paths. Use the 2D fillet search pattern when using multipass with touch sensing.

Simple search requires:

- That the surfaces being searched are perpendicular to each other.
- Searches to be done in two different directions.
- The second search motion to be performed with the desired torch angle.

The first search defines the positional information for that search direction only (x, for example). The second search defines the other direction positional information (z, for example). The starting position of the second search defines the remaining positional information, (y, w, p, r, for example) that determines the torch angle for welding and, in this case, the y value.

Simple search is typically used to find the starting point of a weld path that uses the Thru-Arc Seam Tracking (TAST) option.

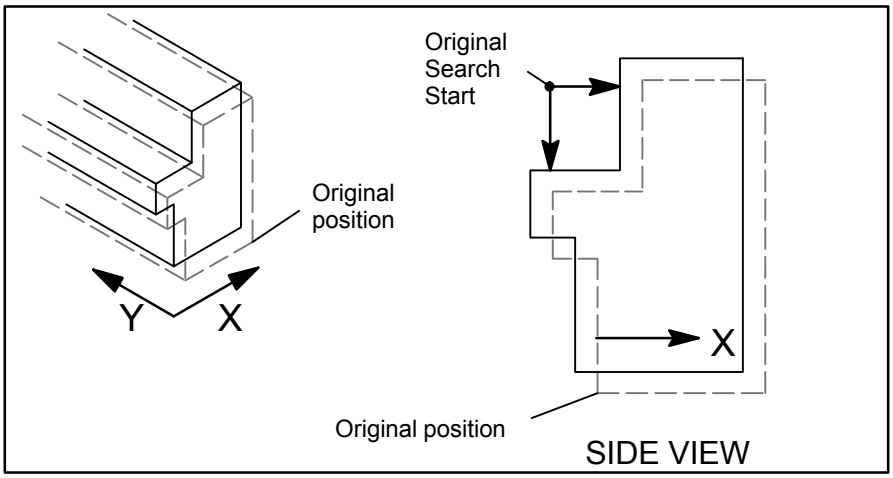
A two-dimensional search is programmed in the software as the **only** valid search pattern type when a simple search is used. Changing the search pattern type has no effect.

The two-dimensional search that Simple Search does is called a *pattern type.* The two-dimensional search is the **only** valid pattern type for a simple search.

Refer to Table 18–2 for information on search patterns and valid pattern types for each search pattern.

Refer to Section 18.3.4 for example programs using simple search. See Figure 18–6 for an illustration of a simple search routine.

Figure 18–6. Simple Search Routine Using Searches in Two Directions



Fillet/Lap Search

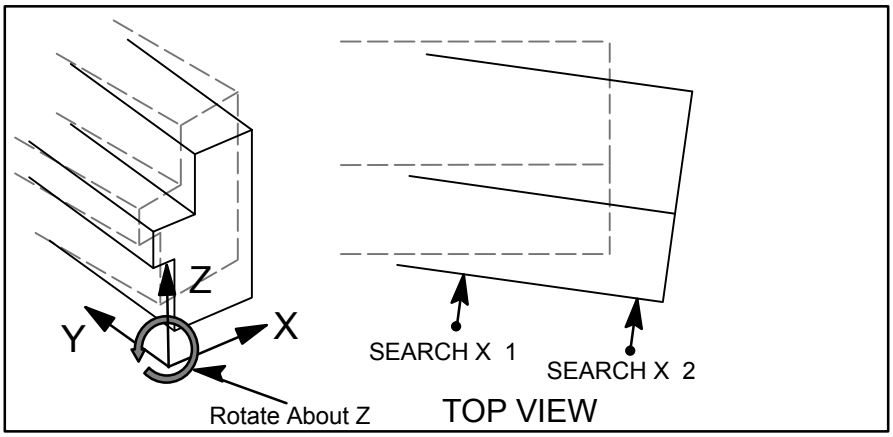
For a **Fillet/Lap Search** a one, two, or three dimensional search is executed to obtain positional offset information. **A Fillet/Lap Search stores positional offset information in a position register PR[ ]** . This offset can be applied to one or more positions in a programmed path.

The offset can be in one, two, or three directions. The offset can also be in two directions plus rotation about the axis of which no searching is performed. For example, if the object is being searched for offset in both x and y directions, a fillet search can offset for a rotation about the z axis.

Another type of offset can be in one direction plus rotation about an axis of which no searching is performed. For example, if the object is being searched for offset in x, a fillet search can offset for a

rotation about z. Note that is this type of search, the first touch point is used as the arc start point. See Figure 18–7 .

Figure 18–7. Fillet Search in One Direction (x) with Rotation about Z



Another type of offset can be in three directions plus rotation about the axis of which no searching is performed. For example, if the object is being searched for offset in x, y, and z directions, a fillet search can offset for a rotation about the z axis. In Figure 18–9 , three searches are performed in the z direction. These searches define the single plane of rotation in the 3D + Rotate pattern. These three searches must be in the same plane and be co-planer to the welding plane.

A fillet search stores an offset into a position register [PR]. The robot program then uses the touch offset commands to begin and end the offset.

The type of searches that a Fillet/Lap Search does is called a *pattern type.* See Figure 18–8 for information on search patterns and valid pattern types for each search pattern.

Refer to Section 18.3.4 for example programs using Fillet/Lap Search. See Figure 18–8 and Figure 18–9 for illustrations of Fillet/Lap Searches.

Figure 18–8. Fillet Search in Two Directions (x and y) with Rotation about Z

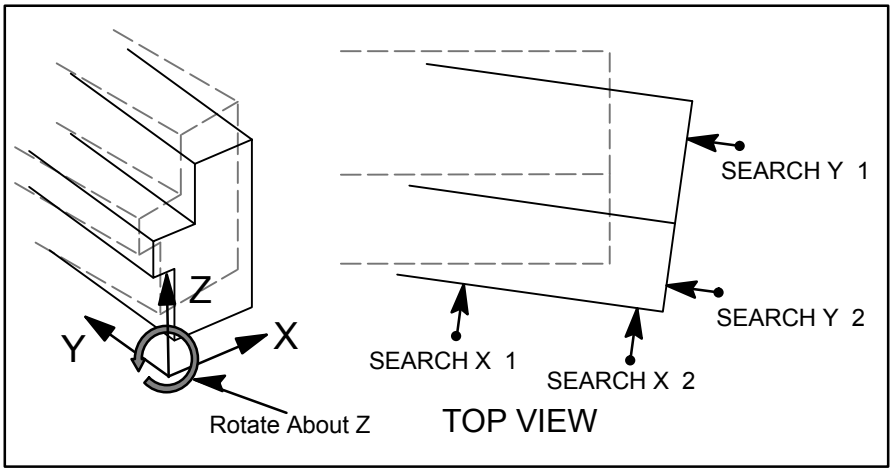
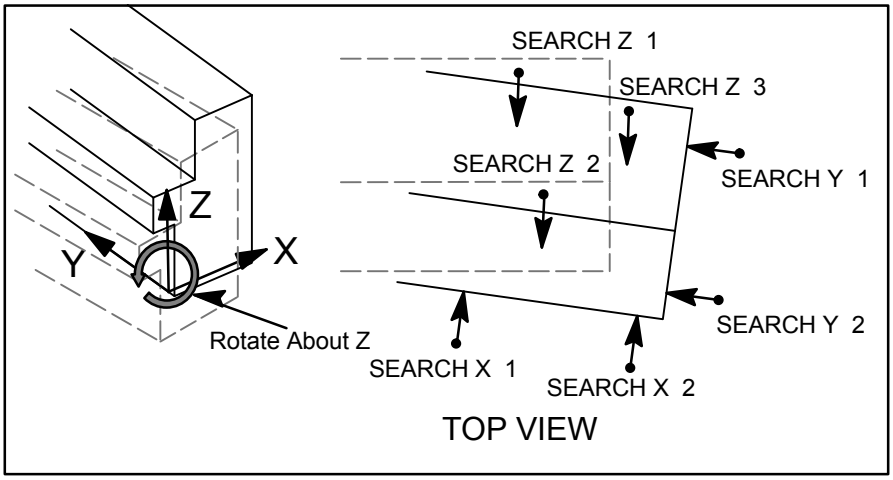


Figure 18–9. Fillet Search in Three Directions (x, y, z) with Rotation about Z



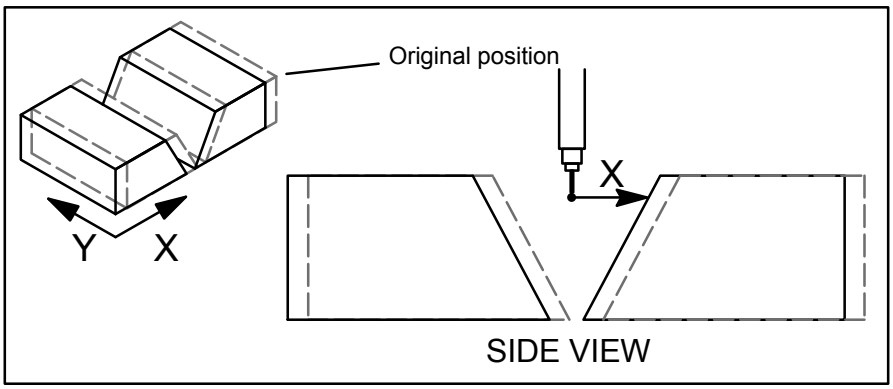
V-Groove Search

For **V-Groove Search** a one-dimensional search is executed to obtain positional offset information. **A V-Groove Search stores positional offset information in a position register [PR**]. This offset can be applied to one or more positions in a programmed path.

The types of searches that a V-Groove Search does is called a *pattern type.* Refer to Table 18–2 for information on search patterns and valid pattern types for each search pattern.

Refer to Section 18.3.4 for example programs using V-Groove Search. See Figure 18–10 for an illustration of a V-Groove Search.

Figure 18–10. V-Groove Search



Outside/Inside Diameter Search (OD/ID)

For **Outside/Inside Diameter Search** (OD/ID Search) a two dimensional search is executed to obtain the positional offset information of the center point of a circular path relative to the original (master) location. **An Outside/Inside Diameter Search stores positional offset information in a position register [PR**]. This offset can be applied to one or more positions in a programmed path.

The types of searches that an Outside/Inside Search does is called a *pattern type.* Refer to Table 18–2 for information on search patterns and valid pattern types for each search pattern.

Refer to Section 18.3.4 for example programs using OD/ID Search. See Figure 18–11 for an illustration of a OD/ID Search.

Figure 18–11. OD/ID Search in Two Directions (X and Y)

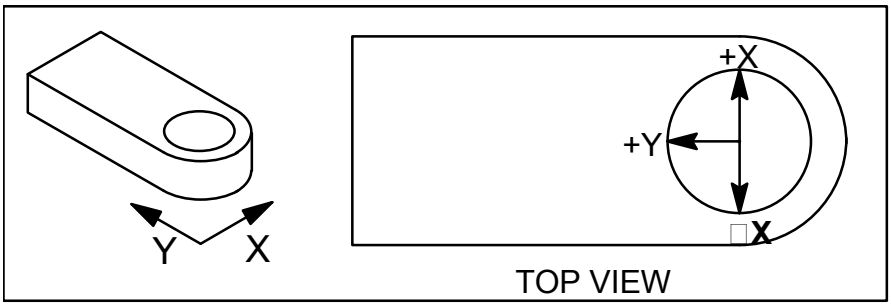


Table 18–2 shows a matrix of possible search pattern and valid pattern types. Select a combination that you would like to use on your application and verify that it will provide the proper results.

Table 18–2. Search Pattern and Valid Pattern Type

Search Patterns	Pattern Type 1_D	Pattern Type 2_D	Pattern Type 3_D	Pattern Type 1_D and Rotation	Pattern Type 2_D and Rotation	Pattern Type3_D and Rotation
Simple Search	Not Valid	Requires 2 different search directions. Minimum 1 search per direction.	Not Valid	Not Valid	Not Valid	Not Valid
Fillet/Lap	Requires 1 search direction. Minimum 1 search per direction.	Requires 2 different search directions, x and y, x and z, y and z. Minimum 1 search per direction.	Requires 3 different search directions, x,y, and z. Minimum 1 search per direction.	Requires 1 search direction. Minimum 2 searches per direction.	Requires 2 different search directions. Minimum 2 searches per direction.	Requires 3 different search directions. 3 searches in one direction (usually -z)2 searches in each of the remaining directions.
V-Groove	Requires 1 search direction. Minimum 1 search per direction.	Not Valid	Not Valid	Not Valid	Not Valid	Not Valid
OD/ID	Not Valid	Requires 3 different searches in 2 different directions. For example, +x,-x,+y, NOT x,y,z. Minimum 1 search per direction.	Not Valid	Not Valid	Not Valid	Not Valid

18.3.4 Touch Schedule

A touch schedule is a series of conditions that control how the search motion is completed. Thirty-two (32) touch schedules are available.

You access touch schedules from the DATA menu. There are two screens associated with touch schedules: the SCHEDULE screen and the DETAIL screen.

The SCHEDULE screen allows you to view and set limited information for nine schedules at one time. DETAIL allows you to view and set the complete information for a single schedule. You can display the SCHEDULE screen by pressing PREV. You can display the DETAIL screen by pressing the function key F2, DETAIL.

Table 18–3 lists and describes each schedule setup item.

Use Procedure 18-4 to define touch schedules. Refer to Section 18.3.3 for a description of search patterns.

Table 18–3. Touch Sensing Schedule Setup Items

ITEM	DESCRIPTION
Touch Schedule	This item indicates the number of the displayed schedule. A comment can be entered.
Master Flag Default = OFF	This item enables the search routine to be used as a mastering routine for those touch sensing programs that generate position offset information. If set to ON, when the search routine is executed, the touched positions are recorded as the reference positions to be used by future searches. This flag must be set to OFF after the master search is completed in order to generate position offset information on the objects to be searched. Also, when the search is performed, the position offset information in the position register is set to all zero values. This means the when touch sensing finds the object in its master position, no offset is to be applied to the weld path. **Note** The Master Flag condition has no effect on simple searches.
Search Speed Default = 50.0 mm/sec	This item specifies how fast the robot will move when performing a Search Motion. **Caution** A search motion is programmed as a motion option at the end of a position instruction. The speed at which the robot will move is determined by the search speed, not by what is indicated in the position instruction. During testing, when dry run is in effect, this search speed is also used. The dry run speed has no effect.
Search Distance Default = 100 mm	This item defines how far the robot can move when it is performing a search. Error code *THSR-017 Pause No contact with part.* is displayed when this distance is reached without making contact with the object.

Table 18–3. Touch Sensing Schedule Setup Items (Cont'd)

ITEM	DESCRIPTION
Touch Frame Default = 1	This item defines the touch frame to be used in the touch schedule. This determines the x, y, and z directions for the search motion. The same touch frame can be used in more than one touch schedule. When using multi-arm simultaneous touch sensing, each touch process robot can define its own touch frame within the same touch frame number.
Search Patterns Default = SIMPLE	This item defines the type of object to be searched and causes the software to compute the found position or positional offset information dependent on the search pattern selected. The computed data is stored in a position register. There are four available search patterns: • Simple Search • Fillet Search • V-Groove Search • OD/I D Search
Pattern Type Default = 1_D Shift **1_D Shift** **2_D Shift** **3_D Shift** **1_D Offset** **2_D Offset** **3_D Offset**	This item selects the type of offset to be stored in the position register. Six pattern types are available: Stores a one dimensional offset. Offsets can be in the x, y, or z direction. Stores a two dimensional offset. Offsets can be in two of the x, y, or z direction. Stores a three dimensional offset to a program. Offsets are in the x, y, or z direction. Stores a one dimensional offset with rotation about the axis of which the search is not performed. Stores a two dimensional offset with rotation about the axis of which no searches are performed. For example, if the object is being searched for an offset in both the x and y directions, a 2_D Shift & Rotate search can offset for a rotation about the z axis. Stores a three dimensional offset with rotation about the axis of which no searches are performed. For example, if the object is being searched for an offset in both the x and y directions, a 3_D Shift & Rotate search can offset for a rotation about the z axis. **Note** Simple, OD/ID, and V-Groove search patterns are pre-defined. Changing the pattern type for these searches has no effect.

Table 18–3. Touch Sensing Schedule Setup Items (Cont'd)

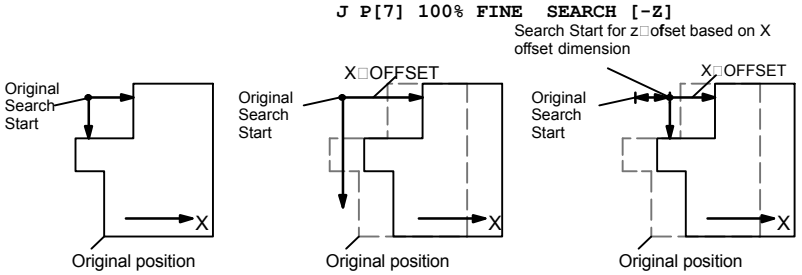
ITEM	DESCRIPTION
Incremental Search Default = ON NOTE: Simple search does not support incremental search	Offsets the starting position of the second etc. search in a search routine by the amount of offset found by the first search motion. If set to OFF, the robot returns to the original starting position. The following illustration shows how the incremental search affects the search routine. Incremental search requires a separate SEARCH START point for each search. **Program Example:** **J P[4] 100% FINE** **J P[5] 100% FINE SEARCH [-X]** **J P[6] 100% FINE** **J P[7] 100% FINE SEARCH [-Z]**  Without incremental search, the robot found the x-offset but cannot find the z□offset.
Auto Return Default = ON	This item moves the robot back to the search start position when contact is made with the object. If set to OFF, the robot stops at the contact point and moves straight to the next position.
Return Speed Default = 100 mm/sec	This item specifies the speed at which the robot will return to the search start position upon making contact with the part.
Return Distance Default = 2000 mm Minimum = 0 mm Maximum = 2000 mm	When Auto Return is set to ON, Return Distance specifies the distance the robot will return automatically. If the return distance passes the initial search start position, the robot will return to the initial start position.

Table 18–3. Touch Sensing Schedule Setup Items (Cont'd)

ITEM	DESCRIPTION
Reference Group	This item specifies how the offset is recorded: • **0** : There is no reference group and no leader group. OFFSET is recorded with respect to the robot itself. • **1** : OFFSET is recorded with respect to the UFRAME of robot group 1. • **2** : OFFSET is recorded with respect to the coordinated frame of robot group 2 (leader) **Caution** Searches and offsets for coordinated motion cannot be mixed with non-coordinated. That is, you cannot do searches for coordinated motion, then do searches for non-coordinated motion, and then use the offset from the coordinated motion search. You must use the offsets for coordinated motion searches before doing any non-coordinated searches. Similarly, you must use any offsets from non-coordinated motion searches before doing any coordinated motion searches. • **3** : OFFSET is recorded with respect to the coordinated frame of robot group 3 (leader). • **4** : OFFSET is recorded with respect to the coordinated frame of robot group 4 (leader). • **5** : OFFSET is recorded with repsect to the coordinated frame of robot group 5 (leader). **NOTE:** For searches other than simple search, Reference Group must equal the frame Reference Group. Otherwise, an error message, "Reference grp mismatch," will be displayed. **For simple search** , Reference Group must be 1. Otherwise an error message, "Illegal motion ref. grp," will be displayed.
Return Term Type Default = Fine	This item specifies the termination type the robot will use to return to the search start position. Five Return Term Types are available: • FINE • CNT20 • CNT40 • CNT100

Table 18–3. Touch Sensing Schedule Setup Items (Cont'd)

ITEM	DESCRIPTION
Contact Record PR Default = 32	The search output position register is used as a temporary buffer to hold the last search contact position. The purpose for this temporary position register buffer is to provide the ability to look at the positional data of an individual search, or to extract data from the buffer in a program. By default, this register is position register 32. The search output position register should be assigned to the last position register number in your system. **Warning** **The data position register is overwritten at each motion so the same position register should not be used to store the final positional data from the search motion. Also, the contents of this temporary buffer is a real position, not an offset. Do not program motion instructions to use this position register data as an offset.**
Error on Failure Default = ON	This item posts error code *THSR - 017(PAUSE) No contact with part* , if the search move exceeds the distance set in Search Distance. When OFF, the program execution continues with the next instruction if the Search Distance is exceeded; **Programming Hint:** If this is set to OFF, the next instruction in the program looks at the contents of the Error Register and branches accordingly.
Error Register Number Default = 32	When Error On Failure is set to OFF, this register is set to 1 when the search distance is exceeded. A successful search sets this register to 0.
Robot Group Mask	This item specifies the motion group performing the search move and to which the offset is applied. When using DualARM or Multi-Robot systems this can be group 1 or 2. For single group systems this will always be 1. For coordinated motion Touch Sensing, the leader and follower groups must be included in the touch frame you select. Robot group mask together with reference group should be equal to TP program group mask that uses this touch schedule, otherwise an error "Group number mismatch" will be posted. When using multi-arm simultaneous touch sensing, all touch process robots used by this schedule should be included in this item.

Procedure 18-4 Defining Touch Schedules

Steps

1. Press DATA.
2. Press F1, [TYPE].
3. Select Touch Sched. You will see a screen similar to the following.

```
DATA Touch Sched 1/32
  (mm/sec)   (mm)  FRAME  MASTER  GMASK
1   50.0     100.0   1     OFF      1
2   50.0     100.0   1     OFF      1
3   50.0     100.0   1     OFF      1
4   50.0     100.0   1     OFF      1
5   50.0     100.0   1     OFF      1
6   50.0     100.0   1     OFF      1
7   50.0     100.0   1     OFF      1
8   50.0     100.0   1     OFF      1
9   50.0     100.0   1     OFF      1
```

4. Move the cursor to the desired schedule number.
5. **To display more information about the schedule**, press F2, DETAIL. See the following screen for an example.

```
DATA Touch Sched 1/17
 1 Touch Schedule:8    [   Touch     ]
 2  Master flag:           OFF
 3  Search speed           50.0  mm/sec
 4  Search distance        100.0 mm
 5  Touch frame            2
 6  Search pattern         Simple
 7  Pattern Type           2_D Shift
 8  Incremental search:    ON
 9  Auto return:           ON
10 Return speed           100.0  mm/sec
11 Return distance:       2000.0 mm
12 Reference Group:       1
13 Return term type:      Fine
14 Contact Record PR:     31
15 Error on failure:      ON
16 Error register num:    32
17 Robot Group:           [1,*,*,*,*,*,*,*]
```

6. Set each schedule item as desired.
7. **To add a comment:**
   a. Move the cursor to the to the comment line and press ENTER.
   b. Select a method of naming the comment.
   c. Press the appropriate function keys to add the comment.

d. When you are finished, press ENTER.

8. To copy schedule information from one schedule to another:

a. Press NEXT, >.

b. Move the cursor to the schedule you want to copy.

c. Press F2, COPY.

d. Type the number of the schedule to which you want to copy the data.

e. Press ENTER. The data will be copied, but the comment will not be copied.

9. To clear the information you have entered for a schedule:

a. Move the cursor to the schedule.

b. Press NEXT, >.

c. Press F2, CLEAR. The data will be cleared, but the comment will not be cleared.

d. To continue, press F4, YES, otherwise, press F5, NO.

18.4 TOUCH SENSING PROGRAMMING

18.4.1 Programming Overview

A touch sensing routine consists of search instructions to locate an object, and offset instructions to displace programmed positions.

Note Any changes to the tool frame affects the touch start position.



Caution

Recorded positions and position registers are affected by UFRAME, and UFRAME has an effect during playback. If you change UFRAME, any recorded positions and position registers will also change.

18.4.2 Touch Sensing Instructions

Touch sensing instructions are used to implement touch sensing programming. Four touch sensing instructions are provided:

- Search Start

- Search End
- Touch Offset
- Touch Offset End

Note See Section 6.29 for detailed information about the touch sensing instructions.

18.4.3 Touch Sensing Motion Option

Search [ ] Motion Option

J P[1] 50% Fine Search [ ]

There is one Touch Sensing motion option: Search [ ]..The Search [ ] motion option directs the motion of the robot (in a positive or negative x,y or z direction) to search for the object. The x, y, and z vectors are defined by the touch frame assigned in the touch schedule. When contact is made with the object, the robot's current TCP position is stored and robot motion is stopped.

The Search [ ] motion option is entered at the end of a motion instruction.

Note Search and Search Start must use FINE termination type.

The recorded position that has the search motion option is not executed, so motion to the search start position must be recorded in a separate motion instruction. See Figure 18–12 .

Figure 18–12. Touch Sensing Motion Option Example

INSTRUCTION	DESCRIPTION
`J P[3] 100% FINE`	Move to search start position
`J P[3] 20% FINE SEARCH [-X]`	Search motion

Warning

Motion speed and direction are controlled by values set in the touch schedule assigned by the Search Start instruction, not by the motion instruction associated with that line of the program. The motion and speed could be different than what is displayed on the motion instruction.

Use Procedure 18-5 to add a Search[ ] instruction.

Procedure 18-5 Adding a Search [ ] Instruction to a Program

Note Refer to Chapter 5 *PLANNING AND CREATING A PROGRAM* for details on creating and modifying a program.

Steps

1. Jog the robot to the search start position and record the position.
2. Record another position at the same location. This second motion instruction will be controlled by the touch sensing software during the search.
3. Move the cursor to the end of the motion instruction line of the selected position.
4. Press F4, [CHOICE], to view the motion option choices.
5. Select 8, Next Page
6. Select Search and press ENTER.
7. Select the direction of the search to be performed and press ENTER.

18.4.4 Motion Instructions Used with Touch Sensing

Touch sensing routines, using a simple search, apply the positional data by using a motion instruction. A simple search stores an actual position in the specified position register.

After a "simple" search routine, the touch sense software will calculate a *real* position (x,y,z,w,p,r) and put the data in the position register defined by the SEARCH START[1] PR[x] instruction. Since this is a real position, the robot will be commanded to move to the position in the position register instead of to a recorded position.

```
Example:  J  PR [4]  100% FINE   ARC START [1]


J PR[4] 100% FINE  ARC START[1] shows where position register
4 is
the position register specified in the simple search routine.
```

18.4.5 Executing a Touch Sensing Program

When executing a touch sensing program, all testing and cautions must be followed. Refer to Chapter 7 *TESTING A PROGRAM AND RUNNING PRODUCTION* for more information about testing programs and running production.

For Fillet/Lap, V-Groove, OD/ID search pattern programs you must establish master positions for all search motion by:

1. Setting the master flag in the touch schedule that is specified in the SEARCH START command used to ON.
2. Running the program to establish master positions for all search motions.
3. Setting the master flag in the touch schedule that is specified in the SEARCH START command to OFF.

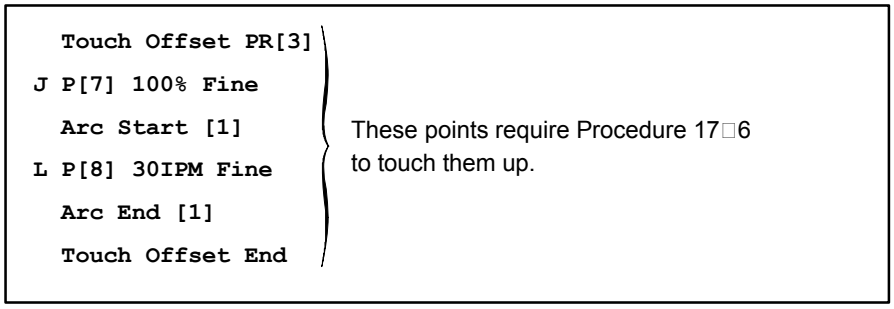
Refer to Chapter 5 *PLANNING AND CREATING A PROGRAM* for details on creating and modifying a program.

18.4.6 Touch Sensing Robot Position Touchup

You can use the function key F5, TOUCHUP, when editing your program to modify the recorded robot position. When you use the TOUCHUP function with touch sensing, the new positional information is added to the offset information to determine the weld path. Use Procedure 18-6 to touch up robot positions in a touch sensing program.

Figure 18–13 shows an example of points that require touching up. Refer also to Section 18.5.2 and Section 18.5.3 .

Figure 18–13. Points that Require Touching Up



In order to correctly touch up Touch Offset positions, follow Procedure 18-6 .

Procedure 18-6 Touching Up Robot Positions in a Touch Sensing Program

Steps

1. Execute the program so that the search data is complete and the position register contains the offset information.
2. Execute the line of your program that contains the Touch Offset instruction.

Caution

Do not execute a Touch Offset End instruction and then use backward execution to move to the program line that contains the robot position you want to touchup. Otherwise, the offset data will be incorrect.

3. Single step to a line in the program that contains the first robot position that you want to touch up.
4. Jog the robot to the new position, press and hold in the SHIFT key and press F5, TOUCHUP.
5. Touch up all necessary robot positions between the Touch Offset Start and Offset End positions.

18.4.7 Programming Examples

Example programs contained in this section include:

- Simple search - Figure 18–14
- One-dimensional search (Fillet/Lap, V-Groove) - Figure 18–15
- Two-dimensional with rotation - Figure 18–16
- Two-dimensional with coordinated motion - Figure 18–17 , Figure 18–18 , and Figure 18–19
- Simple search with coordinated motion - Figure 18–20

Note Do not use a continuous term type (CNT) for motion that is right before a Search. Instead, use the FINE term type. See line 3 in the Simple Search Example Program. If you use continuous, the search cannot compute a valid offset.

Figure 18–14. Simple Search Example Program

INSTRUCTION	DESCRIPTION
`1: J P[1] 100% Fine`	Teach a point in space.
`2: Search Start [4] PR[4]`	Search uses schedule 4, position register 4, to store position
`3: J P[2] 100% Fine`	Teach a search starting position.
`4: J P[3] 100% Fine Search [Y]`	Do a search motion in the Y direction.
`5: J P[4] 100% Fine Search [□Z]`	Do a search in the □Z direction.
`6: Search End`	End of the search.
`7: J PR[4] 100% Fine`	Move the robot to to the computed position PR[4].
`8: ARC START[1]`	

Note Simple search is different from all other searches in two aspects: First, the master flag in the schedule is always set to off. Second, the position register contains an absolute position instead of an offset.

Figure 18–15. One-Dimensional Search Ex. Prog. (Fillet/Lap, V-Groove)

INSTRUCTION	DESCRIPTION
1: J P[1] 100% Fine	Teach a point in space
2: Search Start [1] PR[1]	Search uses schedule 1, register 1 stores Offset
3: J P[2] 100% Fine	Teach a search starting position.
4: J P[3] 100% Fine Search [Y]	Do a search motion in the Y direction.
5: Search End	End of the search
6: J P[4] 100% Fine	Teach an intermediate point (optional)
7: Touch Offset PR[1]	The following positions will be offset by PR [1].
8: J P[5] 100% Fine	P[5] is offset by PR[1].
9: ARC START[1]	
10: L P[6] 30IPM Fine	P[6] is offset by PR[1].
11: ARC END[1]	
12: Touch Offset End	End of offsetting position.

Figure 18–16. Two Dimensional Search Example Program

INSTRUCTION	DESCRIPTION
1: J P[1] 100% Fine	Teach a point in space (optional).
2: Search Start [2] PR[2]	Search uses schedule 2 position, register 2 stores Offset
3: J P[2] 100% Fine	Teach a search starting position.
4: J P[3] 100% Fine Search [Y]	Do a search motion in the Y direction.
5: J P[4] 100% Fine	Go to another search start position
6: J P[5] 100% Fine Search [X]	Do a search in the X direction
7: Search End	End of the search.
8: J P[6] 100% Fine	Teach an intermediate point (optional)
9: Touch Offset PR[2]	The following positions will be offset by PR[2]
10: J P[7] 100 Fine	P[7] is offset by PR[2].
11: ARC START[1]	
12: L P[8] 30IPM Fine	P[8] is offset by PR[2].
13: ARC END[1]	
14: Touch Offset End	End of offsetting position.

Caution

Searches and offsets for coordinated motion cannot be mixed with non-coordinated motion. That is, you cannot do searches for coordinated motion, then do searches for non-coordinated motion, and then use the offset from the coordinated motion search. You must use the offsets for coordinated motion searches before doing any non-coordinated searches. Similarly, you must use any offsets from non-coordinated motion searches before doing any coordinated motion searches.

Figure 18–17. Two Dimensional Search with Coordinated Motion Example Program (See Figure 18–18 and Figure 18–19 for illustrations)

INSTRUCTION	DESCRIPTION
`1:  J P[1] 20% FINE`	Teach a home position (follower/leader)
`2:    Search Start[2] PR[2]`	Search uses schedule 2, register 2 to store offset
`3:  J P[2] 100% FINE`	Teach a search start position (follower/leader)
`4:  J P[3] 100% Search[X]`	Do a search in X-direction relative to part
`5:  J P[4] 100% FINE`	Go to another search start position
`6:  J P[5] 100% FINE Search[X]`	Do a search in the X-direction relative to part
`7:  J P[6] 100% CNT100`	Teach an intermediate point
`8:  J P[7] 100% FINE`	Go to another search start position
`9:  J P[8] 100% FINE Search[Y]`	Do a search in Y-direction relative to part
`10: J P[9] 100% FINE`	Go to another search start position
`11: J P[10] 100% FINE Search[Y]`	Do a search in Y-direction relative to part
`12:   Search End`	End of search
`13: J P[1] 100% FINE`	Go to home position
`14:   Touch Offset PR[2]`	The following positions will be offset by PR[2]
`15: J P[11] 100% FINE`	Go to starting position
`16:   ARC START[1]`	
`17: L P[12] 30mm/sec FINE COORD`	P[12] is offset by PR[2]
`18: L P[13] 30mm/sec FINE COORD`	P[13] is offset by PR[2]
`19:   ARC END[1]`	
`20:   Touch Offset End`	End of offsetting position
`21: J P[1] 100% FINE`	Go to home position

NOTE:
- The search direction is part relative as shown in Figure 17□18. When the part moves, the search direction does not change.
- Offset PR[2] is part relative as shown in Figure 17□19.
- Motions between searches are allowed.

Figure 18–18. First Illustration of Two Dimensional Search with Coordinated Motion Program Example

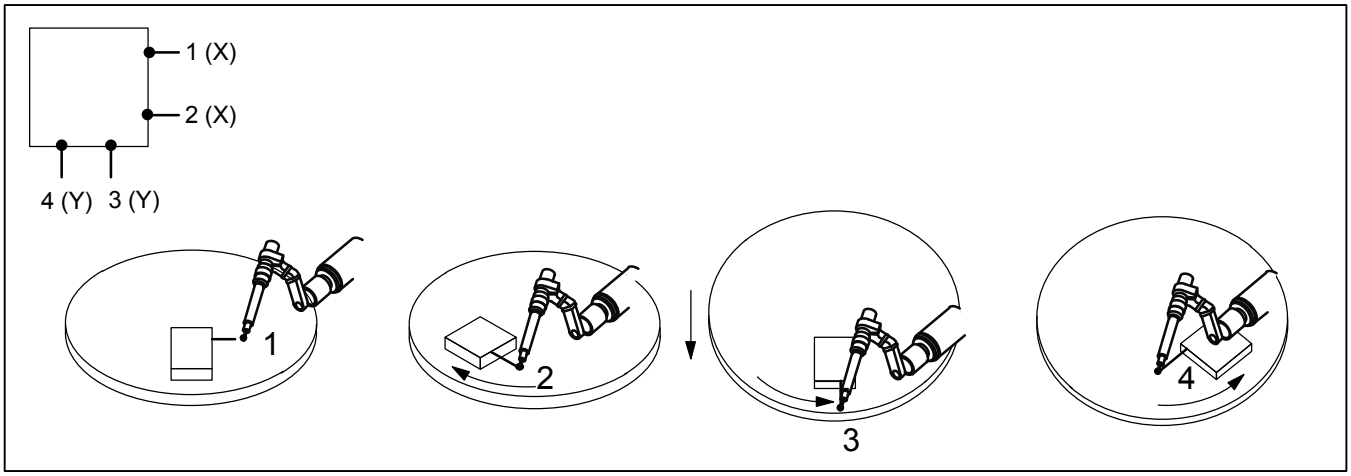


Figure 18–19. Second Illustration of Two Dimensional Search with Coordinated Motion Program Example

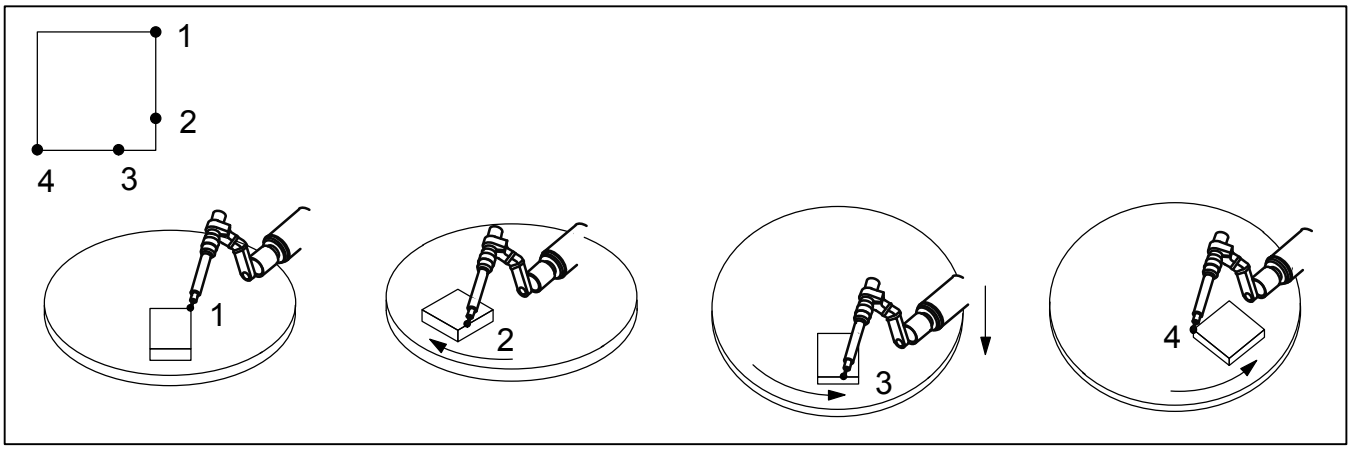


Figure 18–20. Simple Search with Coordinated Motion Example Program

INSTRUCTION	DESCRIPTION
1: J P[1] 100% FINE	Teach a home position (follower/leader)
2: Search Start[2] PR[2]	Search uses shcedule 2, register 2 to store position
3: J P[2] 100% FINE	Teach a search start position (follower/leader)
4: J P[3] 100% Search[X]	Do a search in X-direction
5: J P[4] 100% FINE	Go to another search start position, leader can□t move
6: J P[5] 100% FINE Search[Y]	Do a search in Y-direction
7: Search End	End of search
8: J P[6] 100% FINE	Intermediate position
9: J PR[2] 100% FINE	Move the robot to PR[2]
10: ARC START[1]	Begin welding
11: L P[4] 30IPM FINE COORD	Coordinated motion
12: ARC END[1]	

NOTE:
- The simple search frame can be relative to the follower or to the leader group.
- The stored position is relative to the follower.
- The leader is not allowed to move between the searches.

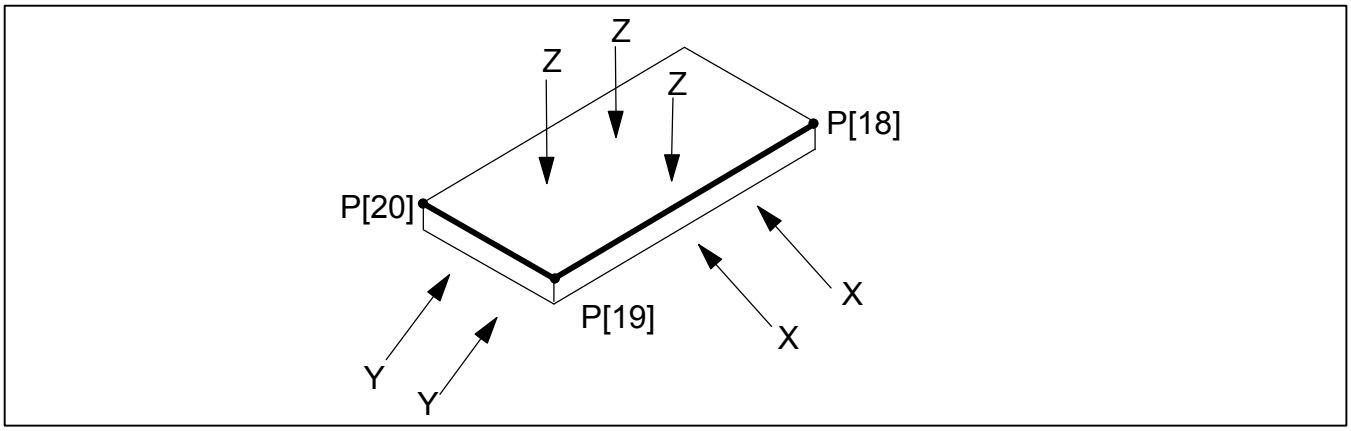
Three Dimensional Search Example Program

The 3D search is very similar to the 2D search. To do a 3D search, add searches in the z-direction.

Figure 18–21. Three Dimensional Search with Rotation Example Program (See Figure 18–22 for an illustration)

INSTRUCTION	DESCRIPTION
`1: J P[1] 100% Fine`	Teach a home position
`2: Search Start [3] PR[3]`	Search uses schedule 3, register 3 to store offset
`3: J P[2] 100% Fine`	Teach a search start position
`4: J P[3] 100% FINE Search[X]`	Do a search in X-direction
`5: J P[4] 100% FINE`	Go to another search start position
`6: J P[5] 100% FINE Search[X]`	Do a search in the X-direction
`7: J P[6] 100% CNT100`	Teach an intermediate point
`8: J P[7] 100% FINE`	Go to another search start position
`9: J P[8] 100% FINE Search[Y]`	Do a search in Y-direction
`10: J P[9] 100% FINE`	Go to another search start position
`11: J P[10] 100% FINE Search[Y]`	Do a search in Y-direction
`12: J P[11] 100% CNT100`	Teach an intermediate point
`13: J P[12] 100% FINE`	Go to another search start position
`14: J P[13] 100% FINE Search[Z]`	Do a search in Z-direction
`15: J P[14] 100% FINE`	Go to another search start position
`16: J [15] 100% FINE Search[Z]`	Do a search in Z-direction
`17: J P[16] 100% FINE`	Go to another search start position
`18: J P[17] 100% FINE Search[Z]`	Do a search in Z-direction
`19: Search End`	End of search
`20: J P[1] 100% FINE`	Go to home position
`21: Touch Offset PR[3]`	The following positions will be offset by PR[3]
`22: J P[18] 100% FINE`	Go to starting position
`23: ARC START[1]`	
`24: L P[19] 30IPM FINE`	P[19] is offset by PR[3]
`25: L P[20] 30IPM FINE`	P[20] is offset by PR[3]
`26: ARC END[1]`	
`27: Touch Offset End`	End of offsetting position

Figure 18–22. Illustration of Three Dimensional Search with Rotation Program Example



18.5 TOUCH SENSING MASTERING

18.5.1 Mastering Overview

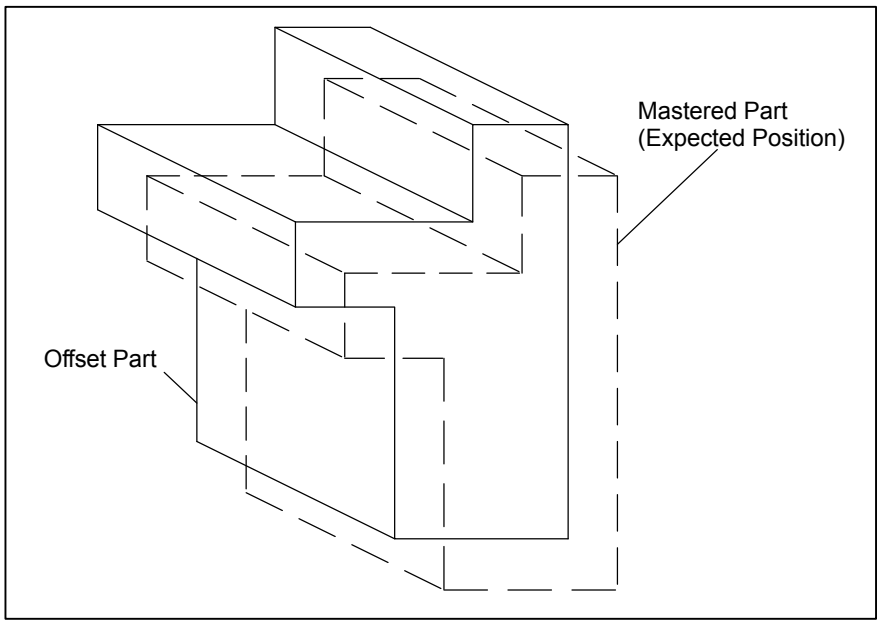
Touch sensing provides a method for determining part location and automatic adjustment of the robot path, to compensate for part displacement. This section contains the details of mastering a part for touch sensing with the following items:

- Mastering
- Remastering
- Offsets
- Patterns
- Master Flag
- Touching up Path Positions and Incorrect Touch Up
- Adding New Positions
- Multiple Searches
- Touching Up Search Start Positions

18.5.2 Mastering

Mastering refers to defining taught positions in a program as the expected locations of positions. When the robot follows the taught positions of the master path, then the offset is zero. An example is shown in Figure 18–23 .

Figure 18–23. Part in Mastered Position and Offset Applied Illustration



18.5.3 Remastering

The touchup procedure described in Section 18.5.10 should work for most instances where the search start positions do not need to be moved or if the parts do not change drastically.

Remastering is required if the search start positions must be retaught. Also, if the path must be altered significantly, it is recommended to remaster to ensure a correct path.

Remastering is accomplished by turning the Master Flag ON and running through the program. The path followed will be the master path with *no offset* applied. Points not in the correct location must be touched up. After executing the program, the Master Flag is turned OFF. For touching up path positions refer to Section 18.5.7 .

In addition, if the specific schedule reference group is not equal to 1 (follower), all of the mastering information is stored with respect to the reference group. If you change the reference group in a schedule, you will have to remaster.

Note Complex parts with multiple searches might only require remastering of specific portions of the path.

18.5.4 Offsets

Offsets generated by touch sensing are relative to the position found while mastering. An offset is computed by comparing the location of the part with the stored location. Figure 18–24 illustrates the offset value.

- The search performed during mastering establishes the expected location (which is indicated by the small straight line).
- This location is stored when mastering the part.
- An offset is computed by comparing the location of the part with this stored location.
- The offset is part relative when the schedule reference group is not equal to 1.

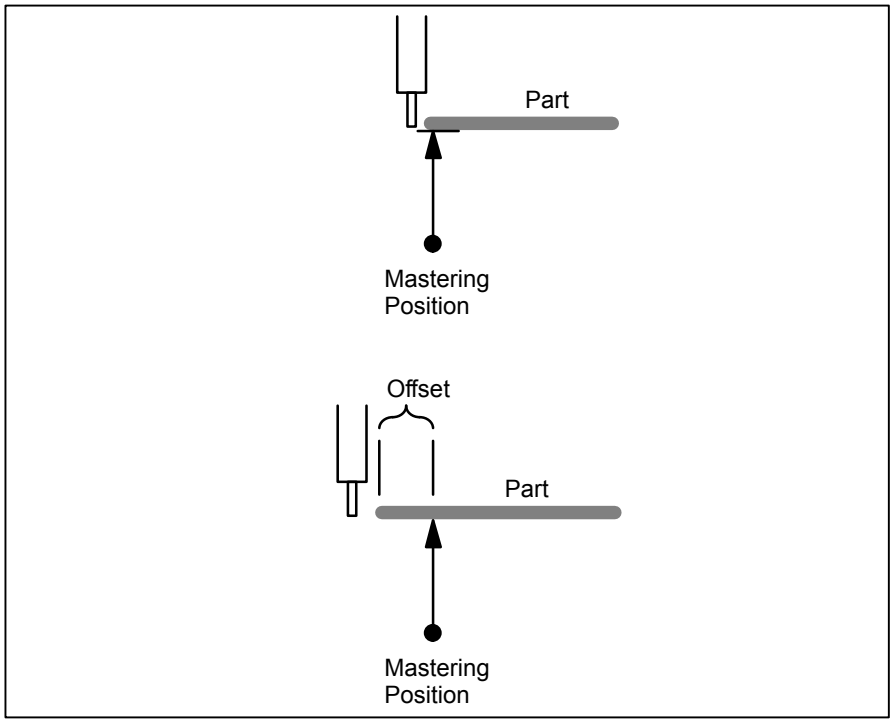
Caution

Do not mix the touch offset with a different touch schedule. If you do, an incorrect touch offset will be applied to the nominal path.

For example, a coordinated touch offset PR[1] is derived when using touch schedule 1 with reference to a leader group, and an absolute touch offset PR[2] is derived when using touch schedule 2 with reference to itself. An incorrect offset will be applied if the system uses schedule 2 to derive PR[2], and uses schedule 1 to derive PR[1], then applies PR[2] to a TP program without leader group and coordinated motion. An incorrect path is generated with PR[2], since the system uses the most recent schedule 1 with reference to a leader group.

The correct sequence is to use touch search to derive the offset, then immediately apply touch offset in the same TP program. Thus, it will force both Search Start and Touch Offset to use the same Touch Schedule with the same reference group.

Figure 18–24. Offset Value Illustration



18.5.5 Patterns

Mastering is needed for search patterns that generate offset data. The search patterns that require mastering are as follows:

- Fillet/Lap
- V-Groove
- Outside and Inside diameter searches

Note A simple search does not require mastering since it produces an actual location stored in a position register.

Program Example

The following program example describes a part with a search start location and three points along a straight path. Refer to Figure 18–25 and the program example screen shown below.

- The points are numbered according to the program example.
- The search is a two dimensional search, one in the x direction and the second in the -z direction.
- A 2_D Fillet/Lap search was performed.
- The type of search and other details are defined in Touch Sense Schedule 3.
- Both searches were started at point 2 and the offset information is stored in position register 1.
- Points 5, 6, and 7 are offset according to the results of the search.

```
PROG_01
1:  J P[1] 100% FINE
2:    Search Start[3] PR[1]
3:  J P[2] 100% FINE
4:  J P[3] 100% FINE Search[X]
5:  J P[2] 100% FINE
6:  J P[4] 100% FINE Search[-Z]
7:    Search End
8:
9:    Touch Offset PR[1]
10: J P[5] 100% FINE
11: L P[6] 20IPM CNT100
12: L P[7] 20IPM CNT100
13:   Touch Offset End
```

To perform incremental searches, each search must have its own start point. In the example program, line 5 was included so the incremental search feature could be used for the second search. If incremental is turned *off* , line 5 could be removed and both searches would start at the taught location of position 2.

18.5.6 Master Flag

The first time the program is executed the part must be mastered.

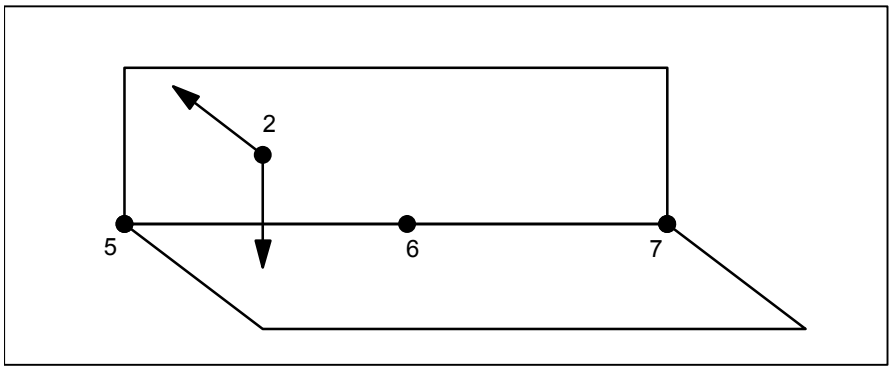
- Mastering is done by turning ON the Master Flag in the Touch Sense Schedule 3.
- Execute the program.
- The search is performed and the path is followed according to the taught positions.
- After the program is completed, the Master Flag is turned OFF.

Note Incremental search is disabled while the Master Flag is turned ON.

Program Example

If incremental search does not appear to be operating as expected, check the Master Flag. The Master Flag might have been inadvertently left on.

Figure 18–25. Part with One Touch Sense Start Position, 2, and Three Points along a Path, 5, 6, 7

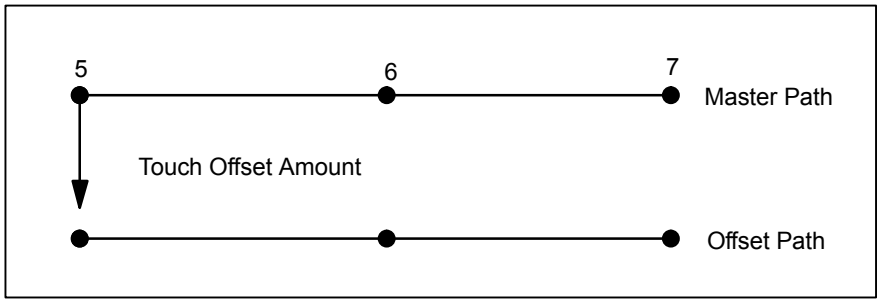



```
PROG_01
1:  J P[1] 100% FINE
2:    Search Start[3] PR[1]
3:  J P[2] 100% FINE
4:  J P[3] 100% FINE Search[X]
5:  J P[2] 100% FINE
6:  J P[4] 100% FINE Search[-Z]
7:    Search End
8:
9:    Touch Offset PR[1]
10: J P[5] 100% FINE
11: L P[6] 20IPM CNT100
12: L P[7] 20IPM CNT100
13:   Touch Offset End
```

- The path represented by points 5, 6, and 7 will be offset by the amount stored in position register 1.
- Figure 18–26 shows the position of the master path.
- The search is performed and the offset from the master location is computed and stored in position register 1.

- The offset is then applied to the master path to produce the *new*, offset path.

Figure 18–26. Illustration of the Path when an Offset is Applied



18.5.7 Touching Up Path Positions

Occasionally the part or its placement on a fixture will change requiring adjustment of the path. The entire process of remastering is not need to accommodate these changes. Refer to Figure 18–27 for an illustration of offset path touchup to adjust the location of points.

Note Touch up must be performed after a successful touch sense and at the same time the offset is being applied. Figure 18–28 illustrates the result of the touch up process.

Figure 18–27. Offset Path Touch Up to Adjust Location of Points 6 and 7

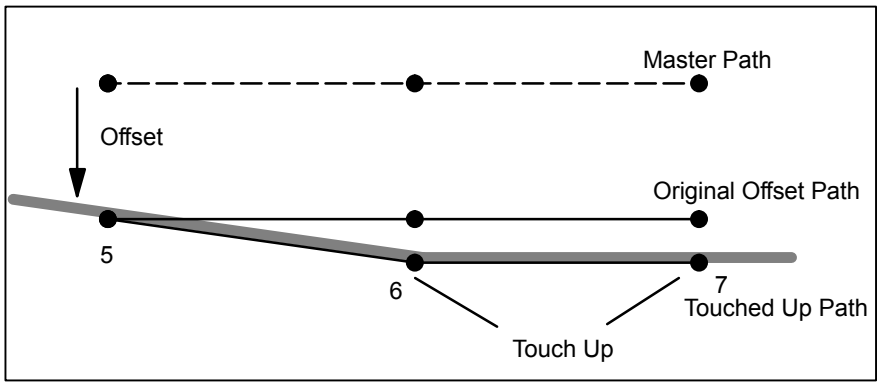
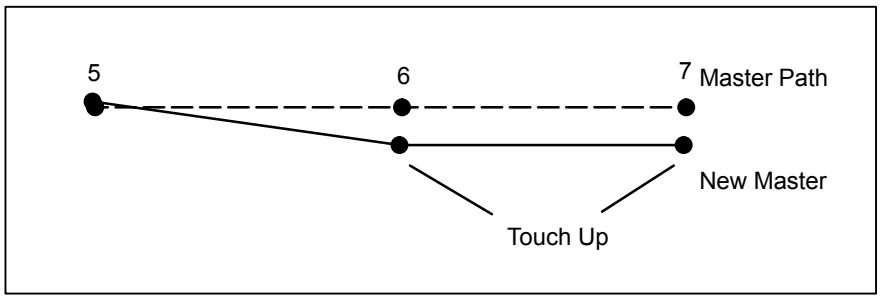


Figure 18–28. New Master Touch Up Illustration



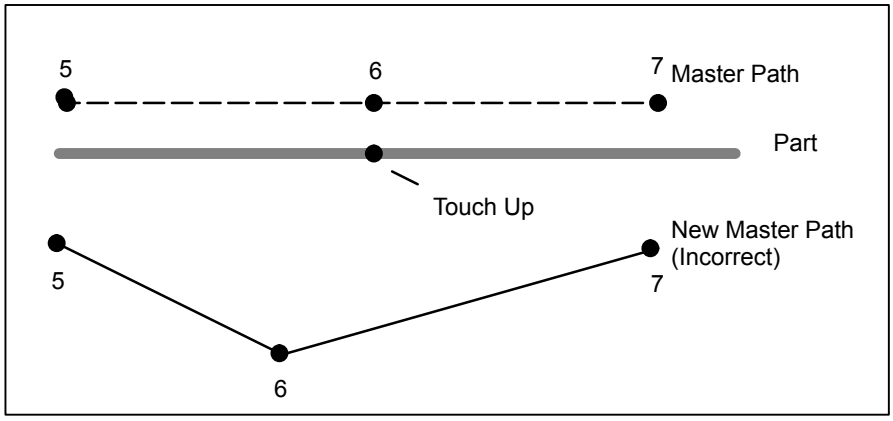
Incorrect Touch Up

A common error is to alter the path without the correct offset being applied. Touching up must be done after executing the search and while the Touch Offset is applied.

An example of an incorrect touch up is as follows:

- You can move through the program without executing the touch sense.
- You can touch up point 6 to place it on the part. Refer to Figure 18–29 .
- The master path has been altered as shown by the new master path. It was originally intended for the path to be straight and follow the part.

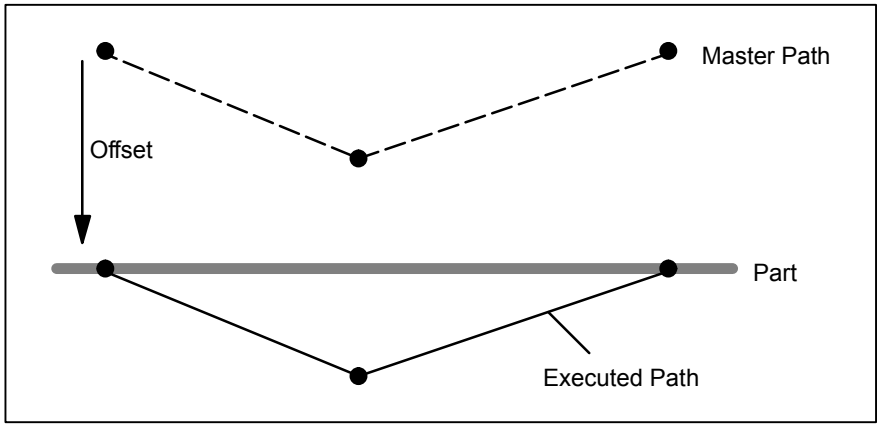
Figure 18–29. Incorrect Touch Up of a Path



- The part will not be followed correctly when the program is run. Refer to Figure 18–30 .
- The offset shifts points 5 and 7 to the correct location along the part. Point 6 will not be along the part since the master path was incorrectly touched up.

- Figure 18–30 exhibits the path that was followed after altering one point. It shows that the part is not followed correctly.

Figure 18–30. Path Followed After Altering One Point



18.5.8 Adding New Positions

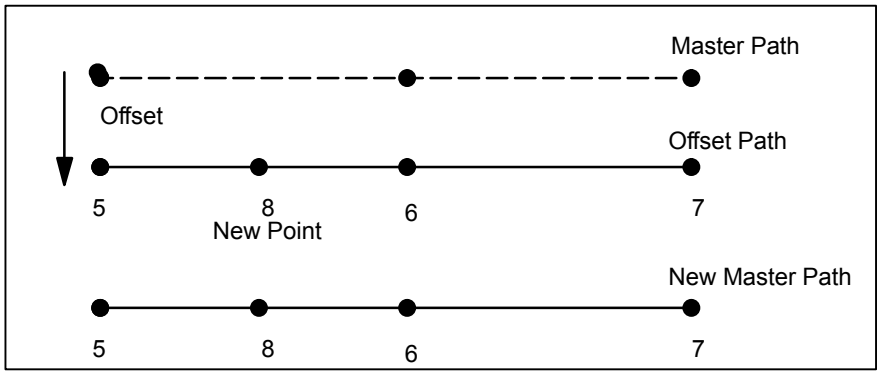
Additional points can be added in the same manner as touching up.

- The search must be completed.
- An accurate offset must be generated.
- Points can then be added to the offset path.
- The program is executed by first performing the search and then generating a valid offset.

Figure 18–31 illustrates adding a point to a path. The offset must be actively applied for the master path to be correctly updated.

Note If the program is ABORTED while adding new positions, the offset is cancelled. New positions will be taught as actual locations rather than positions with an offset applied. The results would be similar to what is shown in Figure 18–30 .

Figure 18–31. New Point Taught while Executing the Offset Path



18.5.9 Multiple Searches

Complex programs can have multiple searches generating several offsets as shown in Figure 18–32 .

Program Example

The following program example shown in Figure 18–32 exhibits two searches that can be performed for complex shapes.

- The first search stores the offset data in position register 1 with positions 10, 11, and 12 using the offset.
- The second search stores offset data in position register 2 with positions 13, 14, and 15 using the offset.

If a position of the taught path is to be touched up, the corresponding search must be performed. Figure 18–33 shows the complex part with a section moved and the path represented by positions, 10, 11, and 12 which must be touched up.

- The first search must be executed to obtain an accurate offset.
- The offset is applied and the positions, 10, 11, and 12 can be touched up as normal.
- The master will be correctly updated.

If the path using positions 13, 14, and 15 must be touched up, the second search must be executed.

- The second search stores offset data in position register 2 with positions 13, 14, and 15 using the offset.

Note Using this method can reduce the amount of time required to adjust a small section of the program.

See the following screen for an example.

```
PROG_01
1:   J P[1] 100% FINE
2:     SEARCH START[3] PR[1]
3:   J P[2] 100% FINE
4:   J P[3] 100% FINE SEARCH [X]
5:   J P[4] 100% FINE SEARCH [-Z]
6:     SEARCH END
7:   J P[5] 100% FINE
8:     SEARCH START[4] PR[2]
9:   J P[6] 100% FINE
10:  J P[7] 100% FINE SEARCH [-X]
11:  J P[8] 100% FINE
12:  J P[9] 100% FINE SEARCH [-Z]
13:     SEARCH END
14:
15:     TOUCH OFFSET PR[1]
16:  J P[10] 100% FINE
17:  L P[11] 20IPM CNT100
18:  L P[12] 20IPM CNT100
19:     TOUCH OFFSET END
20:
21:     TOUCH OFFSET PR[2]
22:  J P[13] 100% FINE
23:  L P[14] 32IPM CNT100
24:  L P[15] 32IPM CNT100
25:     TOUCH OFFSET END
```

Figure 18–32. Multiple Searches can be Performed for Complex Shapes

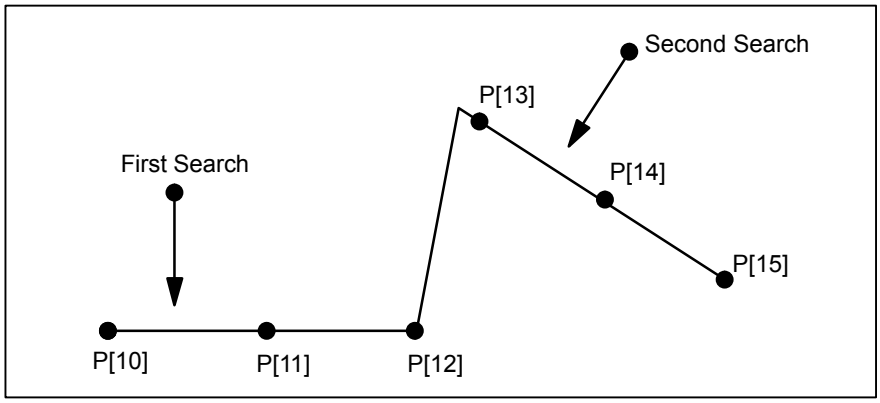
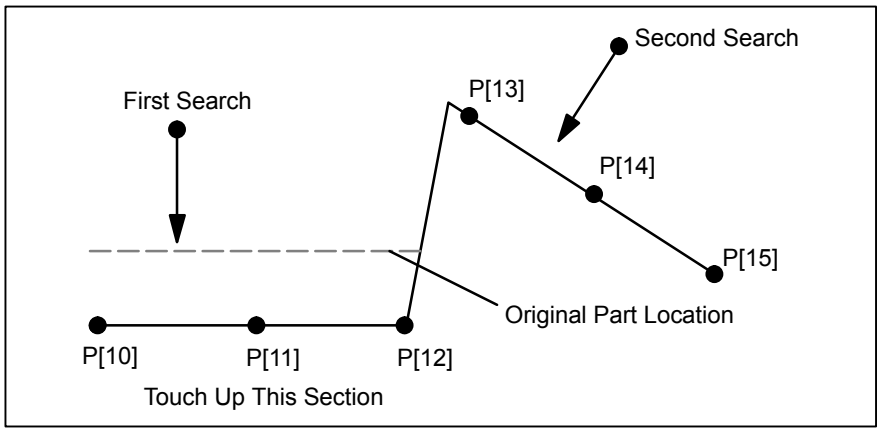


Figure 18–33. Illustration of Part Shape Change and the Effect on Multiple Searches Performed



18.5.10 Touching Up Search Start Positions

Touching up a search start position is different from touching up the path position. If the search start position is moved, then the search and affected path positions must be remastered. There is one exception:

- Moving the search start position along the axis of the search.

Program Example

The following program example shown in Figure 18–34 exhibits a part and search start position.

If the search start position is too close to the part due to poor programming, changes in the part, or a change in the part location, then

- The search start position needs only to be moved back along the search direction.
- This can be accomplished with no effect on the path positions and remastering will not be required.

Program Example

The program example shown in Figure 18–35 shows the search start position moved to a new location off the axis of the search direction.

If the search position is moved off the axis of the search direction, then:

- Remastering is required. To remaster refer to Section 18.5.3 .

Figure 18–34. Moving a Search Start Position along the Search Direction

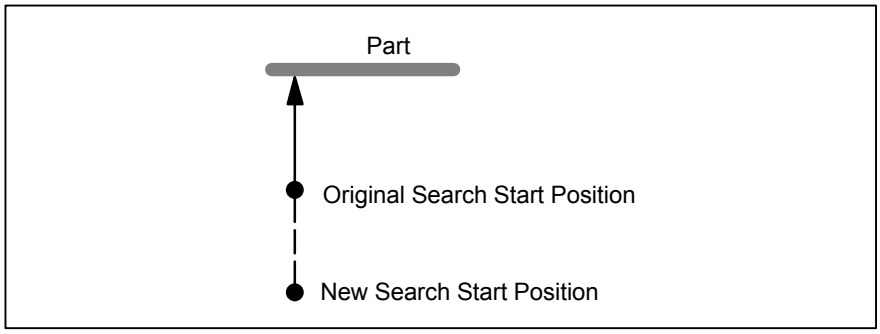
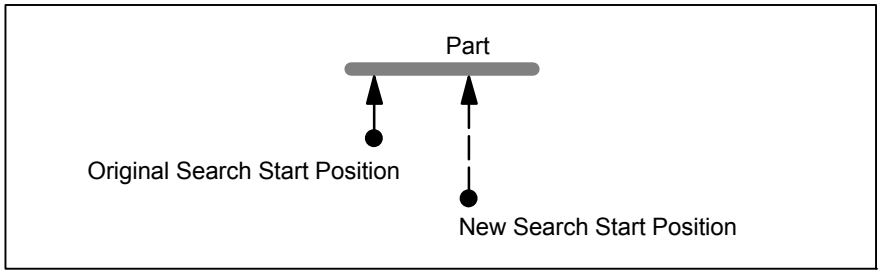


Figure 18–35. Search Start Position moved to a New Location Off the Axis of the Search Direction



18.6 TOUCH SENSING HARDWARE

18.6.1 Hardware Overview

Typically for GMAW (Gas Metal Arc Welding), a low voltage signal is applied to the welding wire. When contact is made with the object, the circuit is completed and the required input signal is sent to the robot. When the input is received, the current robot tool center point (TCP) position is stored and robot search motion is stopped.

The touch sensing circuit is enabled in a program by the SEARCH START instruction that turns on an output that has been assigned for touch sensing.

18.6.2 Touch Sensing Input Signal

The touch sensing input signal being monitored during the touch sensing routine can be any one of the following:

- Robot Digital Inputs (RI) 1-16
- Digital Inputs (DI), can be any DI
- Welding Digital Inputs (WI) 1-8
- Wire stick detection circuit input WSI, an internal input through the process I/O WDI+, WDI-

You can also set up touch sensing to monitor the condition of any RDO or DO signal as an input signal. When the selected output turns on during a touch sensing routine, the controller reads this as a received input signal.

Refer to Section 18.2.2 for more information.

18.6.3 Touch Sensing Enable/Disable Output Signal

Any one of the following outputs can be selected as the output to enable and disable the touch sensing circuitry:

- Robot digital output (RO) 1 - 16
- Digital Outputs (DO), can be any DO

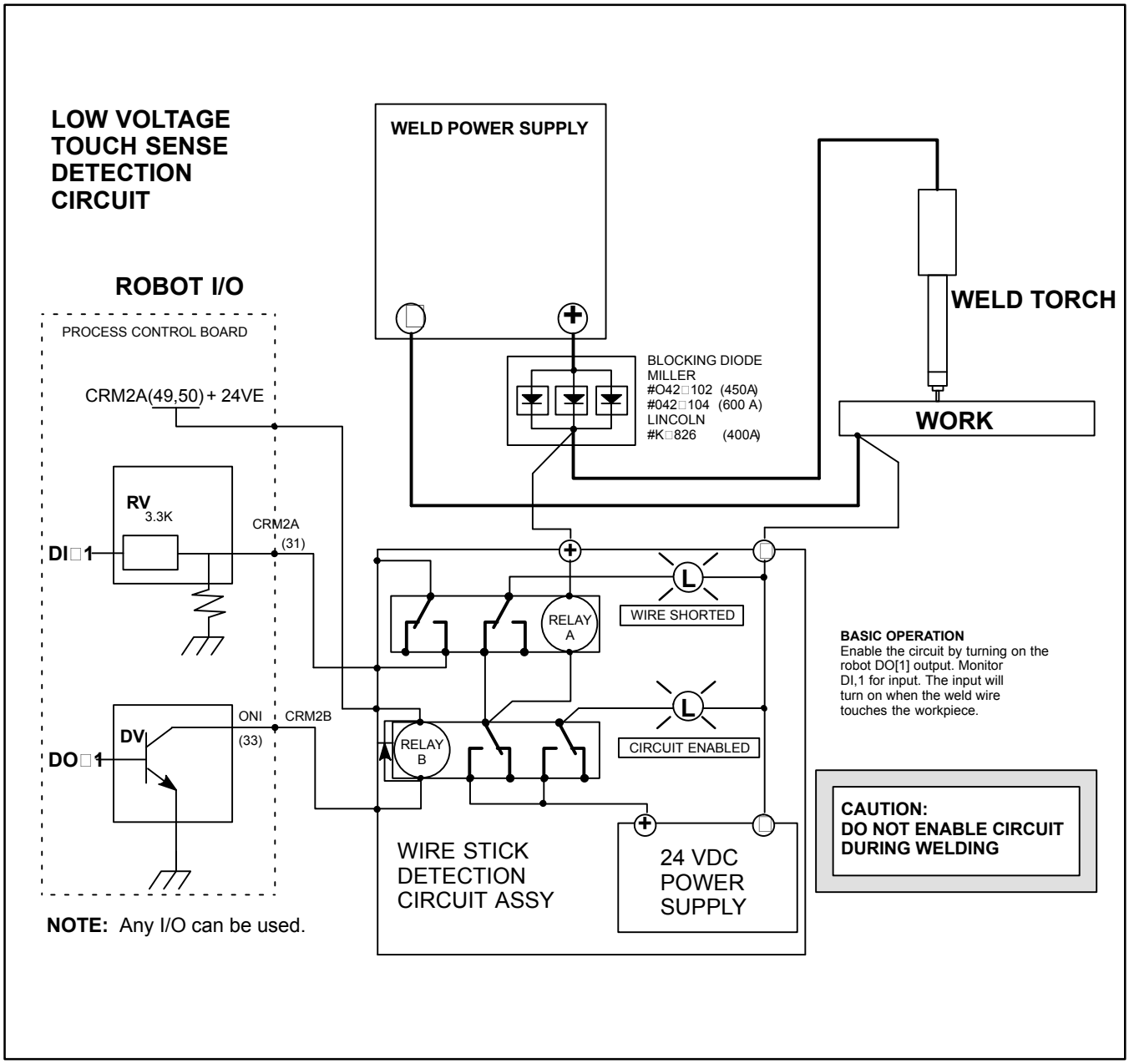
- Welding Digital Outputs (WO) 1 - 8
- Wire stick detection circuit enable WSE, an internal output on the process I/O board that enables the wire stick detection circuit for touch sensing

Refer to Section 18.2 for more information on how to assign this output.

18.6.4 Simple Low Voltage Touch Sense Detection Circuit

Figure 18–36 shows the schematic for a simple low voltage circuit. Any other circuit that will provide the required input can be used.

Figure 18–36. Simple Low Voltage Touch Sense Detection Circuit



18.7 LASER ANALOG INPUT NON-CONTACT TOUCH SENSING

18.7.1 Overview

Touch Sensing Search time requires a significant portion of the overall cycle time. The current search time in the Touch Sensing Search operation normally takes about two seconds per search. Laser Analog Input Non-Contact Touch Sensing reduces the search time to less than one second. There are no additional teach pendant instructions required. Touch Sensing will not issue a search motion if Analog Input is selected. It will read the analog value and convert it to the offset distance based on the calibration data you provide in a data file.

Analog Input Non-Contact Touch sensing has the following limitations:

- Coordinated motion is not supported in the current implementation.
- It currently supports a single arm robot and supports only the first robot arm of a multi arm system.

18.7.2 System Setup

Hardware Setup

You are responsible for the selection of the proper laser sensor and I/O device. The recommended laser sensor should offer CCD technology which is not sensitive to color, texture, and angle changes. Both Keyence LK-503 and Banner LG10A65NIQ can be selected, but Banner LG10A65NIQ is a more cost effective option.

Laser Analog Input Calibration Data

A calibration data file “frs:th_calib.dt” is required with the following format:

```
MaxAnalogInput
MinAnalogInput
TotalDataPair
DistSensorToPart AnalogInput

distance 1    analog voltage 1
distance 2    analog voltage 2
.....

distance 3    analog voltage n
```

Note Use an offline editor to create this file and then save it in the FRS: device.

For example, the following calibration data file is created based on laser sensor Banner LG10A65NIQ. It has 52 analog reading data pairs with a 1 mm distance difference between two consecutive pairs. The minimum number of data pairs is 10, and the maximum is 100.

```
!! Touch Sensing Analog Input Calibration Data File
!
MaxAnalogInput
 909
MinAnalogInput
  256
TotalDataPair   52
!


DistSensorToPart    AnalogInput

125.659             909
124.659             897
123.659             884
122.659             871
121.659             860
120.659             847
119.659             834
118.659             822
117.659             810
116.659             796
!
115.659             785
114.659             771
113.659             759
112.659             746
111.659             734
110.659             722
109.659             709
108.659             697
107.659             685
106.659             673


105.659             661
104.659             648
103.659             636
102.659             623
```

```
101.659             610
100.659             598
99.659              585
98.659              571
97.659              558
96.659              545
!
95.659              532
94.659              519
93.659              506
92.659              493
91.659              480
90.659              467
89.659              454
88.659              441
87.659              428
86.659              414
!
85.659              401
84.659              388
83.659              375
82.659              362
81.659              348
80.659              335
79.659              322
78.659              309
77.659              296
76.659              282
!
75.659              270
74.659              256
```

18.7.3 Analog Input Touch I/O Setup

You can assign the Touch Sensing inputs and outputs from the Touch I/O Setup screen.

Refer to Table 18–4 for information on the Touch I/O Setup Screen Items. Use Procedure Procedure 18-7 to set the sensor port type to Analog Input.

Table 18–4. Touch I/O Setup Screen Items

ITEM	DESCRIPTION
Sensor port type	Select RDI, WDI, DI, or WDI to be assigned to the touch sensor.
Sensor port number	Set the input number for the touch input
Circuit port type	Select RDO, WDO, DO or WDI to to be assigned to the touch circuit.
Circuit port number	Set the output number for the touch output.

Procedure 18-7 Setting the sensor port type in Touch I/O

1. Press MENUS.
2. Select I/O
3. Press F1, [TYPE]
4. Select Touch I/O Setup
5. Move the cursor to Sensor port type and select AI. You will see a screen similar to the following.

```
Touch I/O Setup
                                    1/4
          NAME           VALUE

   1
Sensor port type:    AI

   2 Sensor port number:  1
   3 Circuit port type:   RDO
   4 Circuit port number: 1
```

6. Set each item as desired.

18.7.4 Analog Input Search Direction

Analog Input Search Direction Guidelines

The laser sensor needs to be aligned with the search direction to get an accurate analog reading. The search direction is defined by the touch frame:

- You are responsible for aligning the laser with the search direction for the Search operation.
- The search direction should be perpendicular to the part during the master operation.

- For the rotation search pattern type, the robot TCP needs to be set at the emitted light on the laser sensor front panel.
- You can conduct a dry run by selecting the DI sensor type. Then the laser search direction can be verified with an actual search motion.



Caution

Extra care should be taken during the dry run, otherwise damage to the laser sensor could occur.

Appendix A

ERROR CODES AND RECOVERY

Contents

Appendix A	ERROR CODES AND RECOVERY	A–1
A.1	OVERVIEW	A–2
A.2	ERROR CODE PROPERTIES	A–3
A.2.1	Overview	A–3
A.2.2	Facility Name and Code	A–8
A.2.3	Severity Descriptions	A–12
A.2.4	Error Message Text	A–15
A.3	GENERAL ERROR RECOVERY PROCEDURES	A–17
A.3.1	Overview	A–17
A.3.2	Overtravel Release	A–17
A.3.3	Hand Breakage Recovery	A–19
A.3.4	Pulse Coder Alarm Recovery	A–20
A.3.5	Chain Failure Detection Error Recovery	A–22

A.1 OVERVIEW

Note Refer to the *Error Code Manual* for error code listings, causes, and remedies.

Errors occur because of

- Hardware problems - a broken cable or tooling
- Software problems - incorrect program or data
- External problems - an open safety door or an overtravel has occurred

Depending on the severity of the error, you must take certain steps to recover from it.

A complete listing of error codes is provided in the *Error Code Manual*. Use Procedure A-1 as the recommended error recovery procedure.

Some errors require minimal corrective action to recover from them. Others require more involved procedures. The first step in the error recovery process is to determine the kind and severity of the error. After you determine this information, the appropriate error recovery procedure can be used.

Procedure A-1 Error Recovery Recommendation

Conditions

- An error has occurred.

Steps

1. Determine the cause of the error.
2. Correct the problem which caused the error.
3. Release the error.
4. Restart the program or robot.

   If the basic recovery procedures do not clear the error, try restarting the controller. Refer to Table A–1 for the methods of starting the controller. First try a Cold start. If Cold start does not solve the problem, try a Controlled start and then a Cold start. If the problem still exists, refer to the *Software Installation Manual* to reload software if necessary.

Table A–1. Startup Methods

Start Method	Description	Procedure
Cold start (START COLD)	Initializes changes to system variables Initializes changes to I/O setup Displays the UTILITIES Hints screen	Turn the power disconnect circuit breaker to ON. When the BMON screen is displayed on the teach pendant, press and hold the SHIFT and RESET keys. After you see files beginning to load on the teach pendant screen, release all of the keys.
Controlled start (START CTRL)	Allows you to set up application specific information Allows you to install options and updates Allows you to save specific information Allows you to start KCL Allows you to print teach pendant screens and the current robot configuration Allows you to unsimulate all I/O Does not allow you to load teach pendant programs	Press FCTN and select CYCLE START, select YES, and press ENTER. When the BMON screen is displayed on the teach pendant, press and hold the PREV and NEXT keys. After the Configuration Menu screen is displayed, release the keys. Select Controlled start and press ENTER.

A.2 ERROR CODE PROPERTIES

A.2.1 Overview

An error code consists of:

- The facility name and error code number
- The severity of the error
- The message text of the error

Refer to Section A.2.2 , Section A.2.3 , and Section A.2.4 .

The error code will be displayed as follows:

```
FACILITY_NAME - ERROR_CODE_NUMBER Error message text
```

The Alarm Log screen displays a list of errors that have occurred. There are two ways to display alarms:

- **Automatically** using the Active Alarm screen. This screen displays only active errors (with a severity other than WARN) that have occurred since the last time RESET was pressed.
- **Manually** using the History Alarm screen. This screen displays up to the last 100 alarms, regardless of their severity. You can also display detailed information about a specific alarm. Optionally, you can set up your system to store additional alarms in an alarm log called MD:errext.ls. If you have the Extended Alarm Log option installed and set up, then this file, stored on the MD: device, will display up to the last 1000 alarms. In this case, the alarms are numbered and contain the date, time, error message, cause code, and severity. Programming events will be displayed the same as error messages but will not have cause codes or severities.

  For information on setting up the Extended Alarm Log, refer to the *Software Installation Manual*.

Table A–2 through Table A–7 describe each kind of alarm that can be displayed.

Table A–2. Alarm Log Screen

ITEM	DESCRIPTION
Alarm Status	This item allows you to monitor the entire list of active alarms. Press F3, HIST, to display the history of alarms.

Table A–3. Application Alarm Screen

ITEM	DESCRIPTION
Application Alarm Status	This item allows you to monitor application alarms. Application alarms include anything that is related to the given application-specific tool that is loaded.

Table A–4. Comm Log Screen

ITEM	DESCRIPTION
Comm Log Status	This item allows you to monitor any communication alarms when a communication option is loaded.

Table A–5. Fault Recovery Screen

ITEM	DESCRIPTION
Fault Recovery Status	This item allows you to monitor fault recovery status.

Table A–6. Motion Alarm Screen

ITEM	DESCRIPTION
Motion Alarm Status	This item allows you to monitor the Motion Alarm screen status such as, SRVO alarms or any other alarms related to robot movement.

Table A–7. System Alarm Screen

ITEM	DESCRIPTION
System Alarm Status	This item allows you to monitor system alarm messages such as SYST alarms.

Use Procedure A-2 to display the Alarm Log screen.

Procedure A-2 Displaying the Alarm Log Automatically

Conditions

- To display the Active Alarm screen automatically,
  - — Set the system variable $ER_AUTO_ENB to TRUE either on the SYSTEM Variables menu or by setting Auto display of alarm menu to TRUE on the SYSTEM Configuration menu. Then perform a Cold start.
  - — An error, whose severity is either PAUSE or ABORT must have occurred.

Steps

1. The following screen will automatically be displayed. It lists all errors with a severity other than WARN, that have occurred since the last controller RESET. The most recent error is number 1.

```
SRVO-007 External emergency stop
TEST1           LINE 15           ABORTED
Alarm: ACTIVE
1 SRVO-007 External emergency stop
```

2. To toggle between the Active Alarm screen and Hist Alarm screen, press F3 (ACTIVE or HIST).
3. If you are using an *i*Pendant and are in Single Display Mode, you can toggle between wide screen and normal screen. Press F2, [VIEW] and select Wide or Normal.
4. To disable the automatic display of all errors with a certain severity type, modify the value of the system variable $ER_SEV_NOAUTO[1-5]. These errors will still be logged in the Active Alarm screen, but they will no longer force the screen to immediately become visible. Refer to the *Software Reference Manual* for more detailed information about how to set these variables.

5. To disable the automatic display of a specific error code, modify the $ER_NOAUTO.$noalm_num and $ER_NOAUTO.$er_code system variables. These errors will still be logged in the Active Alarm screen, but they will no longer force the screen to immediately become visible. Refer to the *Software Reference Manual* for more detailed information about how to set these variables.
6. To display the screen that occurred immediately before the alarm, press RESET. If you have toggled between HIST and ACTIVE, the previous screen might not be available.

   When there are no active alarms (the system is not in error status), the following message will be displayed on the Active Alarm screen.

```
There are no active alarms.
Press F3(HIST) to enter alarm history screen.
```

Note When you reset the system by pressing the RESET key, the alarms displayed on this screen are cleared.

Procedure A-3 Displaying the Alarm Log Manually

Steps

1. Press MENUS.
2. Press ALARM.
3. Press F3, HIST.
4. Press F1, [TYPE].
5. Select Alarm Log. The alarm log will be displayed. This lists all errors. See the following screen for an example.

```
SRVO-007 External emergency stop
TEST1          LINE 15         ABORTED
Alarm: HIST
1 SRVO-007 External emergency stop
2 SRVO-001 Operator panel emergency st
3 R E S E T
4 SRVO-029 Robot calibrated (Group:1)
5 SRVO-001 Operator panel emergency st
6 SRVO-012 Power fail recovery
7 INTP-127 Power fail detected
8 SRVO-047 LVAL alarm (Group:1 Axis:5)
9 SRVO-047 LVAL alarm (Group:1 Axis:4)
10 SRVO-002 Teach pendant emergency stop
```

Note The most recent error is number 1.

- To display the complete error message that does not fit on the screen, press F5, DETAIL, and the right arrow key on the teach pendant.
- To display the cause code for an error message, press F5, DETAIL. Cause codes provide further information about the cause of the error. If the specified error has a cause code, the cause code message is displayed immediately below the error line, on the status line. When you press RESET, the error and cause code disappears and the status line is redisplayed.

6. **To display the motion log**, which lists only motion-related errors, press F1, [TYPE], and select Motion Log.
7. **To display the system log**, which displays only system errors, press F1, [TYPE], and select System Log.
8. **To display the application log**, which displays only application-specific errors, press F1, [TYPE], and select Appl Log.
9. **To display the reporting log which displays the top five alarms that occurred while the robot was in Auto mode** , press F1, [TYPE], and select Reporting.
    1. **To view the top five faults by incident**, press F2, INCIDENT.
    2. **To view the top five faults by total asserted time**, press F3, T_TIME.
    3. **To view the top five faults by longest asserted time** for any given fault, press F4, L_TIME.
10. **To display the communication log**, which displays only communication-specific errors, press F1, [TYPE], and select Comm Log.
11. **To display the password log**, which displays only password-specific errors, press F1, [TYPE], and select Password Log.

12. **To display more information about an error**, move the cursor to the error and press F5, DETAIL. The error detail screen displays information specific to the error you selected, including the severity. If the error has a cause code, the cause code message will be displayed. When you are finished viewing the information, press PREV.
13. **To display cause and remedy information specific to an error,** move the cursor to the error and press SHIFT and the Help/Diag button on the *i*Pendant.
14. **To remove all of the error messages displayed on the screen**, press and hold SHIFT and press F4, CLEAR.
15. To change the view displayed on the screen, press F2, [VIEW]. To show or hide the cause codes for each alarm, select Show Cause or Hide Cause. If an alarm has no cause code, the time the alarm occurred will be shown instead. If you are using an *i*Pendant and are in Single Display Mode, you can also toggle between wide screen and normal screen by selecting Wide or Normal.

A.2.2 Facility Name and Code

The facility name and code identify the type of error that occurred. Facility information is displayed at the beginning of the error code:

```
PROG-048 PAUSE Shift released while running
```

In the example, the facility name PROG corresponds to facility code 3. The error code number is 048. Facility codes are used in error handling from a KAREL program. The facility codes are listed in Table A–8 .

Table A–8. Error Facility Codes

Facility Name	Facility Code (Decimal)	Facility Code (Hexadecimal)	Description
ACAL	112	0x70	AccuCal II error code
APSH	38	0x26	Application shell
ARC	53	0x35	Arc welding application
ASBN	22	0x16	Mnemonic editor
ATGP	102	0x66	Attach gro lead through and force control
CALM	106	0x6a	CalMate
CD	82	0x52	Coordinated motion softpart
CMND	42	0x2a	Command processor
CNTR	73	0x4g	Continuous turn softpart

Table A–8. Error Facility Codes (Cont'd)

Facility Name	Facility Code (Decimal)	Facility Code (Hexadecimal)	Description
COND	4	0x4	Condition handler
COPT	37	0x25	Common options
CPMO	114	0x72	Constant Path error code
CUST	97	0x61	Customer specific errors
CVIS	117	0x75	Integrated Vision (Controller Vision)
DICT	33	0x21	Dictionary processor
DJOG	64	0x40	Detached jog
DMDR	84	0x54	Dual Motion Drive
DMER	40	0x28	Data monitor
DNET	76	0x4c	DeviceNet
DX	72	0x48	Delta Tool/Frame softpart
ELOG	5	0x5	Error logger
FILE	2	0x2	File system
FLPY	10	0xa	Serial floppy disk system
FRCE	91	0x5b	Impedance control (force control)
FRSY	85	0x55	Flash file system
FXTL	136		C-flex tool
HOST	67	0x43	Host communications general
HRTL	66	0x42	Host communications run time library
IBSS	88	0x58	Interbus-S
INTP	12	0xc	Interpreter internal errors
ISD	39	0x27	ISD (Integral Servo Dispenser)
JOG	19	0x13	Manual jog task
LANG	21	0x15	Language utility
LECO	109	0x6D	Arc errors from Lincoln Electric
LNTK	44	0x2c	Line tracking

Table A–8. Error Facility Codes (Cont'd)

Facility Name	Facility Code (Decimal)	Facility Code (Hexadecimal)	Description
LSTP	108	0x6C	Local Stop error codes
MACR	57	0x39	MACRO option
MARL	83	0x53	Material removal
MCTL	6	0x6	Motion control manager
MEMO	7	0x7	Memory manager
MENT	68	0x44	ME-NET
MHND	41	0x29	Material Handling shell and menus
MOTN	15	0xf	Motion subsystem
MUPS	48	0x30	Multi-pass motion
NOM	133		Nominal Position
OPTN	65	0x41	Option installation
OS	0	0x0	Operating system
PALL	115	0x73	PalletTool
PALT	26	0x1a	Palletizing application
PMON	28	0x1c	PC monitor
PNT1	86	0x56	Paint Application Errors POST V6.31
PRIO	13	0xd	Digital I/O subsystem
PROF	92	0x5c	Profibus DP
PROG	3	0x3	Interpreter
PWD	31	0x1f	Password logging
QMGR	61	0x3d	KAREL queue manager
RIPE	130		Ros IP errors

Table A–8. Error Facility Codes (Cont'd)

Facility Name	Facility Code (Decimal)	Facility Code (Hexadecimal)	Description
ROUT	17	0x11	Softpart built-in routine for interpreter
RPC	93	0x5d	RPC
RPM	43	0x2b	Root Pass Memorization
RTCP	89	0x59	Remote TCP
SCIO	25	0x19	Syntax checking for teach pendant programs
SEAL	51	0x33	Sealing application
SENS	58	0x3a	Sensor interface
SHAP	79	0x4f	Shape generation
SPOT	23	0x17	Spot welding application
SPRM	131		Ramp motion softpart
SRIO	1	0x1	Serial driver
SRVO	11	0xb	FLTR & SERVO in motion sub-system
SSPC	69	0x45	Special space checking function
SVGN	30	0x1e	Servo weld gun application
SYST	24	0x18	Facility code of system
TAST	47	0x2f	Through-Arc Seam Tracking
TCPP	46	0x2e	TCP speed prediction
TG	90	0x5a	Triggering accuracy
THSR	60	0x3c	Touch Sensing softpart
TJOG	116	0x74	Tracking Jog
TMAT	119	0x77	Torch Mate
TOOL	29	0x1d	Servo tool change
TPIF	9	0x9	Teach pendant user interface
TRAK	54	0x36	Tracking softpart
TRSV	134		Tray server task error text
VARS	16	0x10	Variable Manager Subsystem
WEAV	45	0x2d	Weaving

Table A–8. Error Facility Codes (Cont'd)

Facility Name	Facility Code (Decimal)	Facility Code (Hexadecimal)	Description
WMAP	103	0x67	All wafer handling robot retated Errors
WNDW	18	0x12	Window I/O manager sub-system
XMLF	129		XML errors

A.2.3 Severity Descriptions

The severity of the error indicates how serious the error is. The severity is displayed after the error number. For example:

```
PROG-048 PAUSE Shift released while running
```

Note You can display the severity of the error code on the ALARM screen. Refer to Procedure A-2 .

$ER_SEV_NOAUTO[1-5] System Variable

The $ER_SEV_NOAUTO[1-5] system variable enables or disables the automatic display of all error codes with a particular *severity.* This is used in conjunction with the $ER_AUTO_ENB system variable.

Table A–9. Severity Descriptions

SEVERITY	$ER_SEV_NOAUTO[1-5]
PAUSE	[1]
STOP	[2]
SERVO	[3]
ABORT	[4]
SYSTEM	[5]

WARN

WARN errors only warn of potential problems or unexpected circumstances. They do not directly affect any operations that might be in progress. If a WARN error occurs, you should determine what caused the error and what, if any, actions should be taken.

For example, the WARN error **Singularity position** indicates a singularity position was encountered during a move. No action is required. However, if you do not want the motion to encounter a singularity position, you can reteach the program positions.

PAUSE

PAUSE errors pause program execution but allow the robot to complete its current motion segment, if any are in progress. This error typically indicates that some action must be taken before program execution can be resumed. PAUSE errors cause the operator panel FAULT light to go on and the teach pendant FAULT LED to go on.

Depending on the action that is required, you might be able to resume a paused program at the point where the PAUSE error occurred after you have corrected the error condition. If the program can be resumed, you can either select the RESUME function key or press the operator CYCLE START button, or press the UOP CYCLE START button if the setting of the REMOTE/LOCAL setup item on the System Configuration menu is set to LOCAL.

STOP

STOP errors pause program execution and stop robot motion. When a motion is stopped, the robot decelerates to a stop and any remaining part of the current motion segment is saved, meaning the motion can be resumed. STOP errors usually indicate that some action must be taken before the motion and program execution can be resumed.

Depending on the action that is required, you might be able to resume the motion and program execution after correcting the error condition. If the motion and program can be resumed, you can either select the RESUME function key or press the operator CYCLE START button if the setting of the REMOTE/LOCAL setup item on the System Configuration menu is set to LOCAL. If the robot is in production mode, you must choose the appropriate recovery option.

SERVO

SERVO errors shut off the drive power to the servo system and pause program execution. SERVO errors cause the operator panel FAULT light to go on and the teach pendant FAULT LED to go on.

SERVO errors are usually caused by hardware problems and could require trained service personnel. However, some SERVO errors require you to reset the servo system by pressing the operator panel FAULT RESET button or the teach pendant RESET key. Others require a Cold start of the controller.

ABORT

ABORT errors abort program execution and STOP robot motion. When an ABORT error occurs, the robot decelerates to a STOP and the remainder of the motion is canceled. An ABORT error indicates that the program has a problem that is severe enough to prevent it from continuing to run.

You will need to correct the problem and then restart the program. Depending on the error, correcting the problem might mean editing the program or modifying the data.

SYSTEM

SYSTEM errors usually indicate a system problem exists that is severe enough to prevent any further operation. The problem could be hardware or software related.

You will need the assistance of trained service personnel to correct SYSTEM errors. After the error has been corrected, you will need to reset the system by turning off the robot, waiting a few seconds, and turning on the robot.

If a program was executing when the error occurred, you will need to restart the program.

ERROR

ERROR errors occur during the translation of a KAREL program. When an ERROR error occurs, translation is stopped and a .PC file is not generated. Fix the error in the program and retranslate it. When you translate a program and no ERROR errors occur, translation is successful and a .PC file is generated.

NONE

NONE errors can be returned as status from some KAREL built-in routines and can also be used to trigger KAREL condition handlers. NONE errors are not displayed on the teach pendant or CRT/KB. They also are not displayed on the alarm log screen. NONE errors do not have any effect on programs, robot motion, or servo motors.

Table A–10 summarizes the effects of error severities.

Table A–10. Effects of Error Severity

Severity	Program	Robot Motion	Servo Motors
WARN	No effect	No effect	No effect
PAUSE	Paused	The current move is completed then the robot stops.	No effect
STOP	Paused	Decelerated STOP, motion retained	No effect
SERVO	Paused	Decelerated STOP, motion retained	Power shutdown
ABORT	Aborted	EMERGENCY STOP, motion canceled	No effect
SYSTEM	Aborted	EMERGENCY STOP, motion canceled	Power shut down Requires a FCTN:CYCLE POWER
ERROR	No effect	No effect	No effect
NONE	No effect	No effect	No effect

A.2.4 Error Message Text

The message text describes the error that has occurred. Message text is displayed at the end of the error code. For example:

```
PROG-048 PAUSE Shift released while running
```

Some error messages might contain cause codes, percent (%) notation, or hexadecimal notation. For more information on displaying cause codes, refer to Procedure A-2 .

Percent Notation (%)

A percent sign followed by the letter *s* (%s) indicates that a string, representing a program name, file name, or variable name, actually appears in the error message when the error occurs.

A percent sign followed by the letter *d* (%d) indicates that an integer, representing a program line number or other numeric value, actually appears in the error message when the error occurs.

For example:

```
INTP-327 ABORT (%^s, %d^5) Open file failed
```

When this error occurs, the actual name of the file that could not be opened will appear on the teach pendant error line instead of %s. The actual program line number on which that error occurred will appear on the teach pendant error line instead of %d.

Hexadecimal Notation

Hexadecimal notation is used to indicate the specific axes in error, when one or more axes are in error at the same time.

Most robots have interaction limits, in addition to normal joint limits. Even when all axes are within their respective limits an error might occur. This could possibly be caused by the interaction between multiple axes. In this case, hexadecimal notation can help you to find the specific axis in error. For example:

```
MOTN-017 STOP limit error (G:1 A:6  Hex)
```

The number after the **A** is the hexadecimal digit that shows which axes are out of limit. The **Hex** indicates that the axis numbers are in hexadecimal format. Table A–11 lists the sixteen hexadecimal digits and the corresponding axes that are in error.

Note Hexadecimal digits for the decimal values of 10 through 15 are represented by the letters **A** through **F** respectively. Refer to Table A–11 .

To determine which axes are in error, you must evaluate each digit in the error message separately. Refer to Table A–11 .

Note If only one number appears in the error message after the **A:**, you must read it as the first digit.

Table A–11. Hexadecimal Error Message Display

MOTN-017 limit error (G:1 A:(3)(2) (1) HEX)			
Hexadecimal Digit	**Third Digit (3)**	**Second Digit (2)**	**First Digit (1)**
0	none	none	none
1	axis 9	axis 5	axis 1
2	n/a	axis 6	axis 2
3	n/a	axes 5 & 6	axes 1 & 2
4	n/a	axis 7	axis 3
5	n/a	axes 5 & 7	axes 1 & 3
6	n/a	axes 6 & 7	axes 2 & 3
7	n/a	axes 5, 6, & 7	axes 1, 2, & 3
8	n/a	axis 8	axis 4
9	n/a	axes 5 & 8	axes 1 & 4
A	n/a	axes 6 & 8	axes 2 & 4
B	n/a	axes 5, 6, & 8	axes 1, 2, & 4
C	n/a	axes 7 & 8	axes 3 & 4
D	n/a	axes 5, 7, & 8	axes 1, 3, & 4
E	n/a	axes 6, 7, & 8	axes 2, 3, & 4
F	n/a	axes 5, 6, 7, & 8	axes 1, 2, 3, & 4
Note: If only one number appears in the error message after the **A:**, you must read it as the first digit (1).			

Table A–12 contains some examples of how to interpret Hexadecimal notation in an error message.

Table A–12. Hexadecimal Notation and Axis in Error Examples

Error	Explanation
MOTN-017 (G:1 A:6 Hex)	Axes 2 and 3 are out of their interaction limit.
MJOG-013 (G:1 A:20 Hex)	Axis 6 jogged to limit.
MOTN-017 (G:1 A:100 Hex)	Axis 9 limit error.

A.3 GENERAL ERROR RECOVERY PROCEDURES

A.3.1 Overview

This section contains procedures for recovery from certain errors. These errors are:

- Overtravel release
- Hand breakage recovery
- Pulse coder alarm
- Chain failure detection recovery

A.3.2 Overtravel Release

An overtravel error occurs when one or more of the robot axes moves beyond the software motion limits. When this happens one of the overtravel limit switches is tripped and the system does the following:

- Shuts off drive power to the servo system and applies robot brakes
- Displays an overtravel alarm error message
- Lights the operator panel FAULT light
- Turns on the teach pendant FAULT status indicator
- Limits motion for the axes involved in the overtravel

If you are jogging in JOINT, the axis number indicating the axis (or axes) in an overtravel will be displayed in the error log. You can manually release overtravel on your system from the MANUAL OT Release screen. The axis that is in overtravel will display TRUE in either OT_MINUS or OT_PLUS.

Refer to Table A–13 for information on the MANUAL OT Release items.

Use Procedure A-4 to recover from an overtravel error.

Table A–13. MANUAL OT Release Items

ITEM	DESCRIPTION
AXIS	This item displays the number for each axis.
OT MINUS	This item displays whether a particular axis is in an overtravel condition.
OT PLUS	This item displays whether a particular axis is in an overtravel condition.

Procedure A-4 Recovering from an Overtravel Error

Conditions

- An axis (or axes) are in overtravel and the overtravel alarm has occurred. If you are jogging in JOINT the axis number indicating the axis (or axes) in an overtravel will be displayed in the error log.

Steps

1. Press MENUS.
2. Select SYSTEM.
3. Press F1, [TYPE].
4. Select OT Release. You will see a screen similar to the following. The axis that is overtraveled will display TRUE in either OT_MINUS or OT_PLUS.

```
MANUAL OT Release
    AXIS     OT MINUS      OT PLUS
      1        FALSE        TRUE
      2        FALSE        FALSE
      3        FALSE        FALSE
      4        FALSE        FALSE
      5        FALSE        FALSE
      6        FALSE        FALSE
      7        FALSE        FALSE
      8        FALSE        FALSE
      9        FALSE        FALSE
```

5. Move the cursor to the OT PLUS or OT MINUS value of the axis in overtravel.
6. Press F2, RELEASE. The value of the overtraveled axis should change back to FALSE.
7. **If the robot is calibrated**, you will see the message **Can't Release OT. Press HELP for detail.**
   a. If you press F5, DETAIL, you will see a screen similar to the following.

```
MANUAL OT Release
When robot is calibrated, overtravel
cannot be released.  Press SHIFT &
RESET to clear the error, and jog out
of the overtravel condition.
```

Note For the following steps, press and **hold down the SHIFT key** until you have completed Step 7b through Step 7d .

b. Press and continue pressing SHIFT and press F2, RESET. Wait for servo power.

c. Continuously press and hold the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.

d. Jog the overtraveled axis off the overtravel switch. When you have finished jogging, **you can release the SHIFT key** .

Note If you accidentally release the shift key during Step 7b through Step 7d , you will have to repeat them.

8. If the robot is not calibrated, perform the following steps:

Note For the following steps, press and **hold down the SHIFT key** until you have completed Step 8a through Step 8d .

a. Press and continue pressing SHIFT and press F2, RESET. Wait for servo power.

b. Press COORD until you select the JOINT coordinate system.

c. Continuously press and hold the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.

d. Jog the overtraveled axis off the overtravel switch. When you have finished jogging, **you can release the SHIFT key** .

Note If you accidentally release the shift key during this step, you will need to repeat it.

9. Turn the teach pendant ON/OFF switch to OFF and release the DEADMAN switch.

10. Check CRM68 & CRF7 connection on the amplifier PCB if the robot is not in an actual overtravel condition.

A.3.3 Hand Breakage Recovery

A hand breakage error occurs when the hand breakage detection switch is tripped on robots equipped with hand breakage hardware. The switch is tripped when the robot tool strikes an obstacle, which could possibly cause the tool to break. The system

- Shuts off drive power to the servo system and applies robot brakes
- Displays an error message indicating that the hand is broken
- Lights the operator panel FAULT light

- Lights the teach pendant FAULT LED

The status of the hand breakage detection switch is displayed on the STATUS Safety Signals screen.

Use Procedure A-5 to recover from a hand breakage.

Procedure A-5 Recovering from a Hand Breakage

Conditions

- The hand breakage error message is displayed.

Steps

1. If you have not already done so, continuously press and hold the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.
2. Hold down the SHIFT key and press RESET. The robot can now be moved.
3. Jog the robot to a safe position.
4. Press the EMERGENCY STOP button.
5. Request a trained service person to inspect and, if necessary, repair the tool.
6. Determine what caused the tool to strike an object, causing the hand to break.
7. If the hand breakage occurred while a program was being executed, you might need to reteach positions, modify the program, or move the object that was struck.
8. Test run the program if it has been modified, if new positions have been recorded, or if objects in the work envelope have been moved.

A.3.4 Pulse Coder Alarm Recovery

If the pulse counts at power up do not match the pulse counts at power down, a pulse mismatch error occurs for each motion group and each axis. Use Procedure A-6 to reset a pulse coder alarm.

Procedure A-6 Resetting a Pulse Coder SRVO-062 Alarm

Steps

1. Press MENUS.
2. Select SYSTEM.
3. Press F1, [TYPE].
4. Select Master/Cal.

If Master/Cal is not listed on the [TYPE] menu, do the following; otherwise, continue to Step 5 .

a. Select VARIABLE from the [TYPE] menu.

b. Move the cursor to $MASTER_ENB.

c. Press the numeric key 1 and then press ENTER on the teach pendant.

d. Press F1, [TYPE].

e. Select Master/Cal. You will see a screen similar to the following.

```
SYSTEM Master/Cal
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE
 Press 'ENTER' or number key to select.
```

5. Press F3, RES_PCA. You will see a screen similar to the following.

```
SYSTEM Master/Cal
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE
 Press 'ENTER' or number key to select.
Reset pulse coder alarm? [NO]
```

6. Press F4, YES. You will see a screen similar to the following.

```
SYSTEM Master/Cal
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE
 Pulse coder alarm reset!
```

A.3.5 Chain Failure Detection Error Recovery

A Single Chain Failure Detection fault will be set if one safety chain is in an Emergency Stop condition, and the other is not in an Emergency Stop condition.

When a Single Chain Failure Detection fault occurs, the system does the following:

- Shuts off drive power to the servo system and applies robot brakes
- Displays an error message indicating that a single chain failure has occurred.
- Lights the operator panel FAULT light
- Lights the teach pendant FAULT LED

Chain Failure Detection Errors

Refer to the *Maintenance Manual* for more information on chain failure detection errors, SRVO-230 and 231, SRVO-266 through 275 (external checking -customer), and SRVO-370 through 385 (internal checking)

Procedure A-7 Chain Failure Detection Error Recovery

Conditions

- The system detected either a SRVO-230 Chain 1 (+24V abnormal) or a SRVO-231 Chain 2 (0V abnormal) error.
- You cannot reset the chain failure errors, even after turning the controller OFF and then ON again.

Steps

1. Correct the cause of the alarm.
2. Press MENUS.

3. Select ALARMS. You will see a screen similar to the following.

```
ALARM: Active
SRVO-230 Chain 1(+24V) abnormal
```

4. Press F4, RES_CH1. You will see a screen similar to the following

```
ALARM: Active
SRVO-230 Chain 1(+24V) abnormal
Reset Single Channel Fault [NO]
```

5. Press F4, YES to reset the fault.

6. Press the RESET button on the teach pendant or operator panel.



Warning

If you reset the chain failure fault without fixing the cause of it, the same alarm will occur, but the robot can move until the alarm occurs again. Be sure to fix the cause of the chain failure before you continue. Otherwise, you could injure personnel or damage equipment.

Appendix B

CRT/KB SETUP AND OPERATIONS

Contents

Appendix B	CRT/KB SETUP AND OPERATIONS	B–1
B.1	OVERVIEW	B–2
B.2	CRT/KB SETUP	B–3
B.2.1	Overview	B–3
B.2.2	Set the Communications Parameters on the Controller Port	B–3
B.2.3	Connect the Controller Port to the Remote CRT/KB	B–3
B.2.4	Configure the Terminal Parameters of the Remote CRT/KB	B–4
B.2.5	Program the Key Codes for the Function Keys of the Remote CRT/KB	B–4
B.3	CRT/KB MENUS	B–5
B.4	CRT/KB KEYS	B–6

B.1 OVERVIEW

The cathode ray tube/keyboard (CRT/KB) is an optional user interface device you can use, in addition to the teach pendant, to display teach pendant screens and perform robot operations. In general, you can perform any robot operation from the CRT/KB except operations that involve moving the robot, such as jogging and test cycle.

Note If you are using PaintTool, all functions relating to setting up and operating the PaintTool application cannot be displayed on the CRT/KB. These screens include those relating to setting up the application, setting up and operating the color change system, setting up and operating AccuFlow, and setting up and operating opening devices.

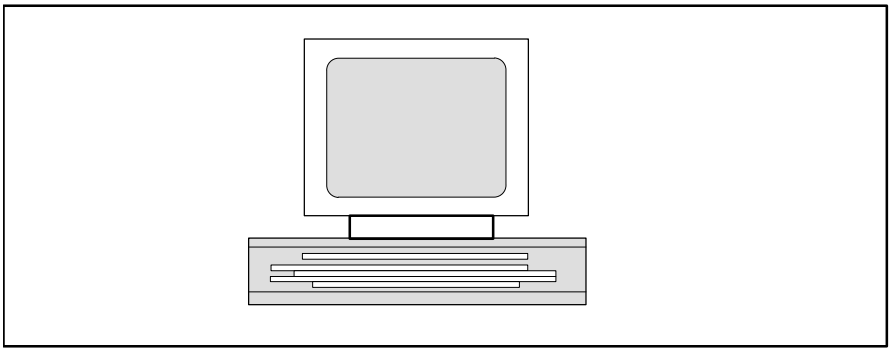
The CRT/KB is external to the controller, or *remote* . You can use the following kinds of remote CRT/KBs:

- FANUC Factory Terminal
- DEC VT-220 terminal
- IBM PC-compatible computer with VT-220 terminal emulation software

Note FANUC Robotics will supply and support only the FANUC Factory Terminal. As a courtesy to our customers, information has been provided to configure other remote CRT/KB devices. This does not imply any intent to support remote CRT/KB devices that are not supplied by FANUC Robotics.

Figure B–1 shows an example of a remote CRT/KB.

Figure B–1. Remote CRT/KBs



This appendix describes how to set up and operate the CRT/KB.

B.2 CRT/KB SETUP

B.2.1 Overview

To set up the remote CRT/KB, you must perform the following steps:

1. Set the communications parameters on the controller port that will connect to the remote CRT/KB.
2. Connect the controller port to the remote CRT/KB with a cable.
3. Configure the terminal parameters of the remote CRT/KB.
4. Program the key codes for the function keys of the remote CRT/KB.

3 and 4 are required only if you are not using the FANUC Factory Terminal as your remote CRT/KB device.

B.2.2 Set the Communications Parameters on the Controller Port

You can connect a remote CRT/KB to any unused RS-232-C port on the controller, but you must set up this port according to the requirements of your CRT/KB. To configure the port for the FANUC Factory Terminal, select the CRT/KCL device. This will configure the port correctly. Refer to Chapter 9 *PROGRAM AND FILE MANIPULATION* for information on setting up ports.

Table B–1 shows the communications settings used for the FANUC Factory Terminal. Refer to the manufacturer's specifications if you are using any other kind of remote CRT/KB. Keep in mind that for an RS-232-C port on the controller to which a CRT/KB device is connected, 9600 baud is the maximum speed available and Xon/Xoff handshaking will be recognized.

Table B–1. Port Settings for the Built-In CRT/KB and the FANUC Factory Terminal

Speed	Parity Bit	Stop Bit	Timeout Value
9600 baud	None	1 bit	0 sec

B.2.3 Connect the Controller Port to the Remote CRT/KB

The RS-232-C ports on the controller are wired in the Data Terminal Equipment (DTE) configuration and use a DB-25 female connector. For the FANUC Factory Terminal, the part number for the connector is A13B-0144-K001. For a DEC VT-220 terminal, use a null modem cable with DB-25 male connectors on both ends. For an IBM PC or compatible computer with terminal emulation

software, use a null modem cable with a DB-25 male connector on one end and either a DB-25 male or DB-9 male connector, based on the connector on your computer.

B.2.4 Configure the Terminal Parameters of the Remote CRT/KB

This step is not necessary for the FANUC Factory Terminal. For any other remote CRT/KB device, refer to the manufacturer's documentation for procedures to set the terminal parameters. The following parameters have been found to work for the DEC VT-220 and software emulation of the DEC VT-220:

- 9600 baud, 8 bits, 1 stop bit, No Parity
- Xon/Xoff handshaking
- Display 80 columns
- Interpret Controls
- No Local Echo
- VT200 Mode, 7 bit controls
- Application Keypad

B.2.5 Program the Key Codes for the Function Keys of the Remote CRT/KB

This step is not necessary for the FANUC Factory Terminal. For any other remote CRT/KB device, refer to the manufacturer's documentation for the procedures to set the correct key code of the function key or soft key. Table B–2 shows the key codes produced by the function keys and soft keys of the FANUC Factory Terminal. The table also contains a suggested mapping of DEC VT-220 function keys that retains the specialized functions of the first four function keys.

Table B–2. Key Codes and Function Key Mapping

CRT/KBKey	Key Code Transmitted	DEC VT-220 Function Key
F1	<esc>[17~	F6
F2	<esc>[18~	F7
F3	<esc>[19~	F8
F4	<esc>[20~	F9
F5	<esc>[21~	F10
F6	<esc>[23~	F11
F7	<esc>[24~	F12
F8	<esc>[25~	F13

Table B–2. Key Codes and Function Key Mapping (Cont'd)

CRT/KBKey	Key Code Transmitted	DEC VT-220 Function Key
F9	<esc>[26~	F14
F10	<esc>[28~	F15
Prev	<esc>[5~	Prev
Next	<esc>[6~	Next
Select	<esc>[4~	Select
Do	<esc>[29~	Do

B.3 CRT/KB MENUS

The contents of the menus on the CRT/KB match the menus on the teach pendant except that the CRT/KB does not display any menus that involve robot motion. If you have the KCL option, KCL will appear as a menu item on the CRT/KB and can be used to send motion commands to the robot. Refer to the *KAREL Reference Manual* for more information.

Figure B–2. CRT/KB Full Menus

```
1 UTILITIES
2 KAREL EDITOR
3 MANUAL FCTNS
4 ALARM
5 I/O
6 SETUP
7 FILE
8 KCL>
9 USER
0 -- NEXT --
```

```
1 SELECT
2 EDIT
3 DATA
4 STATUS
5 POSITION
6 SYSTEM
7 TEST CYCLE
8 MOVE MENU
9
0 -- NEXT --
```

B.4 CRT/KB KEYS

The correspondence between CRT/KB and teach pendant keys is shown in Table B–3 . You cannot jog the robot from the CRT/KB, so no jog keys exist. Numeric keys on the CRT/KB correspond directly to numeric keys on the teach pendant. Alphabetic keys on the CRT/KB are used for direct alphabetic entry.

Table B–3. Correspondenc e Between Teach Pendant and CRT/KB Keys

Teach Pendant Key	CRT/KB Key
F1, F2, F3, F4, F5	F1, F2, F3, F4, F5
Arrow keys	Cursor keys
SHIFT + UP arrow key(page up)	F7
SHIFT + DOWN arrow key (page down)	F8
ITEM	F6
FCTN	F9
MENUS	F10
-	DO key for KCL*

* For DEC VT-220 terminals only

Appendix C

SYSTEM OPERATIONS

Contents

Appendix C	SYSTEM OPERATIONS	C–1
C.1	STARTUP METHODS	C–2
C.1.1	Overview	C–2
C.1.2	Hot Start	C–3
C.1.3	Cold Start(START COLD)	C–6
C.1.4	Cycle Power	C–9
C.1.5	Controlled Start	C–10
C.1.6	INIT Start	C–15
C.2	HARDWARE DIAGNOSIS	C–16
C.3	CONTROLLER BACKUP AND RESTORE	C–22
C.3.1	Overview	C–22
C.3.2	Backing Up a Controller as Images	C–22
C.3.3	Restoring Controller Images	C–25
C.4	MAINTENANCE OPERATIONS	C–28
C.4.1	Overview	C–28
C.4.2	Updating Boot Software from the Configuration Menu	C–28
C.4.3	Updating *i*Pendant Firmware	C–31
C.5	DIAGNOSTIC TOOLS	C–42
C.5.1	Development Tools	C–42
C.5.2	Diagnostic Log	C–42

C.1 STARTUP METHODS

C.1.1 Overview

The BootROM Monitor provides the following startup methods:

- Hot start
- Cold start
- Cycle power
- Controlled start
- INIT start

> **Warning**
>
> **DO NOT turn on the robot if you discover any problems or potential hazards. Report them immediately. Turning on a robot that does not pass inspection could result in serious injury.**

Table C–1 lists the controller startup methods.

Table C–1. Startup Methods

To perform this operation:	Perform Method 1	or, Perform Method 2
Hot Start (not available for PaintTool) (Only if $SEMIPOWERFL is TRUE)	Turn the power disconnect circuit breaker to ON. **Note** Do not turn off the power disconnect circuit breaker and then turn it on again without waiting at least 3 seconds. Otherwise, the controller might not start up again. If the controller does not start up, turn the power disconnect circuit breaker off, wait at least 5 seconds, and then turn it on again.	None

Table C–1. Startup Methods (Cont'd)

To perform this operation:	Perform Method 1	or, Perform Method 2
Cold start	Press FAULT RESET, and turn the power disconnect circuit breaker to ON. Press SHIFT and RESET and turn the power disconnect circuit breaker to ON. Turn the power disconnect circuit breaker to ON and press the ON button (PaintTool only).	Select Item 2 from the Configuration Menu. To perform a Cold start when the controller is on, enable the teach pendant, press FCTN, and then select CYCLE POWER. Press ENTER, and select YES.
Controlled start	Press PREV, NEXT and turn on the power disconnect circuit breaker.	Select item 3 from the Configuration Menu
Start the Boot Monitor (BMON) Menu	Turn the power disconnect circuit breaker to ON, and press and hold F1 and F5 on the teach pendant. **Note** You must press F1 and F5 quickly after turning on the power disconnect circuit breaker. If the BMON menu does not display, try the operation again.	N/A
INIT start	Select item 3 from the Boot Monitor (BMON) Menu	N/A
Display the Configuration Menu	Press and hold PREV and NEXT, and turn the power disconnect circuit breaker to ON; or, select FCTN:CYCLE POWER and press and hold PREV and NEXT as soon as the BMON screen is displayed.	Select Item 1 from the Boot Monitor (BMON) Menu
Cycle Controller Power	Enable the teach pendant, press FCTN, and then select CYCLE POWER	Press FAULT RESET and turn the power disconnect circuit breaker to OFF, wait 3 seconds, then turn it to ON again.

C.1.2 Hot Start

Hot start is a method for turning on power to the robot and controller without entering Boot Monitor (BMON) functions. *Semi hot start* is active when the system variable $SEMIPOWERFL is set to TRUE. You perform a Hot start by turning the power disconnect circuit breaker to OFF (and $SEMIPOWERFL is set to TRUE). At the completion of a Hot start, the screen that was displayed before power was turned off is displayed.

Use Procedure C-1 to perform a Hot start. Use Procedure C-2 to perform a Hot start from the Configuration Menu.

Procedure C-1 Performing a Hot Start

Conditions

- All personnel and unnecessary equipment are out of the workcell.

> **Warning**
>
> **DO NOT turn on the robot if you discover any problems or potential hazards. Report them immediately. Turning on a robot that does not pass inspection could result in serious injury.**

- The controller is plugged in and is working properly.
- The $SEMIPOWERFL system variable is set to TRUE. Or, you can set the Use Hot START option on the System Configuration Screen to TRUE.
- The REMOTE/LOCAL setup item in the System Configuration Screen is set to LOCAL.
- The teach pendant ON/OFF switch is OFF and the DEADMAN switch is released.

Steps

1. **If the controller is turned on**, press FCTN, select CYCLE POWER, press ENTER, and select YES. **If the controller is turned off**, turn the power disconnect circuit breaker to ON.
2. You will see a screen similar to the following.

```
UTILITIES Hints              JOINT 100%
             ApplicationTool
        Vx.xxx            XXXX/XX
              Custom Vx.xxx
  Copyright xxxx, All Rights Reserved
FANUC LTD, FANUC Robotics America, Inc.
Licensed Software: Your use constitutes
your acceptance. This product protected
by several U.S. patents.
```

Note Do not turn off the power disconnect circuit breaker and then turn it on again without waiting at least 3 seconds. Otherwise, the controller might not start up again. If the controller does not start up, turn the power disconnect circuit breaker off, wait at least 5 seconds, and then turn it on again.

Procedure C-2 Performing a Hot Start from the Configuration Menu

Conditions

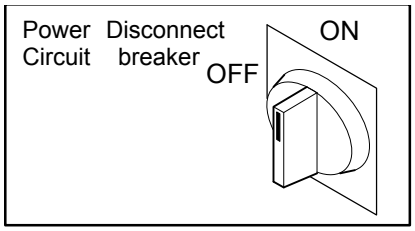
- All personnel and unnecessary equipment are out of the workcell.

> **Warning**
>
> **DO NOT turn on the robot if you discover any problems or potential hazards. Report them immediately. Turning on a robot that does not pass inspection could result in serious injury.**

- The controller is plugged in, is working properly, and is turned on.
- The teach pendant ON/OFF switch is OFF and the DEADMAN switch is released.
- The REMOTE/LOCAL setup item in the System Configuration Menu is set to LOCAL.

Steps

1. **If the controller is turned on**, turn off the power disconnect circuit breaker and wait five seconds. Turn the power disconnect circuit breaker to ON.



   Or, press FCTN, select CYCLE POWER, press ENTER, and select YES.

2. **When the BMON screen is displayed on the teach pendant**, press and hold the PREV and NEXT keys.
3. Release all of the keys.

   You will see a screen similar to the following.

```
---------- CONFIGURATION MENU ----------
1 Hot start
2 Cold start
3 Controlled start
4 Maintenance
Select >
```

4. Select Hot start and press ENTER. After a few seconds, you will see a screen similar to the following.

```
UTILITIES Hints              JOINT 100%
             ApplicationTool
        Vx.xxx            XXXX/XX
              Custom Vx.xxx
  Copyright xxxx, All Rights Reserved
FANUC LTD, FANUC Robotics America, Inc.
Licensed Software: Your use constitutes
your acceptance. This product protected
by several U.S. patents.
```

C.1.3 Cold Start(START COLD)

A *Cold start (START COLD)* is the standard method for turning on power to the robot and controller. If your robot is set up to perform a Hot start, you can force a Cold start. A cold start does the following:

- Initializes changes to system variables
- Initializes changes to I/O setup
- Displays the UTILITIES Hints screen

Note Any programs that were running before the power was turned off will be aborted after a Cold start.

Use Procedure C-3 to perform a Cold start. Use Procedure C-4 to perform a Cold start from the Configuration Menu.

Procedure C-3 Performing a Cold Start

Conditions

- All personnel and unnecessary equipment are out of the workcell.

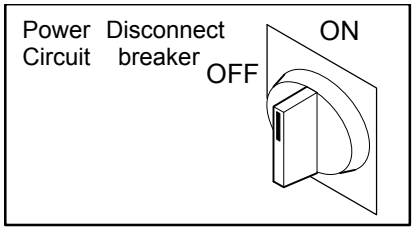
Warning

DO NOT turn on the robot if you discover any problems or potential hazards. Report them immediately. Turning on a robot that does not pass inspection could result in serious injury.

- The controller is plugged in and is working properly.
- The teach pendant ON/OFF switch is OFF and the DEADMAN switch is released.
- The REMOTE/LOCAL setup item in the System Configuration Menu is set to LOCAL.
- The controller is off.

Steps

1. Visually inspect the robot, controller, workcell, and the surrounding area. During the inspection make sure all safeguards are in place and the work envelope is clear of personnel.
2. Turn the power disconnect circuit breaker to ON.



3. If you are using a Paint Robot, press the ON button.
4. **On the teach pendant screen,** you will see a screen similar to the following.

```
UTILITIES Hints              JOINT 100%
             ApplicationTool
         Vx.xxx             XXXX/XX
               Custom Vx.xxx
  Copyright xxxx, All Rights Reserved
FANUC LTD, FANUC Robotics America, Inc.
Licensed Software: Your use constitutes
your acceptance. This product protected
by several U.S. patents.
```

Note Do not turn off the power disconnect circuit breaker and then turn it on again without waiting at least 3 seconds. Otherwise, the controller might not start up again. If the controller

does not start up, turn the power disconnect circuit breaker off, wait at least 5 seconds, and then turn it on again.

Procedure C-4 Performing a Cold Start from the Configuration Menu

Conditions

- All personnel and unnecessary equipment are out of the workcell.

> **Warning**
>
> **DO NOT turn on the robot if you discover any problems or potential hazards. Report them immediately. Turning on a robot that does not pass inspection could result in serious injury.**

- The controller is plugged in and is working properly.
- The REMOTE/LOCAL setup item in the System Configuration Menu is set to LOCAL.
- The teach pendant ON/OFF switch is OFF and the DEADMAN switch is released.

Steps

1. **If the controller is turned on**, press FCTN, select CYCLE POWER, press ENTER, and select YES. **If the controller is turned off**, turn the power disconnect circuit breaker to ON. If you are using a Paint Robot, also press the ON button.
2. **When the BMON screen is displayed on the teach pendant**, press and hold PREV and NEXT.
3. Release all of the keys.

   - **On the teach pendant screen,** you will see a screen similar to the following.

```
---------- CONFIGURATION MENU ----------
1 Hot start
2 Cold start
3 Controlled start
4 Maintenance
Select >
```

4. Select Cold start and press ENTER. When the Cold start is complete, you will see a screen similar to the following.

```
UTILITIES Hints              JOINT 100%
             ApplicationTool
        Vx.xxx            XXXX/XX
              Custom Vx.xxx
  Copyright xxxx, All Rights Reserved
FANUC LTD, FANUC Robotics America, Inc.
Licensed Software: Your use constitutes
your acceptance. This product protected
by several U.S. patents.
```

Note Do not turn off the power disconnect circuit breaker and then turn it on again without waiting at least 3 seconds. Otherwise, the controller might not start up again. If the controller does not start up, turn the power disconnect circuit breaker off, wait at least 5 seconds, and then turn it on again.

C.1.4 Cycle Power

The CYCLE POWER option is available on the FCTN menu after a Cold start. This item toggles the power relay via external I/O. The teach pendant must be enabled for this to work. If the teach pendant is enabled, a prompt box will be displayed asking if you are sure. This is not available on an external robot connection — only from the teach pendant.

Note If Hot Start is enabled (Use Hot Start on the System Configuration Screen is set to TRUE, or $SEMIPOWERFL is set to TRUE) CYCLE POWER will perform a Hot Start.

Use Procedure C-5 to cycle controller power from the FCTN menu.

Procedure C-5 Cycling Controller Power

Conditions

- The teach pendant is enabled.
- You are not using an external robot connection. This is only available on the teach pendant.
- The controller is currently in a Cold start state.

1. Press FCTN.
2. Select CYCLE POWER.
3. Press ENTER. You will see a screen similar to the following.

```
This will cycle power.
Are you sure?
     [NO]      YES
```

4. Use the teach pendant arrow keys to select YES, and press ENTER.

C.1.5 Controlled Start

A *Controlled start (START CTRL)* turns on power to the robot and controller and allows you to do the following:

- Set robot motion parameters
- Execute initial software setup
- Install options and updates
- Load or set system variables

The following tables describe the Controlled start menu items. Use Procedure C-6 to perform a Controlled start.

Caution

The items that appear on the Controlled start screen control how the robot and controller operate. Do not set these items unless you are certain of their effect; otherwise, you could disrupt the normal operation of the robot and controller.

Note If you change any of the items described in the Controlled Start tables, you will need to perform a Cold start for the changes to take effect.

Table C–2. Controlled Start Options for All Software Applications

ITEM	AVAILABLE FUNCTIONS	DESCRIPTION
Controlled Start Menu	You access items using the MENUS and FCTN function keys.	This item displays the Controlled start Menu. Press FCTN and select START (COLD) when you are finished using the Controlled start Menu.

Table C–2. Controlled Start Options for All Software Applications (Cont'd)

ITEM	AVAILABLE FUNCTIONS	DESCRIPTION
MENUS key	S/W Install	This item allows you to install software.
	S/W Version	This item allows you to display information about the current software version.
	Variables	This item allows you to set up robot system variables.
	File	This item allows you to display the FILE menu.
	Alarm	This item allows you to display the ALARM HIST screen.
	Port Init	This item allows you to display the SETUP PORT INIT screen.
	Memory	This item allows you to display the STATUS Memory screen.
	Maintenance	This item allows you to set up robot system variables.
	Program Setup	This item allows you to modify individual software limits related to your program.
	Setup Passwords	This item allows you to set up the password option.
	Host Communications	This item allows you to set up host communication information. Refer to the *Internet Options Setup and Operations Manual* for more information.
	Appl-select	This item allows you to select an application from the Appl selection screen.
	Handling Config	This item allows you to display the Handling Signals screen.
FCTN key	START (COLD)	This item performs a Cold start.
	RESTORE/BACKUP	This item performs a controller backup.
	PRINT SCREEN	This item prints the current screen to a file called TPSCRN.LS on the default device.
	UNSIM ALL I/O	This item unsimulates all I/O settings.

Table C–3. Controlled Start Options for FlexTool

ITEM	AVAILABLE FUNCTIONS	DESCRIPTION
MENUS key	TOOL SETUP	This item performs HandlingTool setup and initialization. You have the following options: **F Number** - This item indicates the robot's F number. **KAREL Prog in select menu** - This item sets the variable $karel_enb . When this item is set to YES, KAREL programs will be displayed on the SELECT menu. When this item is set to NO, KAREL programs will not be displayed on the SELECT menu. Refer to the *FANUC Robotics KAREL Reference Manual* for more information. **Remote Device** - This item sets the system variable $rm t_master, which determines the device that has motion control. This item can be set to the following values: • USER PANEL, where user inputs and outputs are used to run a program and KCL cannot be used to run a program. USER PANEL is the default setting. • KCL, where KCL can be used to run a program but UOP inputs will not start a program. • HOST is not currently used. • NONE is not currently used. Refer to the *KAREL Reference Manual* for more information. **Number of units installed** This item is the number of detected robot libraries that are installed. **Last known HWD Configuration** This item is the last know hardware configuration that the system was running. **Number of units needed** This item tells the system how many are needed.

Table C–4. Controlled Start Options for HandlingTool

ITEM	AVAILABLE FUNCTIONS	DESCRIPTION
MENUS key	TOOL SETUP	This item performs HandlingTool setup and initialization. You have the following options: **F Number** - This item indicates the robot's F number. **KAREL Prog in select menu** - This item sets the variable $karel_enb . When this item is set to YES, KAREL programs will be displayed on the SELECT menu. When this item is set to NO, KAREL programs will not be displayed on the SELECT menu. Refer to the *FANUC Robotics KAREL Reference Manual* for more information. **Remote Device** - This item sets the system variable $rm t_master, which determines the device that has motion control. This item can be set to the following values: • USER PANEL, where user inputs and outputs are used to run a program and KCL cannot be used to run a program. USER PANEL is the default setting. • KCL, where KCL can be used to run a program but UOP inputs will not start a program. • HOST is not currently used. • NONE is not currently used. Refer to the *KAREL Reference Manual* for more information. **Run GM Setup Wizard** — This item, if set to YES, will run the program SWWIZARD.PC if it exists on the controller. This program sets up the GM Gripper menus and macros, Disable Faults menu, and the Alarm Recovery panel. **Intrinsically Safe TP** - This item sets the teach pendant to be intrinsic. The teach pendant LEDs are shut off and appear as circles on the left most column of the teach pendant screen. Intrinsically safe teach pendants are used in hazardous environments where electrical signals need to be minimized.

Table C–5. Handling Config Menu Options

Item	Description
Number of MH Tools	Use this item to configure the number of tools you need for your system. The maximum number of tools is 29. At cold start, you will then have the specified number of tools available in the MH Valves menu under Setup.
Number of Valves Per Tool	Use this item to configure the number of valves you have in your tool. The maximum number of valves is 29. At cold start, you will then have the specified number of valves available in the **MH Valves** menu under Setup. You can then configure any valve to be a vacuum or clamp style tool and set up the required I/O for each.
Number of Clamps	Use this item to configure the number of clamps you need for your tool. The maximum number of clamps is 64. At Cold start, you will then have the specified number of clamps available in the MH Valves menu under Setup to assign to a valve.

Table C–5. Handling Config Menu Options (Cont'd)

Item	Description
Number of Part Present Inputs	Use this item to configure the number of Part Present Inputs you need for your tool. the maximum number of Part Present Inputs is 29. At cold start, you will then have the specified number of Part Present Inputs available in the MH Valves menu under Setup to assign to a valve.
Number of Vacuum Inputs	Use this item to configure the number of Vacuum Inputs you need for your tool. the maximum number of Vacuum Inputs is 29. At cold start, you will then have the specified number of Vacuum Inputs available in the MH Valves menu under Setup to assign to a valve.
Use Active Tool for Arguments	Use this item to change the behavior of the tool macros when no parameters are passed to them. When this is set to YES, the number of valves set in the tool SETUP are used to operate that number of valves for that tool. When set to NO, they act like the original specification.

Procedure C-6 Performing a Controlled Start

Conditions

- All personnel and unnecessary equipment are out of the workcell.

> **Warning**
>
> **DO NOT turn on the robot if you discover any problems or potential hazards. Report them immediately. Turning on a robot that does not pass inspection could result in serious injury.**

- The controller is plugged in and is working properly.
- The teach pendant ON/OFF switch is OFF and the DEADMAN switch is released.
- The REMOTE/LOCAL setup item in the System Configuration Menu is set to LOCAL.

Steps

1. **If the controller is turned on**, press FCTN, select CYCLE POWER, press ENTER, and select YES. **If the controller is turned off**, turn the power disconnect circuit breaker to ON. If you are using a Paint Robot, also press the ON button.
2. When the BMON screen is displayed on the teach pendant, press and hold the PREV and NEXT keys. You will see a screen similar to the following.

```
---------- CONFIGURATION MENU ----------
1 Hot start
2 Cold start
3 Controlled start
4 Maintenance
Select >
```

3. Release all of the keys.
4. Select Controlled start and press ENTER.
5. **When you are finished** with the Controlled start:
   a. **If you want to set configuration items**, refer to Table C–2 for more information on setting these items. When you are finished, press FCTN, select CYCLE POWER, press ENTER, and select YES.
   b. **If you want to operate the robot**, perform a Cold start. Press FCTN select CYCLE POWER, press ENTER, and select YES.
   c. **If you want to load system variable files**, press MENUS, select FILE, and load .SV or .VR files.
   d. **If you have installed an update**,
      - Perform a Cold start.
      - When the Cold start has completed, press FCTN, select CYCLE POWER, press ENTER, and select YES for the changes to take effect.

C.1.6 INIT Start

An initialized (INIT) start occurs automatically as the first phase of software installation.

Note This step must be completed before you perform a Controlled start.



Caution

Do not use INIT start to start the controller. An INIT start erases all information stored in the saved memory pools.

C.2 HARDWARE DIAGNOSIS

Hardware diagnosis allows you to display memory pool sizes, dump memory, and write memory.

Table C–6 lists the hardware diagnostic functions. Use Procedure C-7 to perform hardware diagnosis functions.

Table C–6. Hardware Diagnostic Functions

FUNCTION	DESCRIPTION
Show size of RAM/ROM Modules	This function lists the sizes of the FROM, DRAM, and SRAM modules currently in the controller.
Show list of F-BUS Modules	This function displays a list of F-BUS modules currently in the controller. This shows a list of both standard and optional hardware in the controller.
Dump Memory	This function dumps a list of controller memory information and displays it on the teach pendant screen.
Write Memory	This function allows you to change the contents of controller memory
Check SRAM Memory	This function allows you to display the controller ECC (error correcting C-MOS) status.
Clear Vision SRAM Memory	This function clears the static RAM on a Vision CPU. This function is for Vision systems only.
Check FROM Memory	This function checks FROM memory for bad blocks. If bad blocks are found, you must reload the controller from an image or from a backup.

Procedure C-7 Performing Hardware Diagnosis Functions

Conditions

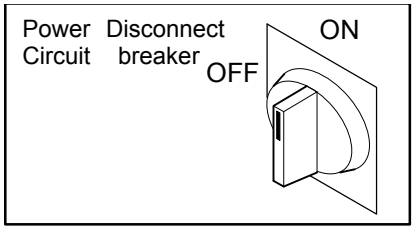
- All personnel and unnecessary equipment are out of the workcell.

> **Warning**
>
> **DO NOT turn on the robot if you discover any problems or potential hazards. Report them immediately. Turning on a robot that does not pass inspection could result in serious injury.**

- The controller is plugged in and is working properly.
- The teach pendant ON/OFF switch is OFF and the DEADMAN switch is released.
- The REMOTE/LOCAL setup item in the System Configuration Menu is set to LOCAL.

Steps

1. **If the controller is turned on,** turn the power disconnect circuit breaker to OFF.
2. Press and hold the F1 and F5 keys on the teach pendant and turn the power disconnect circuit breaker to ON.



You will see a screen similar to the following.

```
Base System Version V7.xxxx (FRNA)
Initializing file devices ... done.
******* BMON MENU *******
1.    Configuration Menu
2.    All software installation (MC:)
3.    INIT start
4.    Controller backup/restore
5.    Hardware diagnosis
6.    Maintenance
7.    All software installation (ETHERNET)
8.    All software installation (USB)
Select:
```

3. Release all of the keys.
4. Select Hardware diagnosis. You will see a screen similar to the following.

```
***BOOT MONITOR for CONTROLLER***
Base System Version V7.xxxx (FRNA)
Select: 5
  ****** Hardware Diagnosis Menu ******
1. Show size of RAM/ROM Modules
2. Show list of F-BUS Modules
3. Dump Memory
4. Write Memory
5. Check SRAM Memory
6. Clear Vision SRAM Memory
9. Return to Main Menu
Select : _
```

5. **To display the size of RAM/ROM modules**, select Show size of RAM/ROM Modules. You will see a screen similar to the following.

```
***BOOT MONITOR for CONTROLLER***
Base System Version V7.xxxx (FRNA)
FROM: 16Mb   DRAM: 16Mb   SRAM: 2Mb
   ****** Hardware Diagnosis Menu ******
1. Show size of RAM/ROM Modules
2. Show list of F-BUS Modules
3. Dump Memory
9. Return to Main Menu
```

6. **To display a list of S-BUS modules**, select Show list of S-BUS Modules. You will see a screen similar to the following.

```
***BOOT MONITOR for CONTROLLER***
Base Version V7.xxxx (FRNA)
   0 04   5    LIME
   1 03   3    ETCC
   2 01   1    QFS
   3 04   1    LIME
  SS-Bus Modules:
 ASIC ID   REV  Name
    1 05   1    NBG
    2 05   1    NBG
    3 05   1    NBG
Press ENTER key _
```

7. To dump controller memory to the screen,

a. Select Dump Memory. You will see a screen similar to the following.

```
***BOOT MONITOR for CONTROLLER***
Base System Version V7.xxxx (FRNA)
Next = [ENTER], Quit=[.][ENTER]
Input Address (HEX):
```



Caution

The input address you type in Step 7b must be a valid address. If not, the Boot Monitor will be corrupted and you will have to repeat Step 1 through Step 4 of this procedure.

b. Type the input address and press ENTER. If you do not know the input address, contact your FANUC Robotics representative. You will see a screen similar to the following.

```
***BOOT MONITOR for CONTROLLER***
Base System Version V7.xxxx (FRNA)
Next = [ENTER], QUIT=[.][ENTER]
Input Address [HEX]:
400200  <-- User Input

00400200  7C  00  04  AC  4C00  01  2C
00400208  7C  00  04  AC  4C00  01  2C
00400210  7C  00  04  AC  4C00  01  2C
00400218  7C  00  04  AC  4C00  01  2C
00400220  7C  00  04  AC  4C00  01  2C
00400228  7C  00  04  AC  4C00  01  2C
00400230  7C  00  04  AC  4C00  01  2C
00400238  7C  00  04  AC  4C00  01  2C
Next = [ENTER], QUIT=[.][ENTER]
Input Address [HEX
```

Note If the controller hangs up or halts during a Dump Memory, this can indicate problems with

- The entered address
- Memory hardware

If this occurs, contact a FANUC Robotics Service Representative.

8. **To display the next screen**, press ENTER.
9. **To quit** and display the Hardware Diagnosis screen, press . (period) and then ENTER.



Caution

Do not use the write memory function unless it is absolutely necessary. The write memory function can write data in the controller and could corrupt the controller software if it is not done correctly. Contact your FANUC Robotics representative for more information.

To write memory,

a. Select Write Memory. You will see a screen similar to the following.

```
****** Hardware Diagnosis Menu ******
Data size [BYTE=0/WORD=1/LONG=2] : 0
Input Address(HEX) : 02000000
Current value is 81
input new value :
```

Note To quit without making any changes, press ENTER.

b. Type the new value and press ENTER. You will see a screen similar to the following.

```
****** Hardware Diagnosis Menu ******
Data size [BYTE=0/WORD=1/LONG=2] : 0
Input Address(HEX) : 02000000
Current value is 81
Input new value :
old value is 81
new value is 20
Press ENTER key
```

10. **To Check the SRAM Memory**,

> **Caution**
>
> Performing this step will require a full load or an image restore of your controller. Be sure you have performed an image backup, or backed up your entire controller, before you continue.

a. Select Check SRAM Memory. You will see a screen similar to the following.

```
****** Hardware Diagnosis Menu ******
  005010120000:01 -> 0000:00
  005033201234:56 -> 1235:56
  00510024789A:BC -> 789:BD
Total correction: 3
Press ENTER key
```

b. Press ENTER to display the Hardware Diagnosis menu.

Note The information displayed on the screen means that errors exist at address 00501012, 00503320, and 00510024. The message xxx:yy means that the SRAM data value is xxxx and the ECC code value is yy.

For example, 1234:56 means that the data is 1234 and the ECC check bits are 56. The message 0000:01 -> 0000:00 means that the error bit exists at the least significant bit of ECC and this bit is corrected to 0. If there are several errors on SRAM, only the first ten locations will be displayed.

11. To Check the FROM Memory,

a. Select Check FROM Memory. When it is finished, you will see a screen similar to the following.

```
Total bad block: 0
Press ENTER key
```

b. **If no bad blocks are found**, press ENTER.

If bad blocks are found, you must reload the controller from an image or a backup.

C.3 CONTROLLER BACKUP AND RESTORE

C.3.1 Overview

Refer to Table C–7 for description of each restore controller function.

Table C–7. Backup and Restore Controller Items

ITEM	DESCRIPTION
Emergency Backup	This item is not yet available.
Backup Controller as Images	This item allows you to back up the controller as memory images.
Restore Controller Images	This item allows you to restore a controller from memory images performed using the Backup Controller as Images function.
Bootstrap to Cfg Menu	Refer to the *Internet Options Setup and Operations Manual* for more information.

C.3.2 Backing Up a Controller as Images

This feature allows you to back up a controller with minimal system interaction. It will make a memory image of the F-ROM and S-RAM memory controller modules.

Procedure C-8 Backing up a Controller as Images

Conditions

- All personnel and unnecessary equipment are out of the workcell.

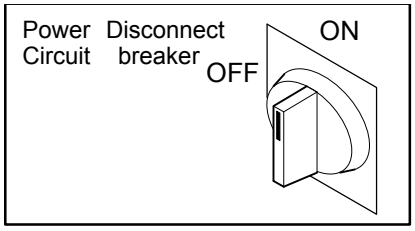
> **Warning**
>
> **DO NOT turn on the robot if you discover any problems or potential hazards. Report them immediately. Turning on a robot that does not pass inspection could result in serious injury.**

- The controller is plugged in and is working properly.
- The teach pendant ON/OFF switch is OFF and the DEADMAN switch is released.
- The REMOTE/LOCAL setup item in the System Configuration Menu is set to LOCAL.
- You have a memory card that has enough free space.

Note You should use a memory card with enough free space to hold the full controller image files. For example, a 16MB F-ROM and 2MB S-RAM configuration will require a card with more than 18MB of free space and has been formatted using the controller.

Steps

1. **If the controller is turned on,** turn the power disconnect circuit breaker to OFF.
2. Press and hold the F1 and F5 keys on the teach pendant.
3. Turn the power disconnect circuit breaker to ON. You will see a screen similar to the following.




```
Base System Version V7.xxxx (FRNA)
Initializing file devices ... done.
******* BMON MENU *******
1.    Configuration Menu
2.    All software installation (MC:)
3.    INIT start
4.    Controller backup/restore
5.    Hardware diagnosis
6.    Maintenance
7.    All software installation (ETHERNET)
8.    All software installation (USB)
Select:
```

4. Release all of the keys.
5. Select Controller backup/restore and press ENTER. You will see a screen similar to the following.

```
***BOOT MONITOR for CONTROLLER***
**** BACKUP/RESTORE MENU ****
0.   Return To Main Menu
1.   Emergency Backup
2.   Backup Controller As Images
3.   Restore Controller Images
4.   Bootstrap To Cfg Menu
Select:
```

6. Select Backup controller as Images and press ENTER. You will see a screen similar to the following.

```
***BOOT MONITOR for CONTROLLER***
** Device Selection **
1. Memory card (MC:)
2. Ethernet (TFTP:)
3. USB (UD1:)
```

7. Select Memory card (MC:) or USB (UD1:) and press ENTER. You will see a screen similar to the following.

```
***BOOT MONITOR for CONTROLLER***
**** BACKUP Controller as Images ****
     Module size to backup:
     FROM: 16 Mb  SRAM: 2 Mb
     Please Insert MC: which has at least
     1Mb free space.
     Are you ready?[Y=1/N=else]:
```

8. Place a memory card enough free space to hold the full controller image files. For example, a 16MB F-ROM and 2MB S-RAM configuration will require a card with 18MB of free space.

9. Type 1 and press ENTER. The memory files will be written to the memory card. You should see messages similar to the following.

```
Writing MC:\FROM00.IMG (1/18)
Writing MC:\FROM01.IMG (2/18)
Writing MC:\FROM02.IMG (3/18)
 . . .
Writing MC:\SRAM0x.IMG (18/18)
```

Note If BMON crashes or hangs during "Writing MC:\SRAM0x.img...," this could be an indication of problems with the SRAM memory. If this occurs, contact a FANUC Robotics Service Representative.

Note If the memory card does not have enough free space for the next image file, you will need to insert a new memory card. If this occurs, you will see the following message:

```
**Not enough free space on MC:
  CHANGE card and Press ENTER>
```

10. When all the image files have been written to the memory card, you will see a message similar to the following.

```
DONE!!
Press ENTER to return>
```

11. Press ENTER to display the main BMON menu.



Caution

Be sure to label this memory card properly so that you can restore it to the correct configuration. Otherwise, if you restore an incorrect configuration you will lose valuable data and possibly corrupt the controller.

C.3.3 Restoring Controller Images

This feature allows you to restore the F-ROM and S-RAM controller memory from an image backup. It can only be restored to a controller with the same memory configuration, or larger.

Procedure C-9 Restoring Controller Images

Conditions

- All personnel and unnecessary equipment are out of the workcell.



Warning

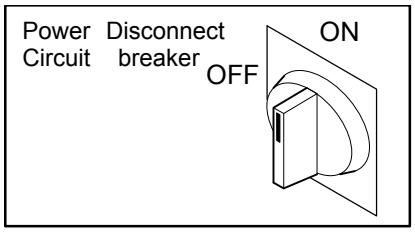
DO NOT turn on the robot if you discover any problems or potential hazards. Report them immediately. Turning on a robot that does not pass inspection could result in serious injury.

- The controller is plugged in and is working properly.
- The teach pendant ON/OFF switch is OFF and the DEADMAN switch is released.
- The REMOTE/LOCAL setup item in the System Configuration Menu is set to LOCAL.
- You have a memory card that has the controller images backed up in Procedure C-8 .

> **Caution**
>
> A restore from a controller backup will be unsuccessful if the memory configuration of the restored CPU is different from the backup CPU.

Steps

1. **If the controller is turned on,** turn off the power disconnect circuit breaker.
2. Press and hold the F1 and F5 keys on the teach pendant.
3. Turn the power disconnect circuit breaker to ON. You will see a screen similar to the following.




```
Base System Version V7.xxxx (FRNA)
Initializing file devices ... done.
******* BMON MENU *******
1.    Configuration Menu
2.    All software installation (MC:)
3.    INIT start
4.    Controller backup/restore
5.    Hardware diagnosis
6.    Maintenance
7.    All software installation (ETHERNET)
8.    All software installation (USB)
Select:
```

4. Release all of the keys.
5. Select Controller backup/restore and press ENTER. You will see a screen similar to the following.

```
***BOOT MONITOR for CONTROLLER***
**** BACKUP/RESTORE MENU ****
0.   Return To Main Menu
1.   Emergency Backup
2.   Backup Controller As Images
3.   Restore Controller Images
4.   Bootstrap To Cfg Menu
Select:
```

6. Select Restore Controller Images and press ENTER. You will see a screen similar to the following.

```
***BOOT MONITOR for CONTROLLER***
** Device Selection **
1. Memory card (MC:)
2. Ethernet (TFTP:)
3. USB (UD1:)
```

7. Select Memory card (MC:) or USB (UD1:) and press ENTER. You will see a screen similar to the following.

```
***BOOT MONITOR for CONTROLLER***
**** RESTORE Controller Images ****
Module size to backup:
     FROM: 16 Mb  SRAM: 2 Mb
     CAUTION: You SHOULD have image files
     From the same size of FROM/SRAM.
     If you don't, this operation will cause
     Fatal damage to the controller.
     Are you ready?[Y=1/N=else]:
```

8. Place a memory card with the controller images backed up in Procedure C-8

9. Type 1 and press ENTER. The memory files will be restored from the memory card. You should see the following messages:

```
Clearing FROM (total 16 Mb)

1M & 2M area ... done

2M & 4M area ... done
```

```
. . .

Reading MC:\SRAM01.IMG ... Done

-- Restore complete --

Press ENTER to return>
```

10. Press ENTER to display the main BMON menu.
11. Turn the controller off and then on again to accept the new image information.

C.4 MAINTENANCE OPERATIONS

C.4.1 Overview

Maintenance operations include the following operations:

- Update boot software
- Development
- Ethernet based loading (Refer to the *Internet Options Setup and Operations Manual* for more information)
- Upgrade iPendant software

Note Making a Boot Memory Card is for FANUC Robotics internal use only.

C.4.2 Updating Boot Software from the Configuration Menu

The BootROM Monitor (BMON) software version must be compatible with the version of loaded system software. Use Procedure C-10 to update Boot Software from the Configuration Menu if your version of Boot software is incorrect. The controller system software checks the BMON version at every Controlled start to Cold start and will automatically update the BMON version if necessary.

Caution

If your version of Boot software is incompatible with the current version of system software, your controller could become corrupted. You should only update the version of Boot software in your controller if you have been instructed to do so by a FANUC Robotics representative or if you are doing it as part of a software update.

Caution

Do not turn off controller power while the BMON code is being updated. Otherwise, you controller could become corrupted.

Note Refer to the SSD.html file on the memory card for information on the BMON version that is compatible with your software.

Procedure C-10 Manually Updating BMON FROM

Note Updating BMON F-ROM takes approximately 30 seconds.

Conditions

- The BMON F-ROM chip is not a valid version.
- The controller is capable of running and has valid loaded software.

Steps

1. **If the controller is turned on**, turn off the power disconnect circuit breaker.
2. **On the teach pendant**, press and hold the PREV and NEXT keys.
3. Turn the power disconnect circuit breaker to ON. You will see a screen similar to the following.

```
----------- CONFIGURATION MENU ---------
1 Hot start
2 Cold start
3 Controlled start
4 Maintenance
Select >
```

4. Release all of the keys.
5. Select Maintenance and press ENTER. You will see a screen similar to the following.

```
---------- Maintenance Menu -----------
0. Top menu
1. Update boot software
2. Development
3. Upgrade iPendant software
Select >
```

6. Select Update boot software and press ENTER. You will see a screen similar to the following.

```
Select > 4
Select function.
0. Cancel (top menu)
1. Write boot ROM directly
Select >
```



Caution

Do not turn off controller power while the BMON code is being updated. Otherwise, you controller could become corrupted.

7. Select Write boot ROM directly and press ENTER. You will see a screen similar to the following.

```
Select > 1
Select source device.
0. Cancel (top menu)
1. Memory card (MC:)
2. USB (UD1:)
3. System flash file (FRS:)
Select >
```

8.

Note In this step, carefully wait until the process is complete! Do not power the controller off until it is finished.

If the Boot ROM load module is installed in the controller select System flash file, and press ENTER. **If the Boot ROM load module is on a memory card** select Memory Card or Memory Stick, and press ENTER. Carefully wait until the process is complete! Do not power the controller off until this process is complete. The screen will indicate: "Reading BOOTROM.LD0". Then it will indicate: Writing Boot FROM, please wait". Finally the "Maintenance Menu" will be displayed. The FROM is now updated. Be sure to wait until all of these occur before proceeding.

9. Type 0 and press ENTER to display the top level Configuration Menu.
10. Select a start mode. Refer to Section C.1 .

C.4.3 Updating *i*Pendant Firmware

The i Pendant Firmware version must be compatible with the version of loaded controller software. The correct version of the *i*Pendant firmware is distributed on the same memory card as the controller software.

If an *i*Pendant is attached to the system , the *i*Pendant firmware version will be checked automatically at the end of the controller software loading process when the controller transitions from Controlled Start to Cold Start. You will be given the opportunity to update the firmware at that point or you can wait until a later time. Refer to Section C.4.3.1 for additional details on the Automatic process or Section C.4.3.2 for details on Manually updating the *i* Pendant Firmware.



Caution

If your version of *i* Pendant firmware is incompatible with the current version of system software, your *i*Pendant operation might become erratic.

C.4.3.1 Automatic Update of *i* Pendant Firmware

If an *i*Pendant is attached to the system, the *i*Pendant firmware version will be checked automatically when the controller transitions from Controlled Start to Cold Start at the end of the software installation process. The following procedure details the steps in this process.

Note The automatic update of the *i* Pendant firmware takes approximately 10 minutes.

Procedure C-11 Automatic *i*Pendant Firmware Update

Conditions

- Controller is at Controlled start.
- An *i*Pendant is attached to the system.

- The system has determined that the *i*Pendant firmware version is incompatible with the current controller software version.

> **Warning**
>
> **DO NOT TURN OFF THE CONTROLLER DURING THIS PROCESS. Turning off the controller during the update process could cause the iPendant to become inoperable, requiring you to return it to FANUC Robotics for service.**

Steps

1. When you select FCTN >COLD to Cold Start the controller, the controller verifies that updated firmware files exist on either the MC: or FRS: device. If the files do not exist, a screen similar to the following is displayed:

```
--------------------------------------
A check of iPendant firmware found:
   System   version: xxxxx
   iPendant version: xxxxx
iPendant firmware should be upgraded,
but necessary files are missing
please contact FANUC Robotics.
```

   Where xxxxx is the version numbers of the controller software and the *i*Pendant firmware that were found.

   You will need to obtain the necessary files or media and manually update the *i*Pendant firmware following the procedure in Section C.4.3.2

   Select F4, CONTINUE to continue with the Cold Start process.

2. If the appropriate files are found, a screen similar to the following will be displayed:

```
---------- iPendant Firmware ----------
   System   version: x.xxxx
   iPendant version: x.xxxx
iPendant firmware should be upgraded.
If you wish to upgrade, please power
off and on the robot controller.
  0. Continue without upgrade
Select >
--------------------------------------
```

3. To continue with the upgrade, power the controller off and then back on. To cancel the upgrade and continue with the Cold Start process, type 0, then ENTER. If you select CANCEL then you will need to manually update the *i*Pendant firmware following the procedure in Section C.4.3.2 .

4. If you are going to upgrade the firmware then a screen similar to the following will be displayed when you power the controller back on.

```
---------- iPendant Firmware ----------
   System   version: x.xxxx
   iPendant version: x.xxxx
iPendant firmware should be upgraded.
Do you wish to upgrade?
  0. Cancel
  1. Upgrade firmware
Select >
---------------------------------------
```

Selecting CANCEL will continue the Cold Start process without upgrading the iPendant firmware. You will need to manually update the *i* Pendant firmware following the procedure in Section C.4.3.2

5. Type 1, and press ENTER to upgrade the firmware. A screen similar to the following will be displayed.

```
 ---------- iPendant Firmware ----------
 DO NOT POWER OFF
  Warning Ipendant firware updating
  takes a long time.
DO NOT POWER OFF
OK=[1], Cancel=[Other] >
---------------------------------------
```

6. Type 1, and press ENTER to continue with the firmware upgrade. The *i*Pendant firmware update process will begin.

During the upgrade process you will see a RED status line in the lower left corner of the *i*Pendant screen, indicating the status of the update process.

7. After the update is complete the following screen will be displayed.

```
---------- iPendant Firmware ----------
iPendant firmware has been upgraded.
Please power off and on the robot
controller to make it effective.
  0. Continue with older firmware
     until next powerup.
Select >
---------------------------------------
```

If you type 0 and then press ENTER, the controller will continue through the Cold start process. The *i*Pendant will continue to operate with the older firmware until the next time the controller is turned off then on again.

8. To activate the new firmware, turn the controller off then on. When the *i*Pendant first turns on you will see the updated firmware version displayed at the bottom of the screen.

C.4.3.2 Manually Updating the *i*Pendant Firmware

The following sections detail the procedures to update and/or backup the *i* Pendant firmware manually. The available options include:

- Updating *i*Pendant Application Firmware (Do Not Use)
- Updating Complete *i*Pendant Firmware
- Creating an *i*Pendant Firmware Backup

Note An additional option, "Create bootable memory card," is for FANUC Robotics internal use only.

The manual updating of the *i*Pendant firmware can be done either from a memory card or over the Ethernet. The procedures outlined here are for updating from a Memory Card. Updating from Ethernet is similar, however it requires that the robot and network are properly configured for software loading. Refer to the *Internet Options Manual* for more information on how to set up the server and controller.

C.4.3.2.1 Updating *i*Pendant Application Firmware

> **Caution**
>
> This option is only available for *i* Pendants with V6.22xx or greater firmware already installed. In addition, the controller software version must be at least V6.22xx. Any earlier version of the *i* Pendant requires a complete firmware update. If you select this option by mistake, type a 0 and press ENTER to return to the Configuration Menu.

> **Caution**
>
> Do not perform this procedure unless you are specifically instructed to do so by a qualified FANUC Robotics Service Technician, or other representative. Otherwise, you could corrupt your software.

Note Updating the *i*Pendant Application firmware takes approximately 2 minutes.

Procedure C-12 Update Application Code

Conditions

- The firmware currently installed on the *i* Pendant is at least version V6.22xx.
- The controller is capable of running, and has valid loaded software.
- The*i*Pendant is connected and capable of running.
- The memory card containing the controller system software and *i*Pendant firmware is properly inserted in the memory card slot on the controller.
- You are sure that the existing *i*Pendant firmware is capable of application update
- You have been directed to perform this procedure by a FANUC Robotics representative.

Note Use Procedure C-13 instead of Procedure C-12 in most situations.

Steps

1. **If the controller is turned on**, turn the power disconnect circuit breaker to OFF.
2. **On the teach pendant**, press and hold the PREV and NEXT keys.
3. Turn the power disconnect circuit breaker to ON. You will see a screen similar to the following.

```
----------- CONFIGURATION MENU ---------
1 Hot start
2 Cold start
3 Controlled start
4 Maintenance
Select >
```

4. Release all of the keys.
5. Select Maintenance and press ENTER. You will see a screen similar to the following.

```
---------- Maintenance Menu -----------
0. Top menu
1. Update boot software
2. Development
3. Ethernet based loading
4. Upgrade iPendant software
Select >
```

6. Select Upgrade iPendant software. You will see a screen similar to the following.

```
---------- iPendant Menu -----------
0. Cancel (top menu)
1. Update application code
2. Update whole firmware
3. Firmware backup
4. Create bootable memory card
Select >
```

7. Select Update application code and press ENTER. You will see a screen similar to the following.

```
Select > 1
You need much time to update.
You should not power off the controller
during updating.
OK=[1], Cancel=[Other]>
```

8. Type 1, and press ENTER to continue. You will see a screen similar to the following:

```
Select source device.
0. Cancel
1. Memory Card (MC:)
2. System flash file (FRS:)
3. Ethernet (TFTP:)
Select >
```

9. Select Memory Card.

The *i* Pendant Application firmware update process will begin. During the process you will see messages displayed in the lower left corner of the screen as the updating proceeds. When the update is complete, you will be returned to the top level Configuration menu.



Warning

DO NOT TURN OFF THE CONTROLLER DURING THIS PROCESS. Turning off the controller during the update process could cause the *i*Pendant to become inoperable, requiring you to return it to FANUC Robotics for service.

10. When the Configuration menu is displayed, after the update is complete, turn off the power disconnect circuit breaker.

11. Turn on the power disconnect circuit breaker again to activate the new application firmware.

C.4.3.2.2 Updating Complete *i* Pendant Firmware

Note Updating the complete *i* Pendant firmware takes approximately 10 minutes.

Procedure C-13 Update Complete Firmware

Conditions

- The *i* Pendant firmware is incompatible with the current version of the controller software.
- The controller is capable of running and has valid loaded software.
- The *i*Pendant is connected and capable of running.
- The memory card containing the controller system software is properly inserted in the memory card slot on the controller.

Steps

1. **If the controller is turned on**, turn the power disconnect circuit breaker to OFF.
2. **On the teach pendant**, press and hold the PREV and NEXT keys.
3. While you hold these keys, turn the power disconnect circuit breaker to ON. You will see a screen similar to the following.

```
----------- CONFIGURATION MENU ---------
1 Hot start
2 Cold start
3 Controlled start
4 Maintenance
Select >
```

4. Release all of the keys.
5. Select Maintenance and press ENTER. You will see a screen similar to the following.

```
---------- Maintenance Menu -----------
0. Top menu
1. Update boot software
2. Development
3. Ethernet based loading
4. Upgrade iPendant software
Select >
```

6. Select Upgrade iPendant software. You will see a screen similar to the following.

```
---------- iPendant Menu -----------
0. Cancel (top menu)
1. Update application code
2. Update whole firmware
3. Firmware backup
4. Create bootable memory card
Select >
```

7. Select Update whole firmware and press ENTER. You will see a screen similar to the following.

```
Select >2
You need much time to update.
You should not power off the controller
during updating.
OK=[1], Cancel=[Other]>
```

8. Type 1 and press ENTER. You will see a screen similar to the following:

```
Select source device.
0. Cancel
1. Memory Card (MC:)
3. Ethernet (TFTP:)
Select >
```

9. Select Memory Card and press ENTER.

The *i* Pendant firmware update process will begin. During the process you will see messages displayed in the lower left corner of the screen as the updating proceeds.



Warning

DO NOT TURN OFF THE CONTROLLER DURING THIS PROCESS. Turning off the controller during the update process could cause the *i* Pendant to become inoperable, requiring you to return it to FANUC Robotics for service.

10. After the update is complete the following screen will be displayed.

```
---------- iPendant Firmware ----------
iPendant firmware has been upgraded.
Please power off and on the robot
controller to make it effective.
  0. Continue with older firmware
     until next powerup.
Select >
---------------------------------------
```

If you type 0 and then press ENTER, the controller will continue through the Cold start process. The *i* Pendant will continue to operate with the older firmware until the next time you turn off the controller then turn it on again.

11. To activate the new firmware, turn the controller off then on. When the *i* Pendant first turns on you will see the updated firmware version displayed at the bottom of the screen.

C.4.3.2.3 Creating an *i* Pendant Firmware Backup

Backing up the iPendant Firmware is not normally necessary since the firmware is included on the Controller software memory card. However if required, the firmware from an operational *i* Pendant can be backed up to a PCMCIA Flash Memory card and can be loaded to an *i* Pendant using the procedure outlined in Procedure C-13 .

Note Backing up the *i*Pendant firmware takes approximately 15 minutes.

Procedure C-14 Firmware Backup

Conditions

- The controller is capable of running and has valid loaded software.
- The *i*Pendant is connected and capable of running.
- An PCMCIA Flash Memory Card is available with at least 5MB of free space.
- You want to back up the *i* Pendant firmware to be used later in case a reload is necessary.

Steps

1. **If the controller is turned on** , turn the power disconnect circuit breaker to OFF.
2. Plug the blank PCMCIA Memory Card into the slot in the controller.
3. **On the teach pendant**, press and hold the PREV and NEXT keys.
4. Turn the power disconnect circuit breaker to ON. You will see a screen similar to the following.

```
----------- CONFIGURATION MENU ---------
1 Hot start
2 Cold start
3 Controlled start
4 Maintenance
Select >
```

5. Release all of the keys.
6. Select Maintenance and press ENTER. You will see a screen similar to the following.

```
---------- Maintenance Menu -----------
0. Top menu
1. Update boot software
2. Development
3. Ethernet based loading
4. Upgrade iPendant software
Select >
```

7. Select Upgrade iPendant software. You will see a screen similar to the following.

```
---------- iPendant Menu -----------
0. Cancel (top menu)
1. Update application code
2. Update whole firmware
3. Firmware backup
4. Create bootable memory card
Select >
```

8. Select Firmware backup and press ENTER. You will see a screen similar to the following.

```
Select >3
You need much time to save the contents
of iPendant firmware.
OK=[1], Cancel=[Other]>
```

9. Type 1, and press ENTER. The *i* Pendant firmware backup process will begin. This process takes about 45 minutes. During the process you will see messages displayed in the lower left corner of the screen as the backup proceeds. When the backup is complete, you will be returned to the top level Configuration Menu.
10. Select a start mode. Refer to Section C.1 .

C.5 DIAGNOSTIC TOOLS

C.5.1 Development Tools

Development Tools are special functions used by a FANUC Robotics engineer to examine the details of the running system. They are used to control the timing to help diagnose problems. Some of these functions include:

> **Caution**
>
> Do not use any of these development functions unless explicitly instructed to do so by a FANUC Robotics representative. Otherwise, you could corrupt your system.

- The Step by step startup is used to control the start sequence in order to examine the system at various stages.
- The Target monitor is used to communicate to an external PC to examine the internal details of the system or a particular function.
- The Memory protect function is used to disable some of the memory management functions of the hardware and operating system.
- The Set Port 1 for Debug Consol function resets the debug console back to Port P2:.
- The Delete All DT Files function is used to delete certain system level .DT (data) files from the controller for diagnostic and repair purposes.

C.5.2 Diagnostic Log

If you need additional information to help you solve a specific problem, and you are instructed to do so by a FANUC Robotics software support representative, at Cold start press FCTN and select Diagnostic Log.

Appendix D

MASTERING

Contents

Appendix D	MASTERING	D–1
D.1	OVERVIEW	D–2
D.2	RESETTING ALARMS AND PREPARING FOR MASTERING	D–4
D.3	SAVING AND RESTORING MASTERING DATA	D–6
D.4	MASTERING TO A FIXTURE (FIXTURE POSITION MASTER)	D–8
D.5	SINGLE AXIS MASTERING	D–11
D.6	QUICK MASTERING	D–15
D.7	ZERO DEGREE MASTERING	D–19

D.1 OVERVIEW

When you *master* a robot you define the physical location of the robot by synchronizing the mechanical information with the robot's positional information. A robot must be mastered to operate properly. Robots are usually mastered before they leave FANUC Robotics. However, it is possible that a robot might lose its mastering data and need to be remastered.

The robot axes are controlled by a closed loop servo system. The controller outputs a command signal to drive each motor. A feedback device mounted on the motor, called a *serial pulse coder,* sends a signal back to the controller. During robot operation, the controller constantly analyzes the feedback signal, and modifies the command signal to maintain proper location and velocity of the end effector at all times.

For the robot to move accurately to recorded positions, the controller has to "know" the position of each axis. It does this by comparing the serial pulse coder reading during operation with a reading taken at a known mechanical reference point on the robot. Mastering records the serial pulse coder reading at a known mechanical reference point. This mastering data is stored with other user data in the controller backup battery (and maintained when the controller is shut down).

When the controller is shut down under normal conditions, each serial pulse coder's present reading is maintained in the pulse coder by backup batteries *on the robot* . (These batteries might be located in the controller on P-series robots.) When the controller is turned on, the controller requests the stored reading from the serial pulse coder. When the controller receives the pulse coder reading, the servo system can operate normally. This process is called *calibration* . Calibration occurs automatically each time the controller is turned on.

When pulse coder backup battery power is interrupted while the controller is powered down, calibration fails at power up and the only robot motion possible is joint mode jogging. To restore proper operation, the robot must be mastered and calibrated.



Warning

When calibration has failed, the axis software travel limits will be ignored, allowing the robot to be jogged farther than normal. Be careful when jogging the robot in an uncalibrated condition or you could injure personnel or damage equipment.

Note The following procedures might be password-protected on your robot. If passwords have been set, you will need access at the Install level to perform mastering. Obtain the Install level password and log in at the Install level or you will be unable to perform the following procedures.

Before mastering the robot, you must clear any faults that prevent servo power from being restored or that prevent mastering completion.

Use Procedure D-1 to clear common faults related to mastering and to prepare the robot for mastering. For more detailed information on fault recovery, refer to the *FANUC Robotics Controller Maintenance Manual.*

If you are using a FANUC Robotics A-series robot you should use a mastering fixture to master your robot. Fixture mastering is performed on **FANUC Robotics P-series robots** by aligning mastering pins and surfaces on the robot. Refer to the *Mechanical Unit Service Manual* specific to your robot model for procedures on how to set up and use a mastering fixture.

If you are using a FANUC Robotics M-series or S-series robot you can either master to a fixture or you can master to zero degrees. Refer to the *Mechanical Unit Service Manual* specific to your robot model for more information about either of these methods.

If you are using a FANUC Robotics P-series robot , and you have zero degree witness marks scored onto your robot, then you can master the robot to zero degrees. Refer to the *Mechanical Unit Service Manual* specific to your robot model for procedures on how to master your robot.

Quick mastering is a convenient way to master a robot **after** you have recorded a reference position. You cannot quick master a robot unless the reference position was taught before mastering was lost.



Caution

Record the quick master reference position after the robot is installed to preserve the factory mastering settings for future remastering.

Refer to Table D–1 for information on each kind of mastering procedure.

Table D–1. SYSTEM Master/Cal Items

ITEM	DESCRIPTION
SET QUICK MASTER REF	This item allows you to set master reference positions. You can save the master reference positions to the default device in case the robot has lost mastery due to an electrical or software problem.
FIXTURE POSITION MASTER	This item allows you to master the robot to a fixture. When you master to a fixture, you use a mastering fixture to align the robot axes and then record their serial pulse coder readings. You can master any robot to a fixture.
SINGLE AXIS MASTER	This item allows you to master a single axis of the robot. You can master a single axis of any robot provided that there is a reference mark at a known position on that axis. When a single axis of a robot is mastered, mastering data for the other axes remains unchanged.

Table D–1. SYSTEM Master/Cal Items (Cont'd)

QUICK MASTER	This item allows you to quick master the robot. Quick mastering allows you to minimize the time required to remaster the robot using a reference position you established when the robot was properly mastered. You cannot quick master the robot unless you have previously recorded this quick master reference position. Record the quick master reference position when the robot is properly mastered. The best time to record the quick master reference position is when the robot is still factory-mastered. Refer to the Mastering appendix in the application-specific *Setup and Operations manual* .
ZERO POSITION MASTER	This item allows you to master the robot to zero degrees. When you master to zero degrees, you position all axes at their zero degree witness marks and record their serial pulse coder readings. You can master any M-series or S-series robot to zero degrees. If you are using a FANUC Robotics P-series robot, and you have witness marks scored onto your robot, you can master the robot to zero degrees.
CALIBRATE	This item allows you to calibrate the robot.
Message Field	This item is a status area at the bottom of the screen that lets you know when the robot has been successfully mastered or calibrated.

D.2 RESETTING ALARMS AND PREPARING FOR MASTERING

When you turn on the robot after pulse coder backup battery power has been interrupted you will see a SRVO-062 BZAL alarm. You might also see a SRVO-038 Pulse Mismatch alarm. Before mastering the robot you must reset these alarms and rotate the motor of each axis that lost battery power to prepare the robot for mastering. Use Procedure D-1 to reset these alarms and prepare the robot for mastering.

Procedure D-1 Preparing the Robot for Mastering

Conditions

- You see a SRVO-062 BZAL or SRVO-038 Servo mismatch alarm.
- You are not mastering a P-200 robot.

Steps

1. If necessary, replace the robot batteries with four new 1.5 volt alkaline batteries, size D. Observe the direction arrows in the battery box for proper orientation of the batteries.
2. Press MENUS.
3. Select SYSTEM.
4. Press F1, [TYPE].

5. **If Master/Cal is not listed on the [TYPE] menu,** do the following; otherwise, continue to Step 6
   a. Move the cursor to VARIABLE and press ENTER.
   b. Move the cursor to $MASTER_ENB.
   c. Press the numeric key "1" and then press ENTER on the teach pendant.
   d. Press F1, [TYPE].
6. Select Master/Cal. You will see a screen similar to the following.



Warning

For the M-6i (ARC Mate 100i), M-16i (ARC Mate 120i), and M-16iL (ARC Mate 120iL) robots, setting TORQUE to OFF using the F4, TORQUE function key on the SYSTEM Master/Cal screen releases the robot brakes. When the brakes are released, the robot arm will drop suddenly unless it is supported. DO NOT use this function key unless instructed to do so, otherwise, you could injure personnel or damage equipment.

```
SYSTEM Master/Cal
TORQUE = ON
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE
Press 'ENTER' or number key to select.
```

Note F4, TORQUE does not appear on the Master/Cal screen for all robot models. When F4, TORQUE appears on the Master/Cal screen, it allows the robot brakes to be released.

7. Press F3, RES_PCA. You will see a screen similar to the following.

```
SYSTEM Master/Cal
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE
Press 'ENTER' or number key to select.
Reset pulse coder alarm? [NO]
```

8. Press F4, YES. You will see a screen similar to the following.

```
SYSTEM Master/Cal
TORQUE = ON
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE
Pulse coder
alarm reset!
```

Note If you exit the Master/Cal screen by pressing F5, DONE, the Master/Cal screen will be hidden. Master/Cal will not be available by pressing F1, [TYPE]. To display the Master/Cal screen again, perform Step 1 through Step 6 .

9. Turn off the controller.

10. Wait a few seconds, then turn the power disconnect circuit breaker on to turn the controller on again.

11. If the SRVO-062 BZAL alarm is still present; there is a battery, cable or pulsecoder problem. Refer to the *FANUC Robotics Controller Maintenance Manual* for further information.

12. If a SRVO-038 Pulse Mismatch alarm is present at this time, repeat Step 1 through Step 8 to reset it. It is not necessary to restart the robot after resetting to clear this alarm.

13. If a SRVO-075 Pulse Not Established alarm is present at this time, press the RESET key to clear it.

14. Rotate each axis that lost battery power by at least one motor revolution in either direction. Failure to do so will result in the SRVO-075 Pulse Not Established alarm recurring and mastering will not be possible.

a. For each rotary axis, jog at least twenty degrees.

b. For each linear axis, jog at least thirty millimeters.

15. Perform any of the mastering procedures from the MASTER/CAL menu.

D.3 SAVING AND RESTORING MASTERING DATA

You can save the master reference positions to the default device in case the robot has lost mastery due to an electrical or software problem. Use Procedure D-2 to save and restore master reference position data.

Procedure D-2 Saving and Restoring Master Reference Position Data

Conditions

- The robot is mastered. Refer to Section D.4 through Section D.7 to choose a mastering method and master your robot.
- The default device is set.
- The robot is on and is working properly.

Steps

1. Press MENUS.
2. Select SYSTEM.
3. Press F1, [TYPE].
4. Select Master/Cal.

   If Master/Cal is not listed on the [TYPE] menu, do the following; otherwise, continue to Step 5

   a. Select VARIABLE from the [TYPE] menu.

   b. Move the cursor to $MASTER_ENB.

   c. Press the numeric key "1" and then press ENTER on the teach pendant.

   d. Press F1, [TYPE].

   e. Select Master/Cal. You will see a screen similar to the following.

```
SYSTEM Master/Cal
1  FIXTURE POSITION MASTER
2  ZERO POSITION MASTER
3  QUICK MASTER
4  SINGLE AXIS MASTER
5  SET QUICK MASTER REF
6  CALIBRATE
Press 'Enter' or number key to select.
```

   Note F4, TORQUE does not appear on the Master/Cal screen for all robot models. When F4, TORQUE appears on the Master/Cal screen, it allows the robot brakes to be released.

5. Press FCTN.
6. Select SAVE. The file SYSMAST.SV is copied to the default device.
7. **To restore the saved file,** press F2, LOAD.

D.4 MASTERING TO A FIXTURE (FIXTURE POSITION MASTER)

When you master to a fixture, you use a mastering fixture to align the robot axes and then record their serial pulse coder readings. You can master any robot to a fixture.

Use Procedure D-3 to master to a fixture.

Procedure D-3 Mastering to a Fixture

Conditions

- You have the appropriate mastering fixture for your robot.
- You are not mastering a P-series robot.
- You have cleared any servo faults that prevent you from jogging the robot.
- You have jogged each axis that has lost mastering at least one motor turn. See Procedure D-1

Steps

1. If automatic brake control is enabled, disable it as follows; otherwise continue to Step 2
   a. Press MENUS.
   b. Select SYSTEM.
   c. Press F1, [TYPE].
   d. Select VARIABLE.
   e. Move the cursor to $PARAM_GROUP and press ENTER twice.
   f. Move the cursor to $SV_OFF_ALL and press F5, FALSE.
   g. Move the cursor to $SV_OFF_ENB and press ENTER.
   h. Move the cursor to each line where the value is TRUE, and press F5, FALSE.
   i. Turn off the controller.
   j. Turn on the controller.
2. Install the mastering fixture on the robot and jog the robot into the mastering position. Refer to the *Mechanical Service Manual* or *Mechanical Connection and Maintenance Manual* specific to your robot model for the procedures for set up and use of a mastering fixture.
3. Press MENUS.
4. Select SYSTEM.
5. Press F1, [TYPE].
6. **If Master/Cal is not listed on the [TYPE] menu,** do the following; otherwise, continue to Step 7
   a. Move the cursor to VARIABLE and press ENTER.

b. Move the cursor to $MASTER_ENB.

c. Press the numeric key "1" and then press ENTER on the teach pendant.

d. Press F1, [TYPE].

7. Select Master/Cal. You will see a screen similar to the following.

```
SYSTEM Master/Cal
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE
Press 'ENTER' or number key to select.
```

Note F4, TORQUE does not appear on the Master/Cal screen for all robot models. When F4, TORQUE appears on the Master/Cal screen, it allows the robot brakes to be released.

8. Move the cursor to FIXTURE POSITION MASTER and press ENTER. You will see a screen similar to the following.

```
SYSTEM Master/Cal
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE
Press 'ENTER' or number key to select.
Master at master position ? [NO]
```

9. Press F4, YES. Mastering will be performed automatically. You will see a screen similar to the following.

```
SYSTEM Master/Cal
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE
Robot Mastered!  Mastering Data:
<0> <11808249> <38767856>
<9873638> <122000309> <2000319>
```

10. Move the cursor to CALIBRATE and press ENTER. You will see a screen similar to the following.

```
SYSTEM Master/Cal
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE
Press 'ENTER' or number key to select.
Calibrate ? [NO]
```

11. Press F4, YES. You will see a screen similar to the following

```
SYSTEM Master/Cal
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE
Robot Calibrated!  Cur Jnt Ang(deg):
<0.000> <29.226> <-100.625>
<0.000> <-79.375> <0.000>
```

Note If you exit the Master/Cal screen by pressing F5, DONE, the Master/Cal screen will be hidden. Master/Cal will not appear on the SYSTEM F1, [TYPE] submenu. To see the Master/Cal screen again perform Step 1 through Step 7

12. Move the wrist clear of the mastering fixture and remove the mastering fixture from the robot. Refer to the *Mechanical Service Manual* or *Mechanical Connection and Maintenance Manual* specific to your robot model for the procedures for use of a mastering fixture.
13. If you have disabled automatic brake control in Step 1 , enable it as follows:
    a. Press MENUS.
    b. Select SYSTEM.
    c. Press F1, [TYPE]
    d. Select VARIABLE.
    e. Move the cursor to $PARAM_GROUP and press ENTER twice.
    f. Move the cursor to $SV_OFF_ALL and press F4, TRUE.
    g. Move the cursor to $SV_OFF_ENB and press ENTER.
    h. Move the cursor to each line where you previously changed the value, and press F4, TRUE.
14. Turn off the controller.
15. Wait a few seconds, then press the ON/OFF button on the operator panel to turn the controller on again.
16. If you are using Dual Check Safety and the alarm "SYST-212 Need to apply to DCS param" occurs, the value of the setting parameters and the value of the DCS parameters are different. To reset this alarm, you must perform the “Apply to DCS parameter” procedure. Refer to the Dual Check Safety Operator’s Manual for more information.

D.5 SINGLE AXIS MASTERING

You can master a single axis of any robot provided that there is a reference mark at a known position on that axis. When a single axis of a robot is mastered, mastering data for the other axes remains unchanged.

Use Procedure D-4 to master a single axis.

Procedure D-4 Mastering a Single Axis

Conditions

- You have cleared any servo faults that prevent you from jogging the robot.
- You have jogged each axis that has lost mastering at least one motor turn. See Procedure D-1

Steps

1. Jog the unmastered axis of the robot to the single axis mastering position.

- **For M-series or S-series robots only:** Using the joint coordinate system, jog the unmastered axis of the robot to the zero degree witness mark. If you are single axis mastering the J3-axis, the J2-axis must first be lined up at its zero degree mark. Otherwise the positions of the other axes are unimportant. Refer to the *Mechanical Service Manual* or *Mechanical Connection and Maintenance Manual* specific to your robot model for the location of the zero degree witness marks.
- **For P-series robots only:** Using the joint coordinate system, jog the unmastered axis of the robot to the standard mastering location and align the mark, pin or surface using a straight edge if necessary. If you are single axis mastering the J3-axis, the J2-axis must first be lined up at its standard mastering location. Otherwise the positions of the other axes are unimportant.

2. Press MENUS.
3. Select SYSTEM.
4. Press F1, [TYPE].
5. **If Master/Cal is not listed on the [TYPE] menu,** do the following; otherwise, continue to Step 6
   a. Select VARIABLE.
   b. Move the cursor to $MASTER_ENB.
   c. Press the numeric key "1" and then press ENTER on the teach pendant.
   d. Press F1, [TYPE].
6. Select Master/Cal. You will see a screen similar to the following.

```
SYSTEM Master/Cal
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE
Press 'ENTER' or number key to select.
```

Note F4, TORQUE does not appear on the Master/Cal screen for all robot models. When F4, TORQUE appears on the Master/Cal screen, it allows the robot brakes to be released.

7. Move the cursor to SINGLE AXIS MASTER and press ENTER. You will see a screen similar to the following.

```
SINGLE AXIS MASTER
ACTUAL   POS      (MSTR POS)    (SEL)     [ST]
J1     0.000     (     0.000)     (0)       [2]
J2     3.514     (    35.000)     (0)       [0]
J3    -7.164     ( -100.000)     (0)       [2]
J4   -35.366     (     0.000)     (0)       [2]
J5    -1.275     (   -80.000)     (0)       [2]
J6     4.571     (     0.000)     (0)       [2]
E1     0.000     (     0.000)     (0)       [0]
E2     0.000     (     0.000)     (0)       [0]
E3     0.000     (     0.000)     (0)       [0]
```

Note A "0" in the [ST] column indicates that the axis is unmastered.

8. Move the cursor to the (MSTR POS) column and move it up or down to the unmastered axis. (Any unmastered axis will have the number 0 in the [ST] column).

9. Enter the position where single axis mastering is to be performed in the (MSTR POS) column for the unmastered axis.

a. For robots where single axis mastering is performed at the zero degree position, press the numeric key "0" and press ENTER.

b. For robots where single axis mastering is performed at the fixture position, enter the fixture position and press ENTER.

Note Some P-series robots are single axis mastered at the fixture position. All other robots are single axis mastered at the zero degree position.

10. Continuously press and hold the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.

11. Move the cursor to the SEL column and move it up or down to the unmastered axis.

12. Press the numeric key "1" and press ENTER.

13. Press F5, EXEC. Mastering will be performed automatically. You will see a screen similar to the following.

```
SINGLE AXIS MASTER
ACTUAL POS    (MSTR POS)  (SEL)  [ST]
J1    0.000   (   0.000)   (0)    [2]
J2    3.514   (   0.000)   (0)    [2]
J3   -7.164   (-100.000)   (0)    [2]
J4  -35.366   (   0.000)   (0)    [2]
J5   -1.275   ( -80.000)   (0)    [2]
J6    4.571   (   0.000)   (0)    [2]
E1    0.000   (   0.000)   (0)    [0]
E2    0.000   (   0.000)   (0)    [0]
E3    0.000   (   0.000)   (0)    [0]
```

Note The J2 axis is now mastered and a 2 is displayed in the [ST] column.

14. Press PREV.

15. Move the cursor to CALIBRATE and press ENTER. You will see a screen similar to the following.

```
SYSTEM Master/Cal
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE
Press 'ENTER' or number key to select.
Calibrate ? [NO]
```

16. Press F4, YES. You will see a screen similar to the following.

```
SYSTEM Master/Cal
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE
Robot Calibrated!  Cur Jnt Ang(deg):
<0.000> <0.000> <-7.164>
<-35.366> <-1.275> <4.571>
```

Note If you exit the Master/Cal screen by pressing F5, DONE, the Master/Cal screen will be hidden. Master/Cal will not be available by pressing 1, [TYPE]. To display the Master/Cal screen again perform Step 1 through Step 6

17. If you are using Dual Check Safety and the alarm "SYST-212 Need to apply to DCS param" occurs, the value of the setting parameters and the value of the DCS parameters are different. To reset this alarm, you must perform the "Apply to DCS parameter" procedure. Refer to the Dual Check Safety Operator's Manual for more information.

D.6 QUICK MASTERING

Quick mastering allows you to minimize the time required to remaster the robot using a reference position you established when the robot was properly mastered. You cannot quick master the robot unless you have previously recorded this quick master reference position.

Record the quick master reference position when the robot is properly mastered. The best time to record the quick master reference position is when the robot is still factory-mastered.

If calibration fails because pulse coder backup battery power has been interrupted, you can use this reference position to master the robot in a minimum amount of time. When the zero position marks do not line up because of mechanical disassembly or repair, you must master to a fixture or perform zero degree mastering.

You can define a quick master reference position and perform quick mastering on any robot model.

Use Procedure D-5 to record the quick master reference position. Use Procedure D-6 to quick master the robot.



Caution

Record the quick master reference position after the robot is installed to preserve the factory mastering settings for future remastering.

Procedure D-5 Recording the Quick Master Reference Position

Conditions

- The robot is properly mastered.

Steps

1. Align each axis of the robot with the reference mark you have chosen as a quick master reference position.

Note It is convenient to use the zero degree marks for the quick master reference position Refer to the *Mechanical Unit Service Manual* or *Mechanical Connection and Maintenance Manual*

specific to your robot model for the location of the zero degree marks. But if you prefer, you can use any robot position as long as you scribe witness marks on each axis at the reference point.

2. Press MENUS.
3. Select SYSTEM.
4. Press F1, [TYPE].
5. **If Master/Cal is not listed on the [TYPE] menu,** do the following; otherwise, continue to Step 6
   a. Select VARIABLE.
   b. Move the cursor to $MASTER_ENB.
   c. Press the numeric key "1" and then press ENTER on the teach pendant.
   d. Press F1, [TYPE].
6. Select Master/Cal. You will see a screen similar to the following.

```
SYSTEM Master/Cal
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE
Press 'ENTER' or number key to select.
```

Note F4, TORQUE does not appear on the Master/Cal screen for all robot models. When F4, TORQUE appears on the Master/Cal screen, it allows the robot brakes to be released.

7. Move the cursor to SET QUICK MASTER REF and press ENTER. You will see a screen similar to the following.

```
SYSTEM Master/Cal
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE
Press 'ENTER' or number key to select.
Set quick master reference ? [NO]
```

8. Press F4, YES. You will see a screen similar to the following.

```
SYSTEM Master/Cal
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE
Quick Master Reference Set!
```

Note If you exit the Master/Cal screen by pressing F5, DONE, the Master/Cal screen will be hidden. Master/Cal will not be available by pressing F1, [TYPE]. To display the Master/Cal screen again, perform Step 1 through Step 6

Procedure D-6 Quick Mastering the Robot

Conditions

- Calibration has failed because backup battery power has been interrupted.

Note If the zero degree marks do not line up because of mechanical disassembly or repair, you cannot perform this procedure. In this case, master to a fixture or master to zero degrees to restore robot mastering.

- The quick master reference position was recorded before calibration failed.
- You have cleared any servo faults that prevent you from jogging the robot.
- You have jogged each axis that has lost mastering at least one motor turn. See Procedure D-1

Steps

1. Jog the robot to the quick master reference position that has previously been recorded.
2. Press MENUS.
3. Select SYSTEM.
4. Press F1, [TYPE].
5. **If Master/Cal is not listed on the [TYPE] menu,** do the following; otherwise, continue to Step 6
   a. Select VARIABLE.
   b. Move the cursor to $MASTER_ENB.
   c. Press the numeric key "1" and then press ENTER on the teach pendant.
   d. Press F1, [TYPE].

6. Move the cursor to Master/Cal and press ENTER. You will see a screen similar to the following.

```
SYSTEM Master/Cal
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE
Press 'ENTER' or number key to select.
```

Note F4, TORQUE does not appear on the Master/Cal screen for all robot models. When F4, TORQUE appears on the Master/Cal screen, it allows the robot brakes to be released.

7. Move the cursor to QUICK MASTER and press ENTER. You will see a screen similar to the following.

```
SYSTEM Master/Cal
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE
Press 'ENTER' or number key to select.
Quick master ? [NO]
```

8. Press F4, YES. Mastering will be performed automatically. You will see a screen similar to the following.

```
SYSTEM Master/Cal
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE
Robot Mastered!  Mastering Data:
<0> <11808249> <38767856>
<9873638> <122000309> <2000319>
```

9. Move the cursor to CALIBRATE and press ENTER. You will see a screen similar to the following.

```
SYSTEM Master/Cal
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE
Press 'ENTER' or number key to select.
Calibrate ? [NO]
```

10. Press F4, YES. You will see a screen similar to the following.

```
SYSTEM Master/Cal
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE
Robot Calibrated!  Cur Jnt Ang(deg):
<0.003> <-0.0012> <-0.050>
<0.009> <0.015> <-0.002>
```

Note If you exit the Master/Cal screen by pressing F5, DONE, the Master/Cal screen will be hidden. Master/Cal will not be available by pressing F1, [TYPE]. To display the Master/Cal screen again, perform Step 1 through Step 6

11. If you are using Dual Check Safety and the alarm "SYST-212 Need to apply to DCS param" occurs, the value of the setting parameters and the value of the DCS parameters are different. To reset this alarm, you must perform the "Apply to DCS parameter" procedure. Refer to the Dual Check Safety Operator's Manual for more information.

D.7 ZERO DEGREE MASTERING

When you master to zero degrees, you position all axes at their zero degree witness marks and record their serial pulse coder readings. You can master any M-series or S-series robot to zero degrees.

If you are using a FANUC Robotics P-series robot, and you have witness marks scored onto your robot, you can master the robot to zero degrees.

Use Procedure D-7 to master to zero degrees.

Procedure D-7 Mastering to Zero Degrees

Conditions

- You have cleared any servo faults that prevent you from jogging the robot.
- You have jogged each axis that has lost mastering at least one motor turn. Refer to Procedure D-1

Steps

1. Using the joint coordinate system, jog each axis of the robot to the zero degree witness mark. Refer to the *Mechanical Service Manual* or *Mechanical Connection and Maintenance Manual* specific to your robot model for the location of the witness marks.
2. Press MENUS.
3. Select SYSTEM.
4. Press F1, [TYPE].
5. **If Master/Cal is not listed on the [TYPE] menu,** do the following; otherwise, continue to Step 6
   a. Select VARIABLE.
   b. Move the cursor to $MASTER_ENB.
   c. Press the numeric key "1" and then press ENTER on the teach pendant.
   d. Press F1, [TYPE].
6. Select Master/Cal. You will see a screen similar to the following.

```
SYSTEM Master/Cal
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE
Press 'ENTER' or number key to select.
```

Note F4, TORQUE does not appear on the Master/Cal screen for all robot models. When F4, TORQUE appears on the Master/Cal screen, it allows the robot brakes to be released.

7. Move the cursor to ZERO POSITION MASTER and press ENTER. You will see a screen similar to the following.

```
SYSTEM Master/Cal
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE
Press 'ENTER' or number key to select.
Master at zero position ? [NO]
```

8. Press F4, YES. Mastering will be performed automatically. You will see a screen similar to the following.

```
SYSTEM Master/Cal
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE
Robot Mastered!  Mastering Data:
<0> <11808249> <38767856>
<9873638> <122000309> <2000319>
```

9. Move the cursor to CALIBRATE and press ENTER. You will see a screen similar to the following.

```
SYSTEM Master/Cal
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE
Press 'ENTER' or number key to select.
Calibrate ? [NO]
```

10. Press F4, YES. You will see a screen similar to the following.

```
SYSTEM Master/Cal
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE
Robot Calibrated!  Cur Jnt Ang(deg):
<0.000> <0.000> <0.000>
<0.000> <0.000> <0.000>
```

Note If you exit the Master/Cal screen by pressing F5, DONE, the Master/Cal screen will be hidden. Master/Cal will not be available by pressing F1, [TYPE]. To display the Master/Cal screen again perform Step 1 through Step 6

11. If you are using Dual Check Safety and the alarm "SYST-212 Need to apply to DCS param" occurs, the value of the setting parameters and the value of the DCS parameters are different. To reset this alarm, you must perform the “Apply to DCS parameter” procedure. Refer to the Dual Check Safety Operator’s Manual for more information.

Appendix E

PROGRAM EXAMPLES

Contents

Appendix E	PROGRAM EXAMPLES	E–1
E.1	Program Examples Overview	E–2
E.2	POSITION REGISTER ELEMENT	E–2
E.3	POSITION REGISTER VALUE	E–2
E.4	REGISTER ANALOG INPUT	E–3
E.5	CONDITIONAL BRANCHING; USING LABELS	E–4
E.6	CALL KAREL SOFTPART	E–5
E.7	REGISTER INCREMENT	E–5
E.8	GROUP OUTPUT; WAIT INSTRUCTION PULSE INSTRUCTION	E–6
E.9	LABELS	E–6
E.10	LABEL; JUMP LABEL; MESSAGE	E–7
E.11	MACRO INSTRUCTION	E–8

E.1 Program Examples Overview

This appendix contains program examples that illustrate the use of teach pendant program instructions.

Each example shows one or more program elements used in a program. Comments are included with each program.

E.2 POSITION REGISTER ELEMENT

Position Register Element

```
/PROG PREG_ELE
   1:  !POSITION REG VALUE
   2:J P[1:ABOVE JOINT] 100% FINE
   3:J P[2] 100% FINE
   4:  PR[1]=LPOS
   5:  PR[1,2]=600
   6:L PR[1] 100.0inch/min FINE
   7:J P[1:ABOVE JOINT] 100% FINE
/END
```

Program Comments

1. REMARK instruction, identified by an !, with the message "POSITION REG VALUE" displayed within the program.
2. Joint move to position 1:ABOVE JOINT with 100% travel speed and Fine termination.
3. Joint move to position 2 with 100% travel speed and Fine termination.
4. Position register 1 equals the current Cartesian coordinates position (x,y,z,w,p,r,config).
5. The second element of position register 1 equals 600.
6. Linear move to position register 1 with 100 inches per minute travel speed and Fine termination.
7. Joint move to position 1:ABOVE JOINT with 100% travel speed and Fine termination.

E.3 POSITION REGISTER VALUE

Position Register Value

```
/PROG  PREG_VAL
   1:  !POSITION REG VALUE
   2:J P[1:ABOVE JOINT] 100% FINE
   3:J P[2] 100% FINE
   4:  PR[1]=LPOS
```

```
   5:  PR[1]=PR[1]+PR[2]
   6:J PR[1] 100% FINE
   7:J P[1:ABOVE JOINT] 100% FINE
/END
```

Program Comments

1. REMARK instruction, identified by an !, with the "POSITION REG VALUE" displayed within the program.
2. Joint move to position 1:ABOVE JOINT with 100% travel speed and Fine termination.
3. Joint move to position 2 with 100% travel speed and Fine termination.
4. Position register 1 equals the current Cartesian coordinates position (x,y,z,w,p,r,config).
5. Position register 1 equals position register 1 plus position register 2.
6. Joint move to position register 1 with 100% travel speed and Fine termination.
7. Joint move to position 1:ABOVE JOINT with 100% travel speed and Fine termination.

E.4 REGISTER ANALOG INPUT

Register Analog Input

```
/PROG  REG_AI
   1:  !REGISTER : ANALOG IN
   2:  R[1]=0
   3:  R[2]=0
   4:  R[1]=AI[3]
   5:  R[1]=R[1]-256
   6:  R[2]=R[1] DIV 256
   7:  LBL[1:zero check ]
   8:  R[2]=R[2]+1
   9:  IF R[2]=0,JMP LBL[1]
  10:J P[1:SAFE POSITION] 100% FINE
/END
```

Program Comments

1. REMARK instruction, identified by an !, with the message "REGISTER : ANALOG IN" displayed within the program.
2. Register 1 equals the value 0.
3. Register 2 equals the value 0.

4. Register 1 equals the value of analog input 3.
5. Register 1 equals register 1 minus 256.
6. Register 2 equals register 1 divided by 256.
7. The Label marks the program as the destination of a program branch. The label can have an identifier; for example, ‘zero check’ (LBL 1:zero check).
8. Register 2 equals register 2 plus 1.
9. If instruction branches based upon the decision. If register 2 equals zero then jump to label 1 (program step 7).
10. Joint move to position 1:SAFE POSITION with 100% travel speed and Fine termination.

E.5 CONDITIONAL BRANCHING; USING LABELS

Condition Branching and Using Labels

```
/PROG  REG_GI
   1:  !REGISTER : GROUP INPUT
   2:  LBL[1:check schd num]
   3:  R[1]=GI[1]
   4:  IF GI[1]>32,JMP LBL[1]
   5:  IF GI[1]<=0,JMP LBL[1]
   6:J P[1:ABOVE PART] 100% FINE
   7:J P[2] 40% FINE
   8:J P[1:ABOVE PART] 100% FINE
/END
```

Program Comments

1. REMARK instruction, identified by an !, with the message "REGISTER : GROUP INPUT" displayed within the program.
2. The Label marks the program as the destination of a program branch. The label can have an identifier; for example, check schd num’ (LBL 1:check schd num).
3. Register 1 equals the value of group input 1.
4. If instruction branches based upon the decision. If group input 1 is greater than 32 then jump to label 1 (program step 2).
5. 5. If instruction branches based upon the decision. If group input 1 is less than or equal to 0 then jump to label 1 (program step 2).
6. Joint move to position 1:ABOVE PART with 100% travel speed and Fine termination.
7. Joint move to position 2 with 40% travel speed and Fine termination.

8. Joint move to position 1:ABOVE PART with 100% travel speed and Fine termination.

E.6 CALL KAREL SOFTPART

This program is an example of using the Register instructions to pass values to a KAREL softpart. The values determine which routine to run.

R[1] is set apart to contain the KAREL softpart routine number to execute. For example, 922 is interference zones, and 923 is safe zone.

Call KAREL Softpart

```
/PROGRAM INTFZONE
   1:  !Run interference zones
   2:  R[1] = 922   -- used to determine routine
   3:  CALL KARELPG     -- A KAREL softpart
```

Program Comments

1. A remark instruction, identified by an !.
2. Setting which routine in the softpart to execute.
3. This calls the KAREL softpart that will read the register and execute the interference zone checking routine and error handling.

E.7 REGISTER INCREMENT

This program is an example of using the Register instructions to store the number of cycles. R[11] is used to store the cycle count.DI[1] is the digital input that signals a completed cycle.

Register Increment

```
/PROGRAM CYCLECNT
   1:  !This program counts cycles
   2:  R[11] = R[11] + DI[1]
```

Program Comments

1. A remark instruction, identified by an !.
2. Register 11 is added to the digital input. If the digital input signal is high the count is incremented. The result is stored in R[11].

E.8 GROUP OUTPUT; WAIT INSTRUCTION PULSE INSTRUCTION

This program is an example of using the Register instructions and digital input and output signal to do some external handshaking.R[11] is used to store the style type. R[12] is used to store the error codes. GO[11] is used to send style type to external device. GI[11] is used to receive the error codes. DI[11] digital input that signals style number on group output DO[11] digital output that signals error codes on group input

Group Output, Wait Instruction, Pulse Instruction

```
/PROGRAM SIGNAL
   1:  !
   2:  GO[11] = R[11]
   3:  DO[11] = PULSE ,.5sec
   4:  WAIT DI[11]
   5:  R[12] = GI[11]
   6:  DO[11] = PULSE ,.5sec
```

Program Comments

1. A remark instruction, identified by an !.
2. Style number stored in register[11] is put on Group output lines 11.
3. Program signals via digital output 11 that group output lines have a valid number.
4. Wait until received acknowledge is received. This signal is also used to indicate that a valid error code in on group input lines.
5. Store error code in register 12.
6. Send an acknowledgment of group input received.

E.9 LABELS

This program is an example of using the register instruction to pass values to a KAREL softpart.

Labels

```
/PROGRAM MAIN
  1: !Main program of example
  2: P[1] L 50mm/sec


3:  CALL SIGNAL
  4: IF R[12] = 0 JMP LBL[1]
  5: JMP LBL[R[12]]
```

```
 6: ABORT
 7: LBL[23: BAD STYLE]
 8: MESSAGE[ BAD STYLE]
 9: JMP LBL[1]
10: LBL[24: TOOL BROKEN]
11: MESSAGE [TOOL BROKEN ]
12: LBL[1: CONTINUE]
```

Program Comments

1. Remark instruction, identified by an !.
2. Move robot to communication position.
3. Call signal program to send style number to dispense equipment and get error codes back.
4. If no error (=0) then jump to continue.
5. Jump to label of error code.
6. Otherwise, abort.
7. Label 23 bad style.
8. Print message on user screen that a bad style was selected.
9. Then jump over rest of error messages.
10. Label 24 tool broken.
11. Print message on user screen that a broken tool was discovered.
12. Continue to other programs.

E.10 LABEL; JUMP LABEL; MESSAGE

This program is an example of using the JMP and LBL instructions. R[12] is the register that contains the error codes. It is set in program signal. LBL[1] is the continue label. LBL[23] is the part of the program that handles bad style error codes. LBL[24] is the part of the program that handles a broken tool error code. SIGNAL is a program that communicates to external hardware and sets R[12] to the error codes received by the external hardware.

Label, Jump Label, Message

```
/PROGRAM MAIN
   1: !Main program of example
   2: P[1] L 50mm/sec
   3: CALL SIGNAL
   4: IF R[12] = 0 JMP LBL[1]
   5: JMP LBL[R[12]]
```

```
 6: ABORT
 7: LBL[23: BAD STYLE]
 8: MESSAGE[ BAD STYLE]
 9: JMP LBL[1]
10: LBL[24: TOOL BROKEN]
11: MESSAGE [TOOL BROKEN ]
12: LBL[1: CONTINUE]
```

Program Comments

1. Remark instruction, identified by an !.
2. Move robot to communication position.
3. Call signal program to send style number to dispense equipment and get error codes back.
4. If no error (= 0) then jump to continue.
5. Jump to label of error code.
6. Else Abort.
7. Label 23 Bad Style.
8. Print message on user screen that a bad style was selected.
9. Then jump over rest of error messages.
10. Label 24 Tool broken.
11. Print message on user screen that a broken tool was discovered.
12. Continue to other programs.

E.11 MACRO INSTRUCTION

The following program is an example of using the interference zone macros. ENTER ZONE 1 and EXIT ZONE 1 are macro instructions that signal the other that this program is entering an interference zone. It will wait and post an error if the other machine is already there. Otherwise it will enter the zone and then leave when done.

Macro Instruction

```
/PROGRAM MAIN
   1: !Main program of example
   2: P[1] L 50mm/sec
   3: P[2] L 50mm/sec
   4: ENTER ZONE 1
   5: P[3] L 50mm/sec
   6: P[4] L 50mm/sec
   7: P[5] L 50mm/sec
```

```
 8: EXIT ZONE 1
 9: P[6] L 50mm/sec
10: P[7] L 50mm/sec
```

Program Comments

1. Remark instruction, identified by an !.
2. Move robot to position outside of zone.
3. Move robot to position outside of zone.
4. Asking to enter zone, waiting if not clear.
5. Moving to points inside the zone.
6. Moving to points inside the zone.
7. Moving to points inside the zone.
8. Clearing signals to zone so other equipment can use it.
9. Moving to other points outside the zone.
10. Moving to other points outside the zone.

Appendix F

ADVANCED *i*PENDANT FUNCTIONS

Contents

Appendix F	ADVANCED *i*PENDANT FUNCTIONS	F–1
F.1	Overview	F–2
F.2	*i*Pendant Controls Installation	F–2
F.3	Remote Monitoring	F–3
F.3.1	Overview	F–3
F.3.2	Setup	F–3
F.3.3	Operation	F–6
F.3.4	Limitations	F–9
F.4	Remote Operation	F–11
F.4.1	Overview	F–11
F.4.2	Setup	F–11
F.4.3	Remote *i*Pendant Operation	F–12
F.4.4	Limitations	F–19
F.5	Screen Customization	F–20
F.5.1	Overview	F–20
F.5.2	Easy Panel Environment Setup	F–20
F.5.3	Panel Wizard	F–21
F.5.4	*i*Pendant HMI Screen Customizations	F–25
F.5.5	Displaying Web Page Macros	F–33
F.5.6	Customizing User Menus	F–34

F.1 Overview

This chapter provides information about the advanced connectivity and customization features of the *i*Pendant. These include:

- **Installing *i*Pendant Controls** This allows you to install the components necessary to perform any *i*Pendant advanced functions.
- **Remote Monitoring** This allows you to monitor the *i*Pendant screens remotely using a PC and Microsoft® Internet Explorer.
- **Remote Operation** This allows you to display an *i*Pendant-like screen remotely, using a PC and Microsoft® Internet Explorer, and navigate the various menus available in the system. This provides the capability to diagnose system problems remotely.
- **Building Custom Screens** The *i*Pendant provides the capability to display custom HTML pages that can be used for many purposes, including: dynamic data display, production monitoring, or system operation.

F.2 *i*Pendant Controls Installation

Some of the *i*Pendant advanced functions require you to have the FANUC Robotics *i*Pendant Controls loaded on the PC that you will be using. Procedure F-1 describes how and where to obtain these controls and install them on your PC.

Procedure F-1 Installing FANUC Robotics *i*Pendant Controls

Conditions

- Microsoft® FrontPage 2000 ™ or greater is loaded on your PC
- You are using a PC that is connected to the Internet
- You have obtained a FANUC Robotics Customer Resource Center (cRc) login username and password.

Steps

1. Use Microsoft® Internet Explorer on your PC to connect to the FANUC Robotics Customer Resource Center (cRc).
2. Type your username and password at the prompt. Contact FANUC Robotics if you need to request login information.
3. Click Downloads.
4. Click Download PC Software.
5. Scroll to iPendant Controls, click the appropriate version, and follow the directions to begin downloading a self-extracting zip file that contains all the necessary setup files.
6. Log off of the FANUC Robitcs cRc website.

7. On your PC, navigate to the temporary directory where you saved the file iPendant_controls_Vx.xx.exe (where x.xx is the appropriate version number).
8. After the files are unzipped, double-click setup.exe in the setup directory.

The FANUC Robotics *i*Pendant Controls are now installed on your PC and can be used to create custom screens or for remote *i*Pendant operation or monitoring.

F.3 Remote Monitoring

F.3.1 Overview

Remote monitoring provides you with the capability to display and monitor the current *i*Pendant screens and operations on a PC using Microsoft® Internet Explorer. It is meant as a DISPLAY ONLY mode and therefore, the remote connection normally cannot interact with the screens or affect the operation of the *i*Pendant or robot controller. See Section F.3.4 for limitations.

The following can be displayed remotely:

- All *i*Pendant screens available from the MENUS and [TYPE] keys.
- All popup menus, and windows.
- Multiple window configurations (Double and Triple modes on the *i*Pendant for example).
- Any input from the *i*Pendant numeric keypad, Function keys or cursor movement.
- Any custom screens that are accessible from the [TYPE] menu in the BROWSER Screen.
- Any top level HELP or DIAGNOSTIC Screens.

F.3.2 Setup

Setup consists of:

- Identifying requirements
- Configuring Internet Explorer ™
- Testing the Network Connection

F.3.2.1 Requirements

The following are the requirements for remote display of the *i* Pendant screens.

- The PC must have Microsoft® Internet Explorer 5.5 or greater installed.
- The PC must have the *i*Pendant Controls installed. Refer to Procedure F-1 for installation instructions.
- The PC must be connected to a network , and be properly configured to allow a TCP/IP connection to the robot controller with the *i*Pendant connected.
- The robot controller must be connected to a network and be properly configured for Network access to the above PC.
- If there is a firewall between the robot controller and the personal computer, the robot controller must be able to open a TCP connection to port 60005 on the personal computer. The personal computer must also be able to open a TCP connection to port 3002 on the robot controller. This is in addition to allowing the basic HTTP request from the personal computer to the robot controller (TCP connection to port 80 on the robot).

F.3.2.2 Configuring Internet Explorer ™

The following settings are required in Microsoft® Internet Explorer for proper operation of the Remote Monitoring function.

Procedure F-2 Configuring Internet Explorer

1. In Internet Explorer ™ go to TOOLS>Internet Options.
2. **If you are using a proxy server** , make sure that your access to the robot **does not** go through the proxy.
   a. Select the Connections Tab.
   b. Select LAN Settings.
   c. Click Advanced.
   d. Go to Exceptions and include local network which includes the robots. For example, 192.* would exclude from the proxy all the robots on the 192.x.x.x network from going through the proxy.
3. Select the Security Tab.
4. Select Custom Level.
5. Make sure that under ActiveX Controls and Plug-ins the following is checked:
   - Run ActiveX Controls and plug-ins – Enabled.

F.3.2.3 Testing the Network Connection

This section describes a method to verify that the network connection from the PC to the robot is configured correctly and operational.

Procedure F-3 Testing the Network Connection

Conditions

Before performing the test, make sure the following conditions are met:

- The PC is connected to a network that can be used to access the robot.
- The PC has Microsoft® Internet Explorer 5.5 or greater loaded and is properly configured as detailed in Section F.3.2.2 above.
- The robot is turned on and is connected to a network that is accessible by the above PC.

Steps

1. Bring up Internet Explorer on the PC.
2. In the Internet Explorer Address field, Enter “http://<myrobot_name_ or_address>”.

   Where <myrobot_name_ or_address> is either the DNS name of your robot i.e. pderob111.frc.com) or the IP address of your robot. (for example 192.168.1.100)

If the connection is successfully made you will see the HOME page of the robot displayed in Internet Explorer. It will be similar to that shown in Figure F–1 .

Figure F–1. Robot HOME page



If you are unable to make this connection, refer to the *Internet Options Setup and Operations Manual* for additional information or contact your System Network Administrator.

F.3.3 Operation

After you have properly configured Microsoft® Internet Explorer and verified that you can connect to the robot as detailed above you can now access the Remote *i*Pendant screen. The following sections detail the procedure required for this and the limitations of this feature.

F.3.3.1 Remotely Monitoring the *i*Pendant

This section will describe the method to connect to the robot controller and display the remote iPendant screen for monitoring the *i*Pendant operation.

Procedure F-4 Remotely Monitoring the *i* Pendant

Conditions

- The PC is connected to a network that can be used to access the robot.
- The PC has Microsoft® Internet Explorer 5.5 or greater loaded and is properly configured as detailed in Section F.3.2.2 above.
- The PC has the *i*Pendant Controls installed as detailed in Procedure F-1 .
- The robot is turned on and connected to a network that is accessible by the above PC.
- The robot has a functional *i*Pendant connected and operational.

Steps

1. Bring up Internet Explorer on the PC.
2. In the Internet Explorer Address field, Enter “http://<myrobot_name_ or_address>/frh/cgtp/echo.htm”.

   Where <myrobot_name_ or_address> is either the DNS name of your robot (i.e. pderob111.frc.com) or the IP address of your robot. (i.e. 192.168.1.100)

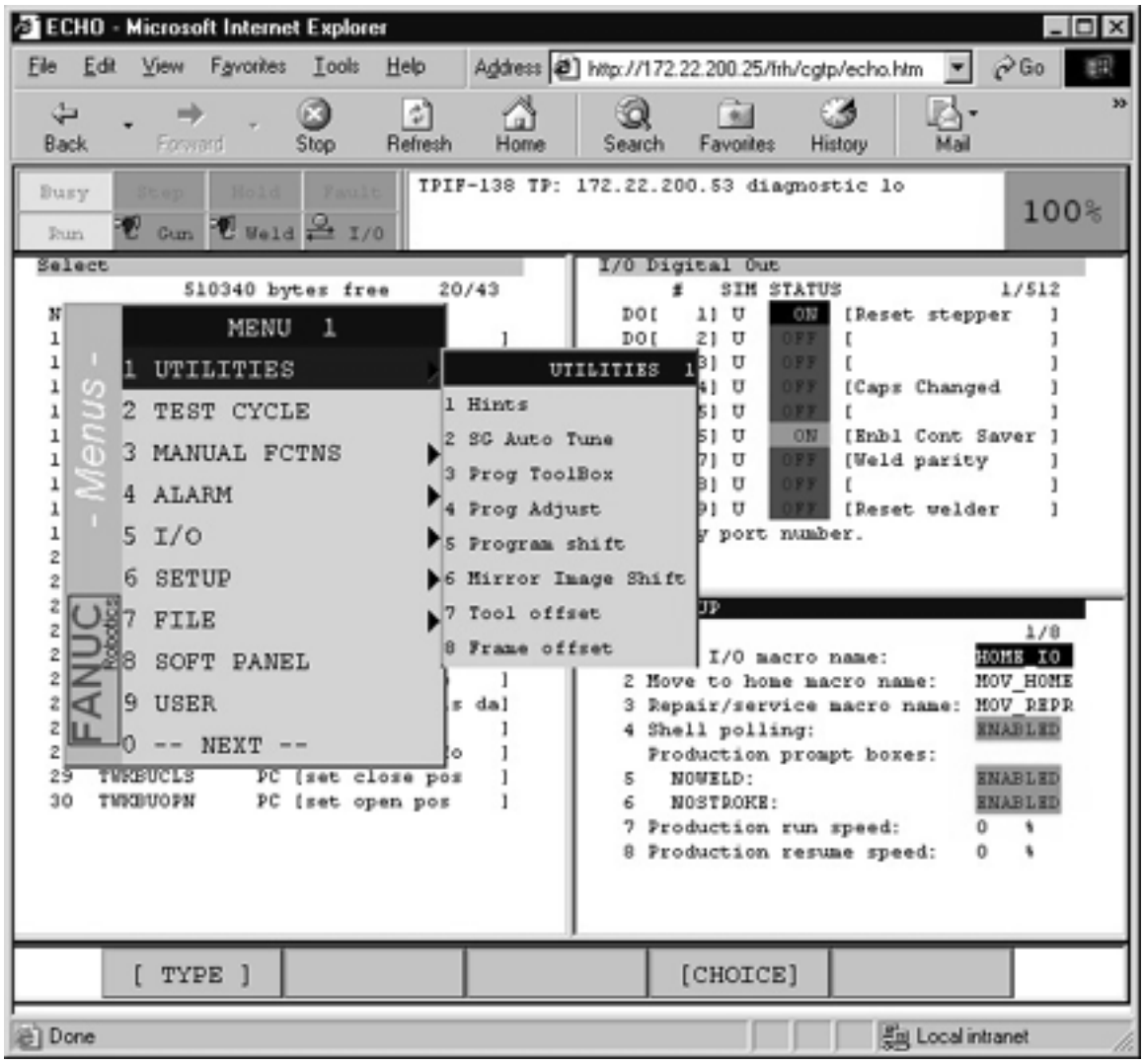
If the connection is successfully made you will briefly see the LOGIN screen, similar to that shown in Figure F–2 , displayed in Internet Explorer.

Figure F–2. Remote *i*Pendant LOGON Screen



After the LOGON Screen, you will see a remote display of the current *i* Pendant screen. The display will show the current *i*Pendant screen and any activity that might be occurring on it (popup menus, dynamic data, screen reconfigurations, for example). It might be similar to that shown in Figure F–3 .

Figure F–3. Remote *i* Pendant Monitoring Screen



F.3.4 Limitations

Remote monitoring provides the user with the capability to display and monitor the current *i*Pendant screens and operations on a PC using Microsoft® Internet Explorer.

The following limitations apply to the remotely displayed *i*Pendant screen during the remote monitoring operation:

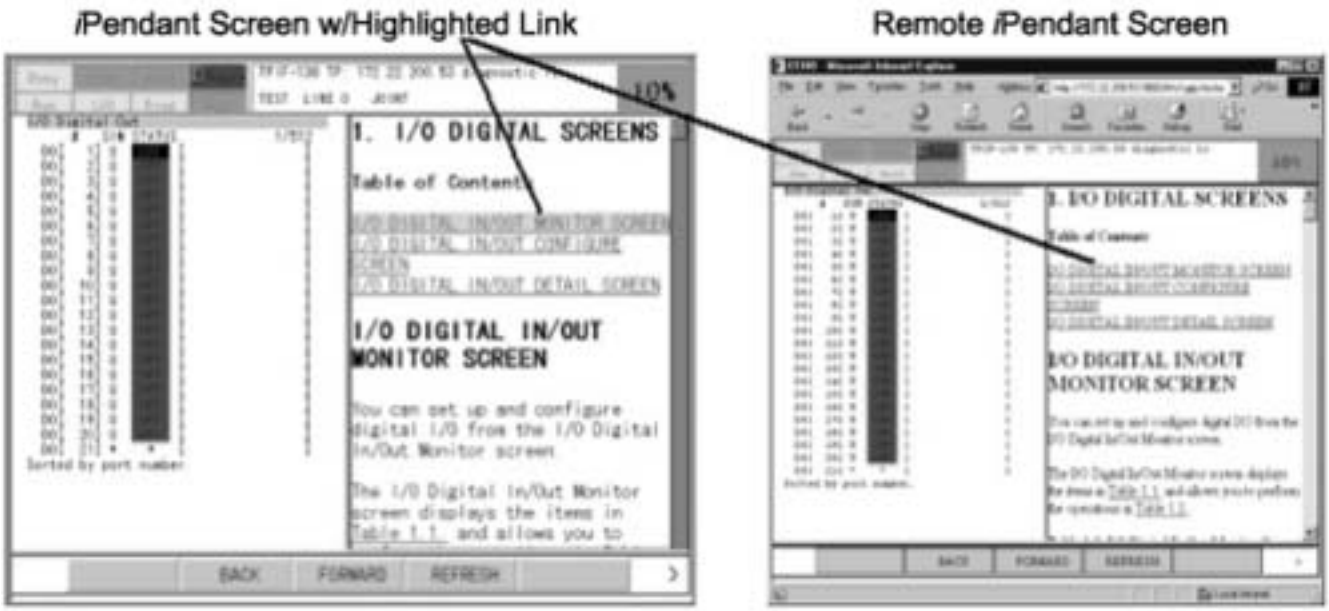
- It is meant as a DISPLAY ONLY mode, as such the remote connection normally cannot interact with the screens or affect the operation of the *i*Pendant or robot controller except in the following cases:

Warning

If any web page is being displayed, any active link or component (ie *i*Pendant Control) can be selected on the remote *i*Pendant screen and activated. This may cause an interaction with the robot or with the actual *i*Pendant. Care must be taken when viewing web pages remotely.

- The remote display will look similar to the *i*Pendant but will not be an exact duplicate. Fonts, Character sizes, colors, images, and overall format might be different.
- If a popup menu is being displayed on the *i*Pendant when the remote monitoring connection is made to the robot controller, that popup menu will not appear on the initial remote display. Subsequent pop-ups will be displayed on the remote screen as they occur on the *i*Pendant.
- Highlighted"Links" on a web page, a custom screen, a Help or Diagnostics screen, being displayed on the *i*Pendant, are not highlighted on the remote screen as shown in Figure F–4 .

Figure F–4. Highlighted Links



F.4 Remote Operation

F.4.1 Overview

Remote Operation allows you to display an *i*Pendant-like screen on a PC using Microsoft ® Internet Explorer. The remote user can configure the remote display, navigate the controller menus and screens, and enter data remotely. Remote Operations are only available if the Internet Connectivity and Connection option is loaded.

> **Warning**
>
> **Remote Operation is completely independent from the operation of the actual *i*Pendant therefore it can affect the operation the robot controller. Extreme caution must be exercised when using this feature.**

Note The remote operator cannot see what is currently on the actual *i*Pendant or what the operator is doing nor can the *i*Pendant operator see what the remote operator is doing or what screens are being displayed on the remote PC.

For information on the limitations of this feature, refer to Section F.4.4 .

The following functions can be performed on the remote *i*Pendant operation display:

- All *i* Pendant screens available from the MENUS and/or [TYPE] keys can be displayed as well as any custom screens and HELP or DIAGNOSTIC Screens.
- All popup menus, and windows.
- Multiple window configurations (i.e. Double and Triple modes on the *i* Pendant).
- Any input from the *i*Pendant numeric keypad, Function keys or cursor movement.

F.4.2 Setup

F.4.2.1 Requirements

The following are the requirements for remote operation of the *i*Pendant screens.

- PC must have Microsoft® Internet Explorer 5.5 or greater installed.
- PC must have the iPendant Controls installed. Refer to Procedure F-1 for installation instructions.
- PC must be connected to a network , and be properly configured to allow a TCP/IP connection to the Robot Controller with the *i*Pendant connected.

- The robot controller must be connected to a network and be properly configured for Network access to the above PC.
- The robot controller must have the Internet Protocol Connectivity and Customization (IPCC) Option installed.
- The HTTP Authentication must be properly configured on the robot to allow iPendant access. Refer to the *Internet Options Setup and Operations Manual* for configuration information.

F.4.2.2 Configuring Microsoft® Internet Explorer

See Section F.3.2.2 for information on configuring Microsoft® Internet Explorer.

F.4.3 Remote *i*Pendant Operation

After you have properly configured Internet Explorer ™ and verified that you can connect to the robot, you can now display the remote *i*Pendant Operation screen. The following sections detail the operation and the limitations of this feature.

F.4.3.1 Connecting to the Controller

This section will describe the method to connect to the robot controller and display the remote *i*Pendant Operation screen.

Procedure F-5 Remote *i*Pendant Connection

Steps

1. Bring up Internet Explorer on the PC.
2. In the Internet Explorer Address field, type the following: `http://<myrobot_name_ or_address>/frh/cgtp/cgtp.htm`

   Where <myrobot_name_ or_address> is either the DNS name of your robot (i.e. pderob111.frc.com) or the IP address of your robot (i.e. 192.168.1.100).
3. If you have the HTTP Authentication for the *i*Pendant set to AUTHORIZE, the prompt shown in Figure F–5 will be displayed on the PC.

Figure F–5. Remote *i* Pendant Operation Password Screen



Type the USERNAME and PASSWORD that you set on the Robot Controller in the HOSTCOMM Setup>HTTP Authentication for the *i*Pendant.

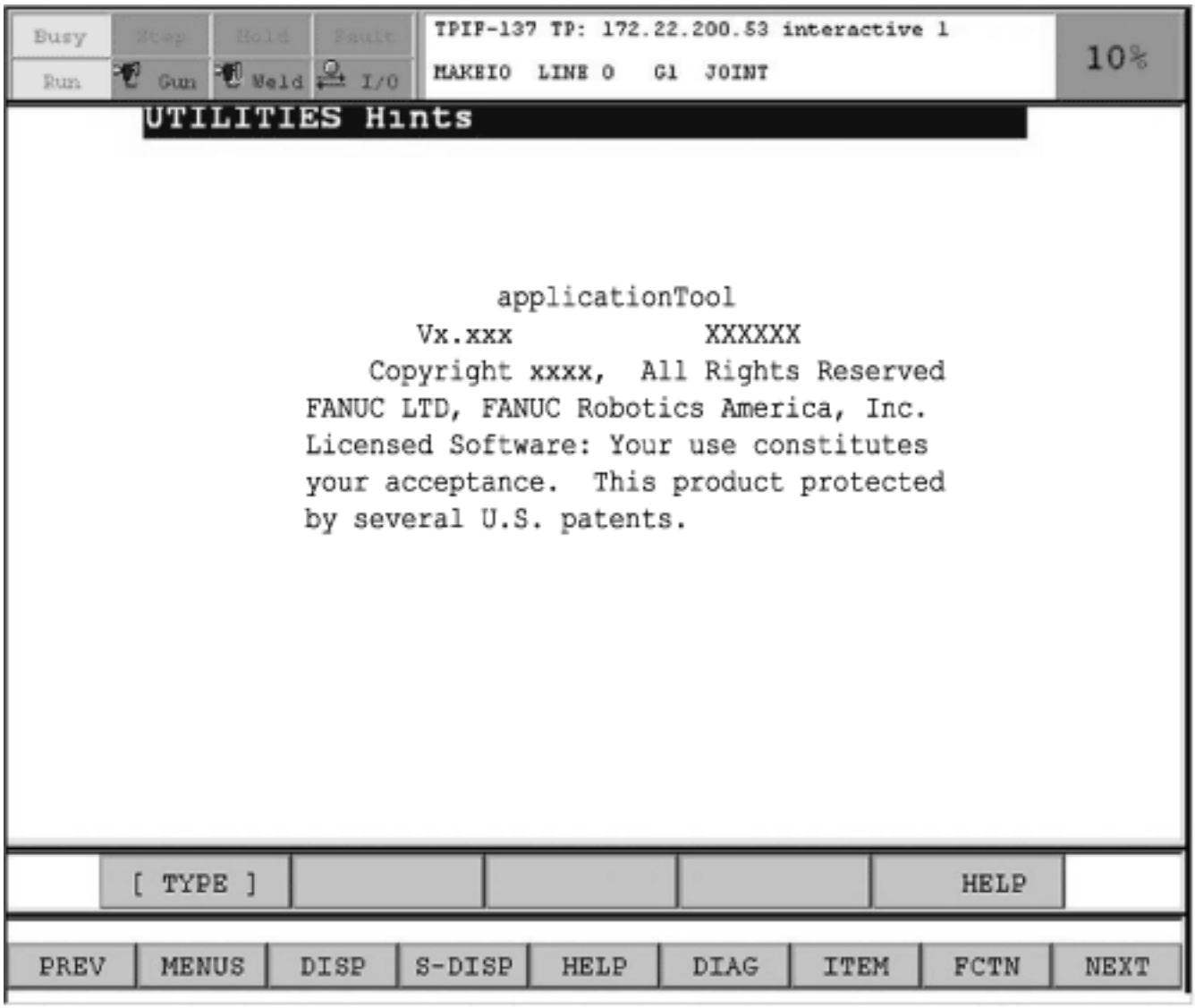
If the connection is successfully made you will briefly see the LOGIN screen, similar to that shown in Figure F–6 , displayed in Internet Explorer.

Figure F–6. Remote *i* Pendant LOGON Screen



If this is the first time you are logging in, you will see an *i*Pendant screen similar to the one shown in Figure F–7 If you have logged in before you might see a different screen configuration, depending on how your controller was set up.

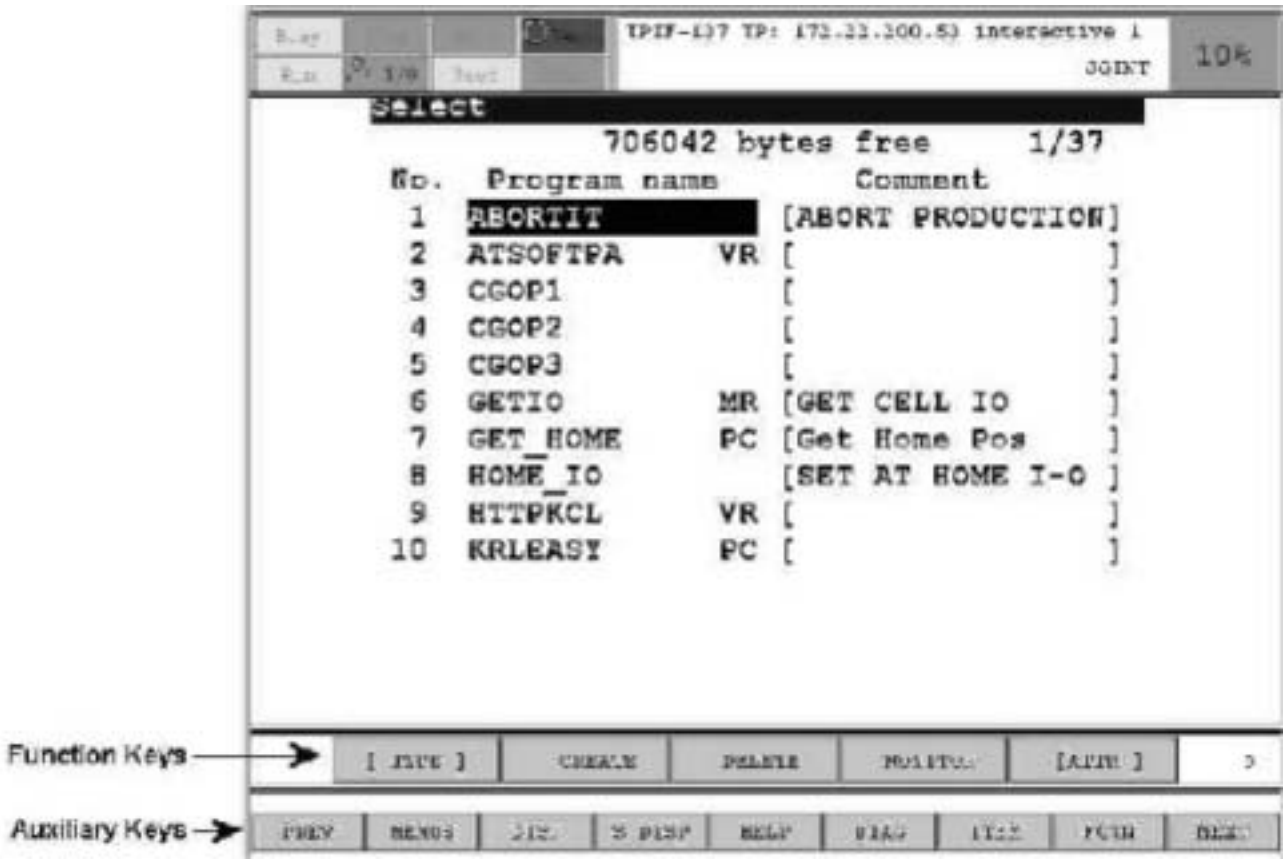
Figure F–7. Remote *i*Pendant Operation Screen



F.4.3.2 Keys

While in the Remote Operation Mode you can use your mouse to select any of the available function keys or the auxiliary keys available below the *i*Pendant screen as shown in Figure F–8

Figure F–8. Function Keys and Auxiliary Keys



These auxiliary keys allow you access to certain functions available as Hard Keys on the actual *i*Pendant. Some of these keys are also mapped to keys on the PC keyboard as shown in Table F–1 below:

Table F–1. Remote Operation Key Mapping

***i*Pendant Function Key**	**Description**	**PC Key Mapping**
PREV	This key moves to Previous Screen or Cancels a pending operation	ESC
MENUS	This key activates the MENUS popup	F9
DISP	This key activates the DISPLAY Menu	
S-DISP	This key changes the active window	
HELP	This key activates HELP for the current screen	
DIAG	Activates DIAGNOSTICS for the current Error	
ITEM	Allows you to select an item on the screen by entering an Item Number	

Table F–1. Remote Operation Key Mapping (Cont'd)

*i*Pendant Function Key	Description	PC Key Mapping
FCTN	Activates the FUNCTION Popup Menu	F12
NEXT	Moves to the next page of softkeys if they are available.	F6
	Move the cursor in the appropriate direction	UP, DOWN, LEFT and RIGHT ARROWS
	Page Up and Page Down	SHIFT+UP Arrow, SHIFT+DOWN Arrow, PAGE UP and PAGE DOWN

In addition mouse events are supported in most screens as shown in Table F–2 below:

Table F–2. Remote Operation Mouse Events

Mouse Event	Description
Page Up	Mouse click above the lines in any screen.
Page Down	Mouse click below the lines in any screen.
Select a line	Mouse click on a line.
Select an item	Mouse click on an item.
Select Links, or custom controls on Web pages, Help, Diagnostics, and any custom iPendant screen.	Use Mouse click to select.
Select a program	Double mouse click on the program.
When ENTER is a valid selection.	Double mouse click instead.
When CHOICE is a valid selection.	Right mouse click instead.
When PREV is a valid selection.	Mouse click to the left of the function keys.
When NEXT is a valid selection.	Mouse click on > to the right of the function keys.
To display Cause & Remedy for an alarm.	Double mouse click on an alarm in the Alarm Log screen.
To display DETAIL for an alarm.	Right mouse click on an alarm in the Alarm Log screen.

For further information on navigating *i*Pendant screens and operating the *i*Pendant, refer to Section F.1

F.4.3.3 Editing Guidelines

- LOOK/MONITOR mode is available in any window.
- Each window can have a unique default program.
- Selecting a teach pendant program from the SELECT screen in the window will cause that program to be the default program for that window.
- The window will use $UI_DEFPROG[3] to $UI_DEFPROG[8] based on the order of connection.
- The Status line will always show the default program for the left-hand window even if it does not have focus.
- The Editor title line shows the program that is being displayed.
- The program selected will be retained during cycle power. The current line number will not be retained during cycle power.
- The same program can be displayed in multiple windows. The cursor is independent.
- The Editor will not allow editing in the foreground. If you try to make a change to the program, then the warning "TPIF-069 Use background edit" is posted
- The program is open while in the EDIT screen; otherwise the program will be closed.
- Closing a window will cause the selected program to be closed.
- Disconnecting the Internet Explorer session will cause all the programs in all windows to be closed.
- FWD/BWD is not available from Internet Explorer

F.4.3.4 Background Editing Guidelines

- Background edit is supported in each window. There will be a background edit program unique for each window so multiple edit sessions will not affect each other.
- The background edit programs will be –BCKED4- through –BCKED9- based on the order of connection.
- On Internet Explorer, you have no control over what connection you get. However, if you time out and reconnect, the connection should be the same. If you edit at your desk and stay logged in, then go to the lab you will get a different connection and you will not be able to continue with the same background edit program while in the lab.
- When a program is selected for background edits, then the comment for the background edit program will show the selected program.
- The same program cannot be background edited in multiple windows. If you select the same program already being background edited in another window, then the warning " TPIF-167 Program already in background edit" is posted. You need to End_edit the background session in the other window first.
- Background edit programs are retained during cycle power.

- The three windows within the same connection will share the same copy and paste buffer but they will not share with other Internet Explorer connections. This provides the ability to copy and paste from one teach pendant program to another.
- Each window has its own undo and redo buffers so multiple edit sessions will not affect each other.

F.4.4 Limitations

Remote operation allows you to display and monitor any *i*Pendant screen and perform many *i*Pendant operations on a PC using Internet Explorer ™ .

The following limitations apply to the Remote Operation of the *i*Pendant screen:

- You cannot make any changes through this connection if read only is enabled. To allow changes, set $UI_CONFIG.$READONLY[2]=FALSE. Refer to Section 8.7 for information on setting system variables.
- Many operations that are available on the actual *i*Pendant are not allowed through the remote monitoring function. These include:
  - — Jogging the Robot
  - — Running a Program using the FWD/BWD Keys
  - — Changing COORD
  - — Changing Speed Override
  - — Foreground editing a program (Only background editing is allowed)
  - — Functions associated with the application-specific keys on the *i*Pendant such as WIRE+/-, BACKUP, and so forth. Refer to the *Teach Pendant Keys* Section in the Preface of this manual for more information.

- Several of the *i*Pendant screens can only be displayed by a single device at any given time. The actual *i*Pendant always has precedence. If the *i*Pendant is currently displaying a screen that the Remote Operator has requested, and that screen can only be displayed in one place, the error "TPIF 110– Screen used by other device", will be posted and the screen on the remote device will not be changed. Also, if the remote device is currently displaying a screen that is requested by the actual *i*Pendant operator, the remote display will be forced to the UTILITIES>HINTS screen. However, an error will not be posted.
- The remote display will look similar to the *i*Pendant but will not be an exact duplicate. Fonts, Character sizes, colors, images, and overall format might be different.

F.5 Screen Customization

F.5.1 Overview

The *i*Pendant screen can be customized using the following tools:

- EasyPanel environment
- PanelWizard
- HMI screen customizations
- Displaying Web Page Macros
- Customizing User Menus

F.5.2 Easy Panel Environment Setup

F.5.2.1 Overview

The *i* Pendant can be easily customized to fit your needs. These custom screens can be used to display data, variables, I/O, or other system information specific to your application. The FANUC Robotics EasyPanel environment is used for this development, and consists of a set of custom "OBJECTS" or "CONTROLS" that are specially designed to work with the *i*Pendant, the FANUC Robotics Robot Controller, and the Microsoft® FrontPage ™ Web development environment.

F.5.2.2 Requirements

The following items are required to develop and run custom screens on the *i*Pendant in the EasyPanel environment:

- FANUC Robotics *i*Pendant Controls installed on the PC .
- Microsoft® Frontpage 2000 ™ or greater installed on the PC.
- The IPCC (Internet Protocol Connectivity and Customization) Option loaded on the controller if you want to use Remote Operation of the *i*Pendant. Refer to Section F.4 for more information.

F.5.2.3 Configuring Your PC

In order to develop custom pages you will need to download and install the FANUC Robotics *i*Pendant Controls setup from the FANUC Robotics WebFTP site. Refer to Procedure F-1 for instructions.

After the controls are installed you can begin to create custom screens for the *i*Pendant. Review the *iPendant Customization Guide*. This guide will provide further instructions on setting up FrontPage, Using the *i*Pendant Controls, and loading and accessing your custom screens.

F.5.3 Panel Wizard

The Panel Wizard option allows you to use the controller to create up to four operation panels to be displayed on the *i* Pendant.

To find out more customization information perform Steps 1 through 7 of the "Installing FANUC Robotics iPendant Controls" procedure in your application-specific *Setup and Operations Manual*, then review the *iPendant Customization Guide*.

The *iPendant Customization Guide* will provide further instructions on setting up and accessing your custom screens while you are using the Panel Wizard. Use Procedure F-6 to set up the Panel Wizard.

Procedure F-6 Setting Up the Panel Wizard

1. Press MENUS.
2. Select BROWSER.
3. Press F1, [TYPE].
4. Select Panel Wizard. You will see a screen similar to the following.

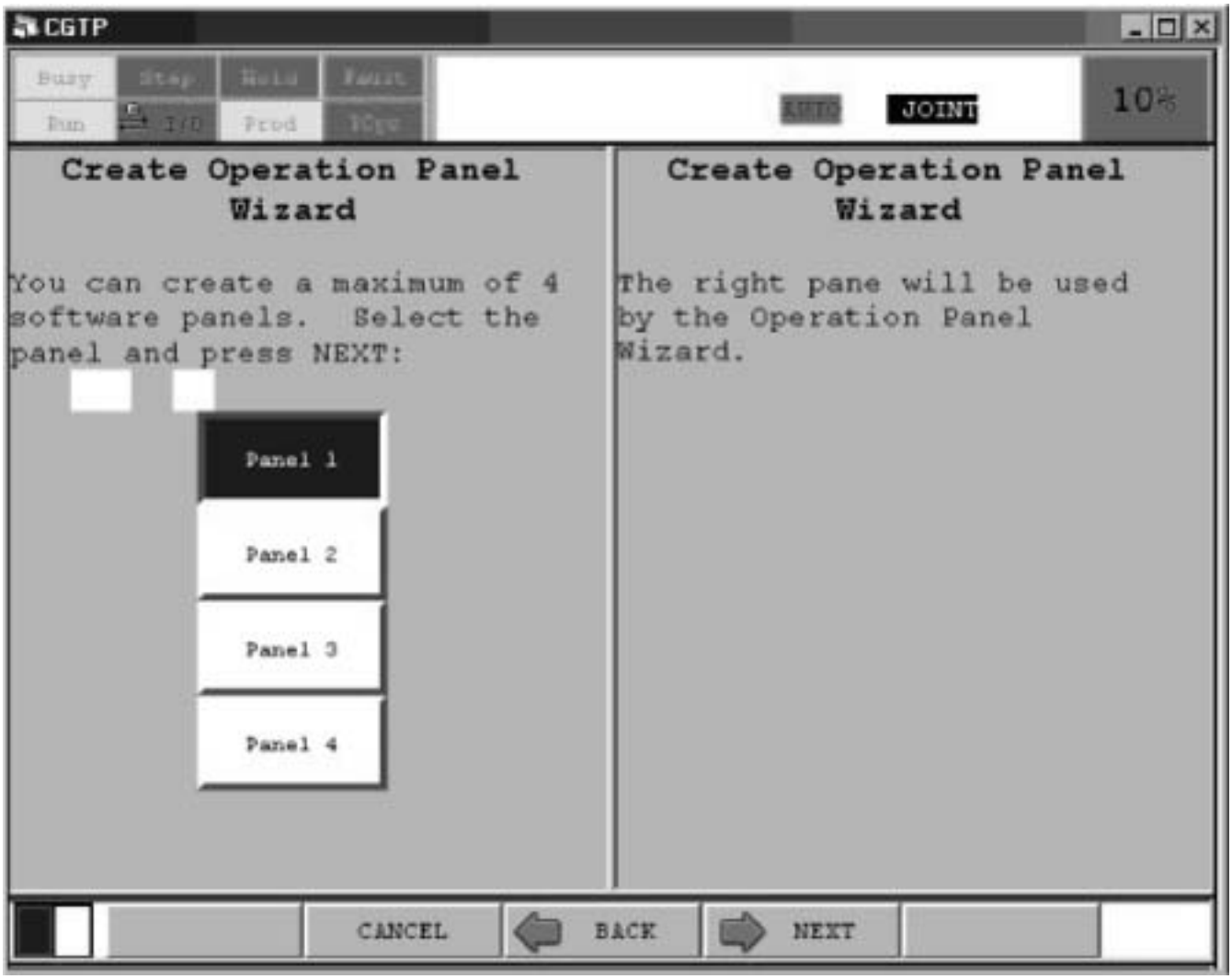
Figure F–9. Main Panel Wizard Screen



5. Select Start Wizard. You will see a screen similar to the following.

Note Double Panel mode will be displayed automatically.

Figure F–10. Create Operation Panel Wizard Screen

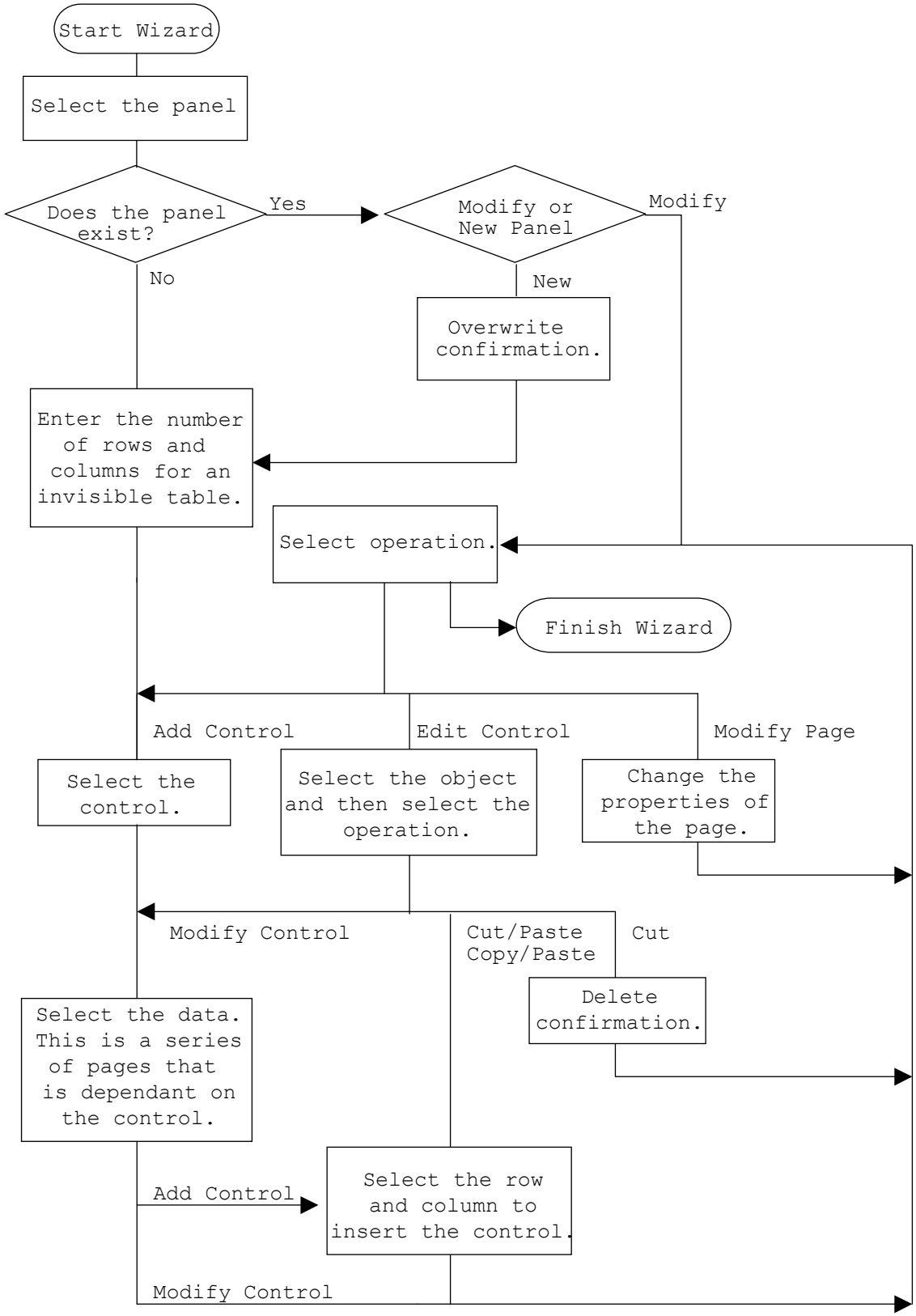


Note **After you make a selection on any screen**, press NEXT. **To cancel the Panel Wizard at any time**, press F2, CANCEL. **To display the previous screen at any time** , press F3, BACK.

6. Select the panel that you want to create. Refer to Figure F–11 for a flowchart overview of the steps required to create an operator panel. Refer to the *iPendant Customization Guide* for more information.

See Figure F–11 for a flowchart that describes the sequence of operations for using the Panel Wizard.

Figure F–11. Create Operation Panel Flowchart



F.5.4 *i*Pendant HMI Screen Customizations

When you are using HMI on the *i*Pendant, you can customize your teach pendant screens to fit your application. The HMI screen customization feature allows you to set the available menu options to display a specific set of HTM or STM menus. You can either display the default set of menus or, you can generate your own set of .HTM or .STM menus to display.

The user menus are stored in FRH:CGTP\USRHMIQK.HTM and FRH:CGTP\USRHMIFL.HTM. For remote connections the default files are FRH:CGTP\REMHMIQK.HTM and FRH:CGTP\REMHMIFL.HTM. The default pages that are created each time the controller is turned on are FRH:CGTP\IPHMIQK.HTM and FRH:CGTP\IPHMIFL.HTM. Typically, the operator does not have access to the FRH:CGTP\ device so that the files cannot be accidentally modified or deleted.

User Procedure F-7 to create a user-defined HMI Full or Quick Menu. Use Procedure F-8 to define and set up HMI Full, Quick, and Remote Menus.

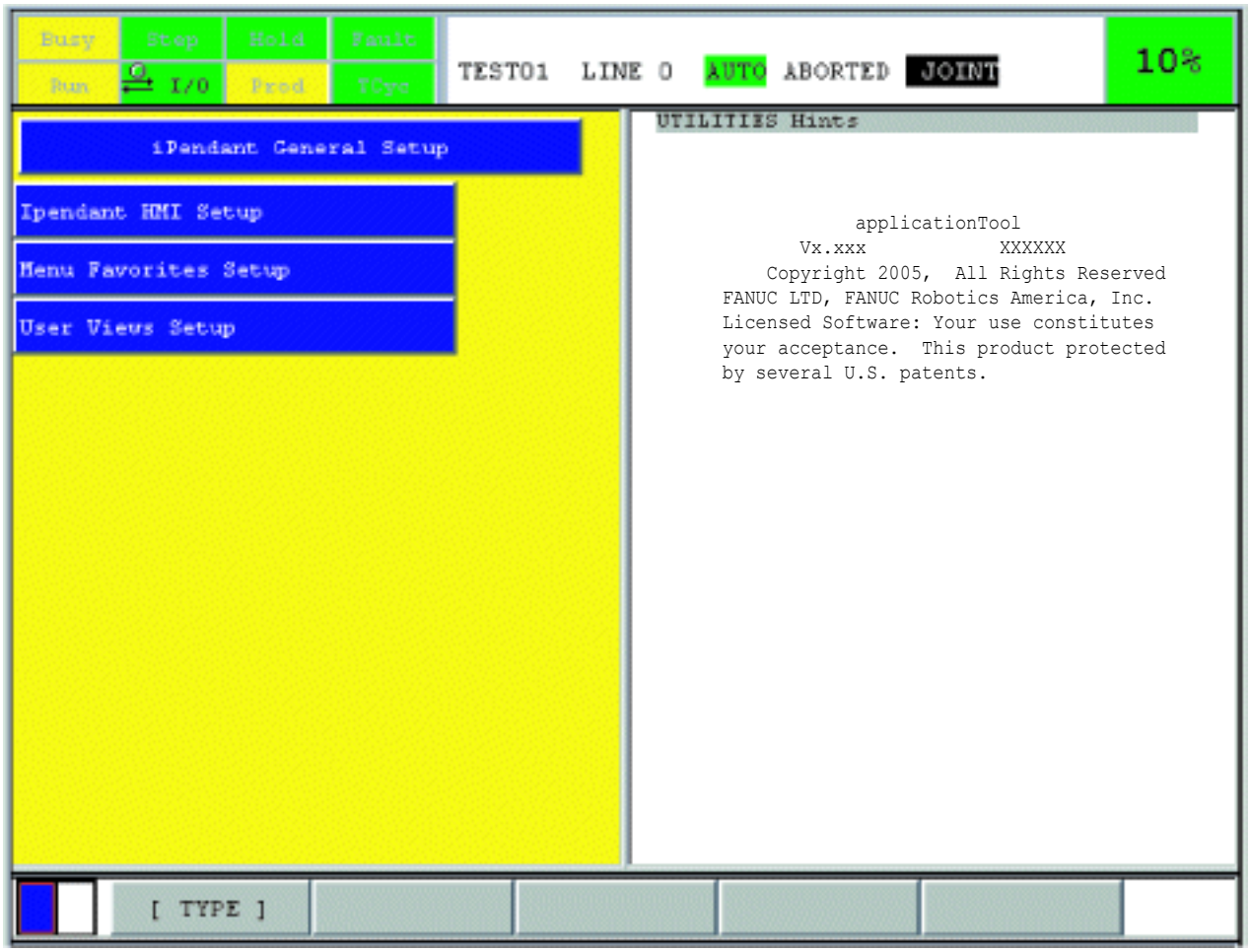
Procedure F-7 Creating a User-Defined HMI Full or Quick Menu

Conditions

- You are using the *i*Pendant.
- The default device is set appropriately. Refer to the "Program and File Manipulation Chapter" in the application-specific *Setup and Operations Manual* for more information.

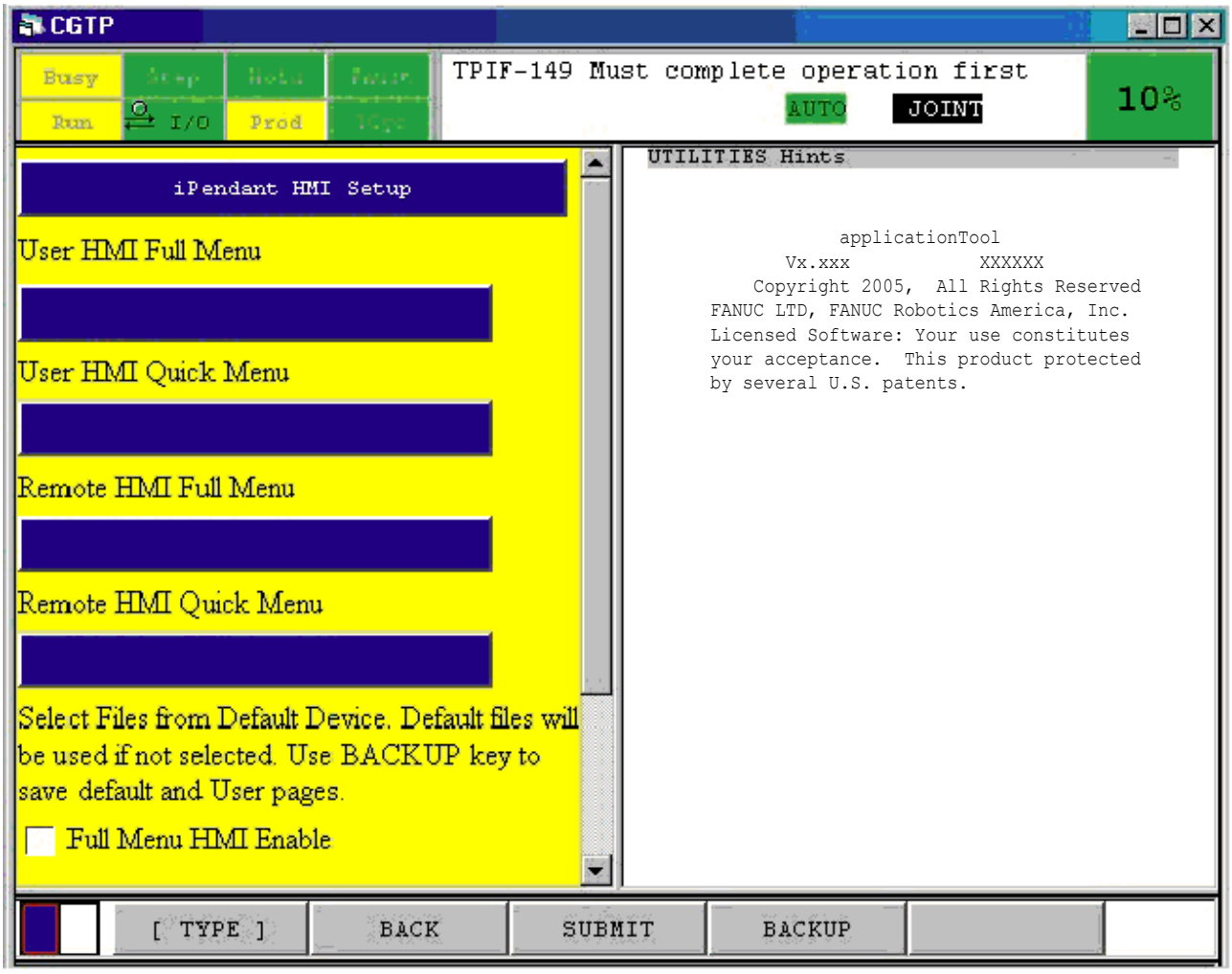
1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select iPendant Setup. You will see a screen similar to the following.

Figure F–12. *i*Pendant General Setup Screen



5. Select iPendant HMI Setup. You will see a screen similar to the following.

Figure F–13. *i*Pendant HMI Setup Menu



6. By default, the *i*Pendant HMI Full Menu is stored in IPHMIFL.HTM and the iPendant Quick Menu is stored in IPHMIQK.HTM. You can make a copy of these files and then edit them to make your own Full and Quick HMI menus. **To make a copy of the default files**, press F4, BACKUP. This will save a copy each of the files on to the default device.

7. Move the copied file to a personal computer so that you can edit the file.

> **Caution**
>
> In the HMI Full Menu file (IPHMIFL.HTM), you must have the *i*Pendant Setup as a menu item. In the HMI Quick Menu file (IPHMIQK.HTM), you must have Password Setup as a menu item if passwords are available on your controller. Make sure to retain these menu items as appropriate; otherwise, the generated menu files will not operate properly.

8. Add or remove menu items within this file as necessary.
9. Save the file with a new name if necessary.
10. Move the file back to the controller.
11. Use Procedure F-8 to set up and use the file you just created.

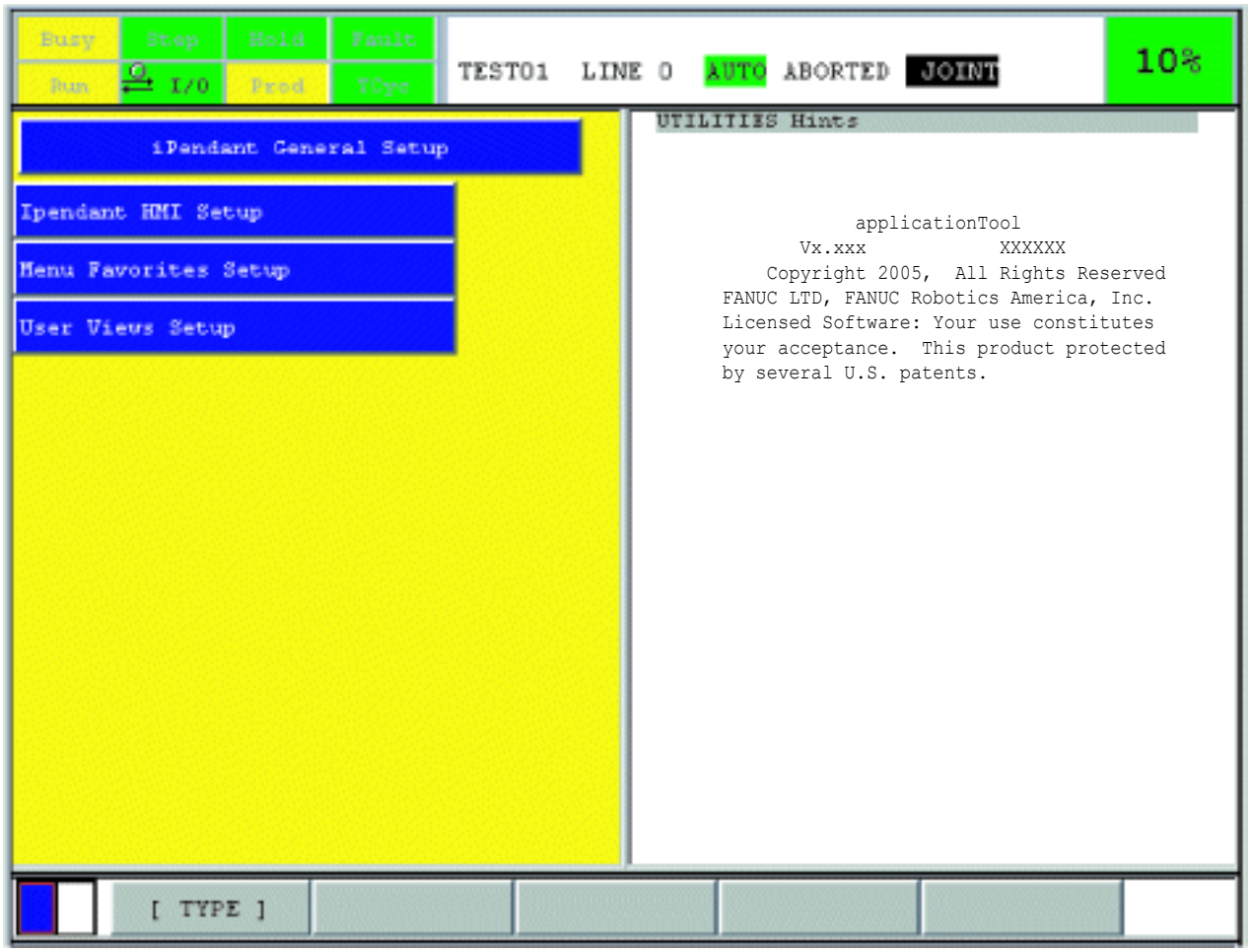
Procedure F-8 Setting up the *i*Pendant HMI Full, Quick, and Remote Menu Selections

Conditions

- You are using the *i*Pendant.

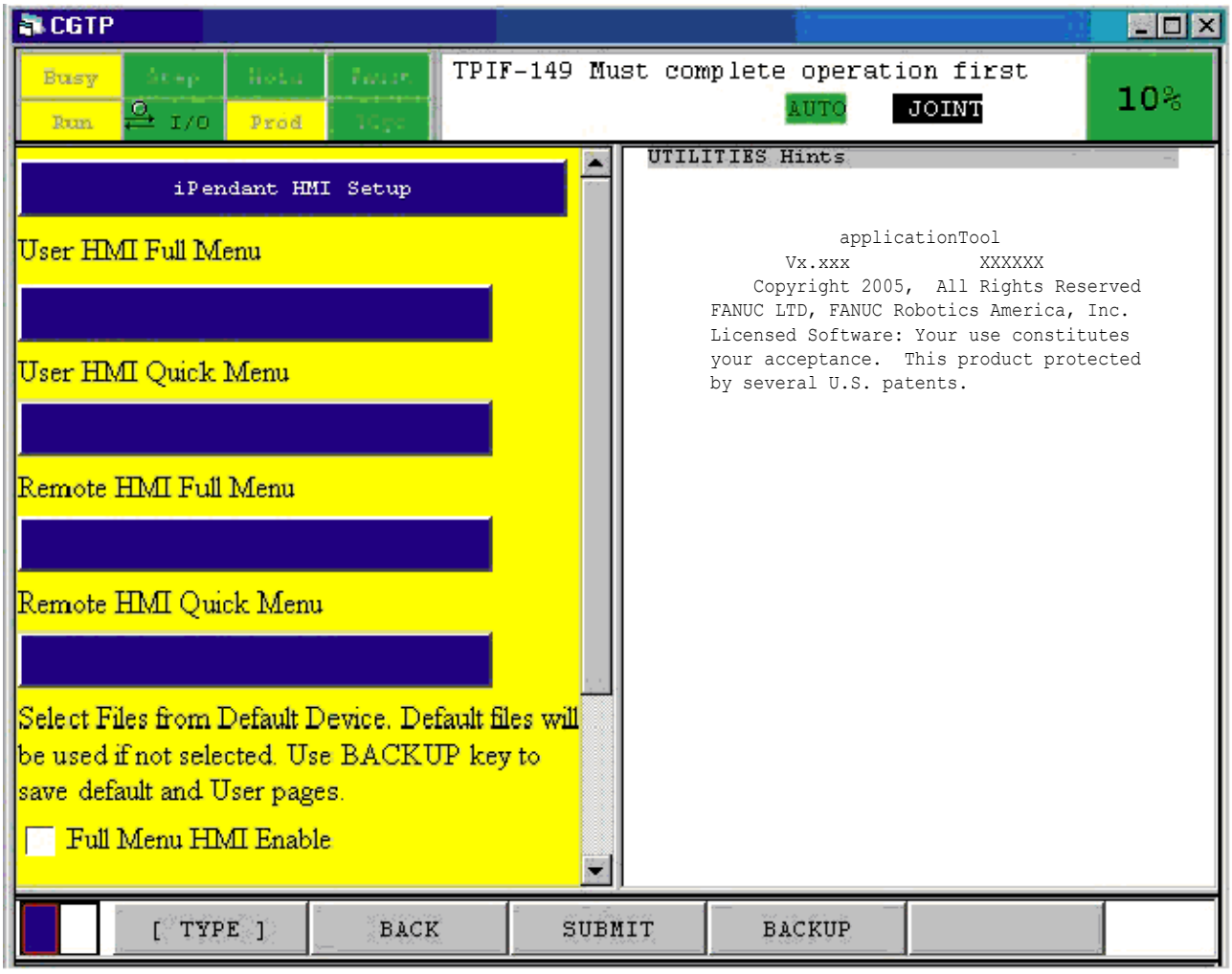
1. Press MENUS.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select iPendant Setup. You will see a screen similar to the following.

Figure F–14. *i*Pendant General Setup Screen



5. Select iPendant HMI Setup. You will see a screen similar to the following.

Figure F–15. *i*Pendant HMI Setup Screen



6. You can now select HTM or STM files to be displayed in place of Full Menus, Quick Menus, Remote Full Menus, and Remote Quick Menus.

a. To choose an HTM or STM file that resides on the default device on the controller, click the blue bar below the desired menu you want to set. A list of available HTM and STM files will be displayed.

b. Select the desired file, and press ENTER.

c. Continue selecting files for each Menu (User HMI Full, User HMI Quick, Remote HMI Full, and Remote HMI Quick).

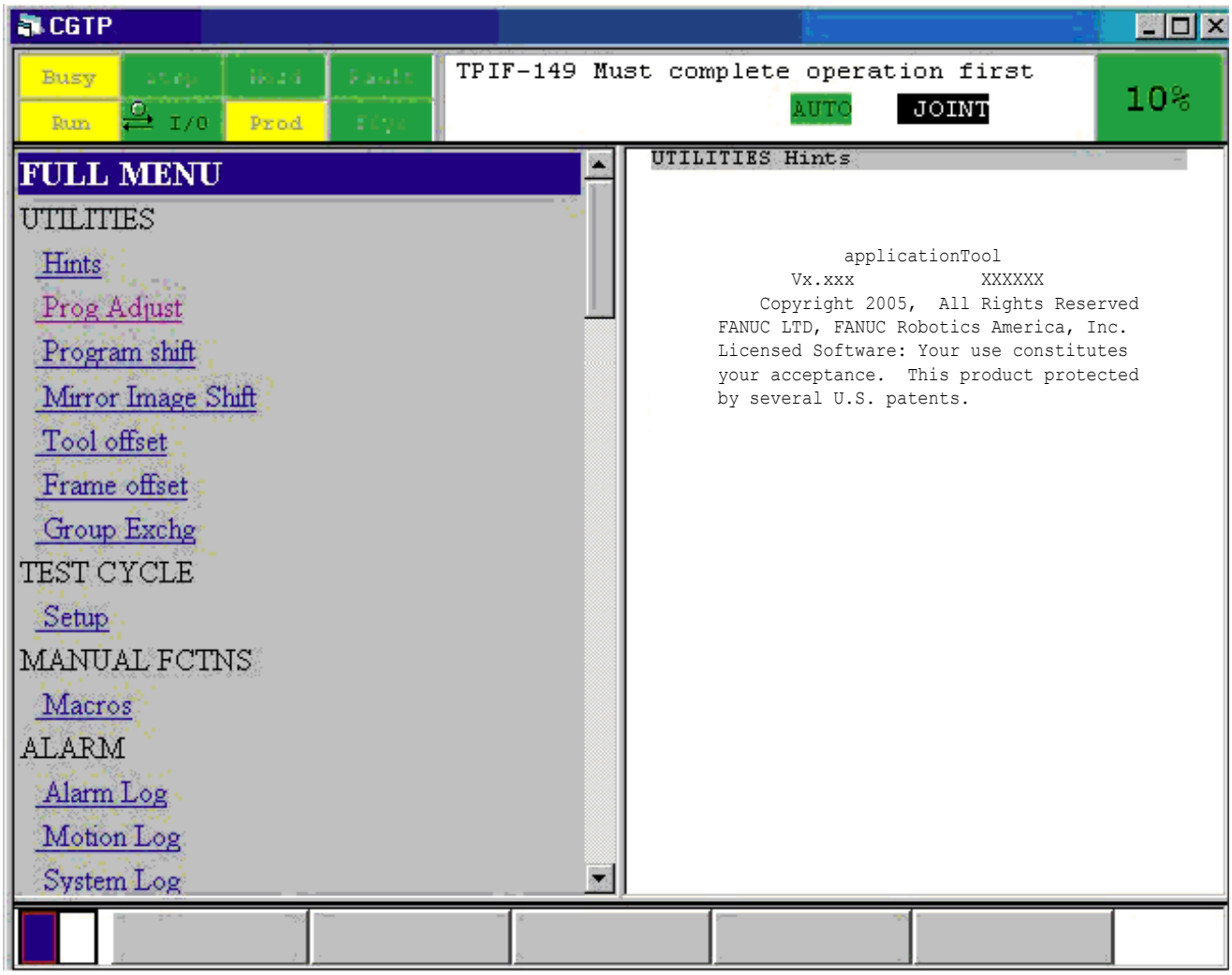
d. Press F3, SUBMIT. The names in the blue bar will be erased showing you that they have been set.

7. To activate the selected Full or Quick HMI Menus, or the Remote Full or Quick HMI Menu modes , click the box next to the selection at the bottom of the screen, then press F3,

SUBMIT. The check box will disappear after you press F3 to show you that the selection has been made.

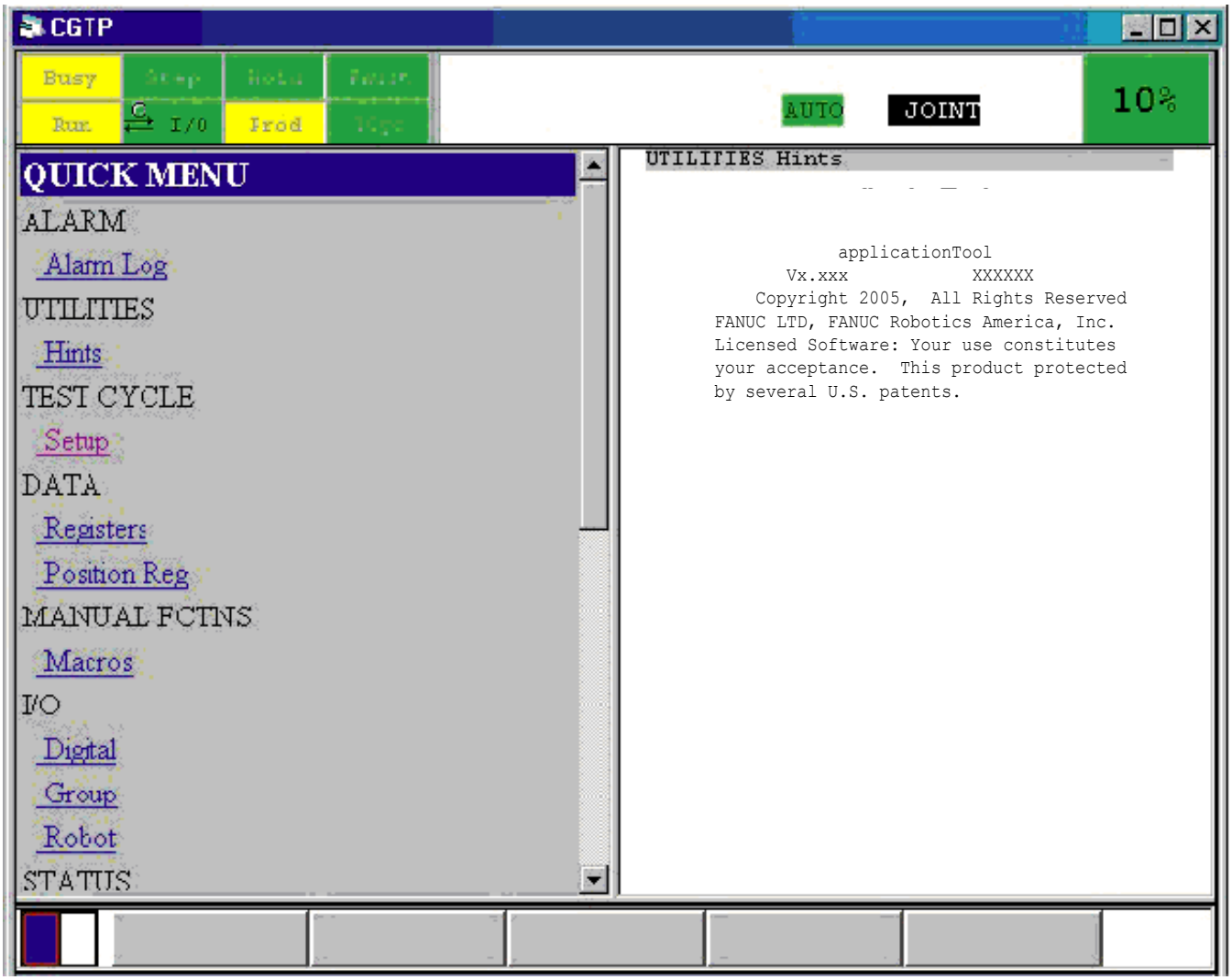
If you have set the HMI Full Menus, when you press MENUS, you will see a screen similar to the following.

Figure F–16. *i*Pendant HMI FULL Menu



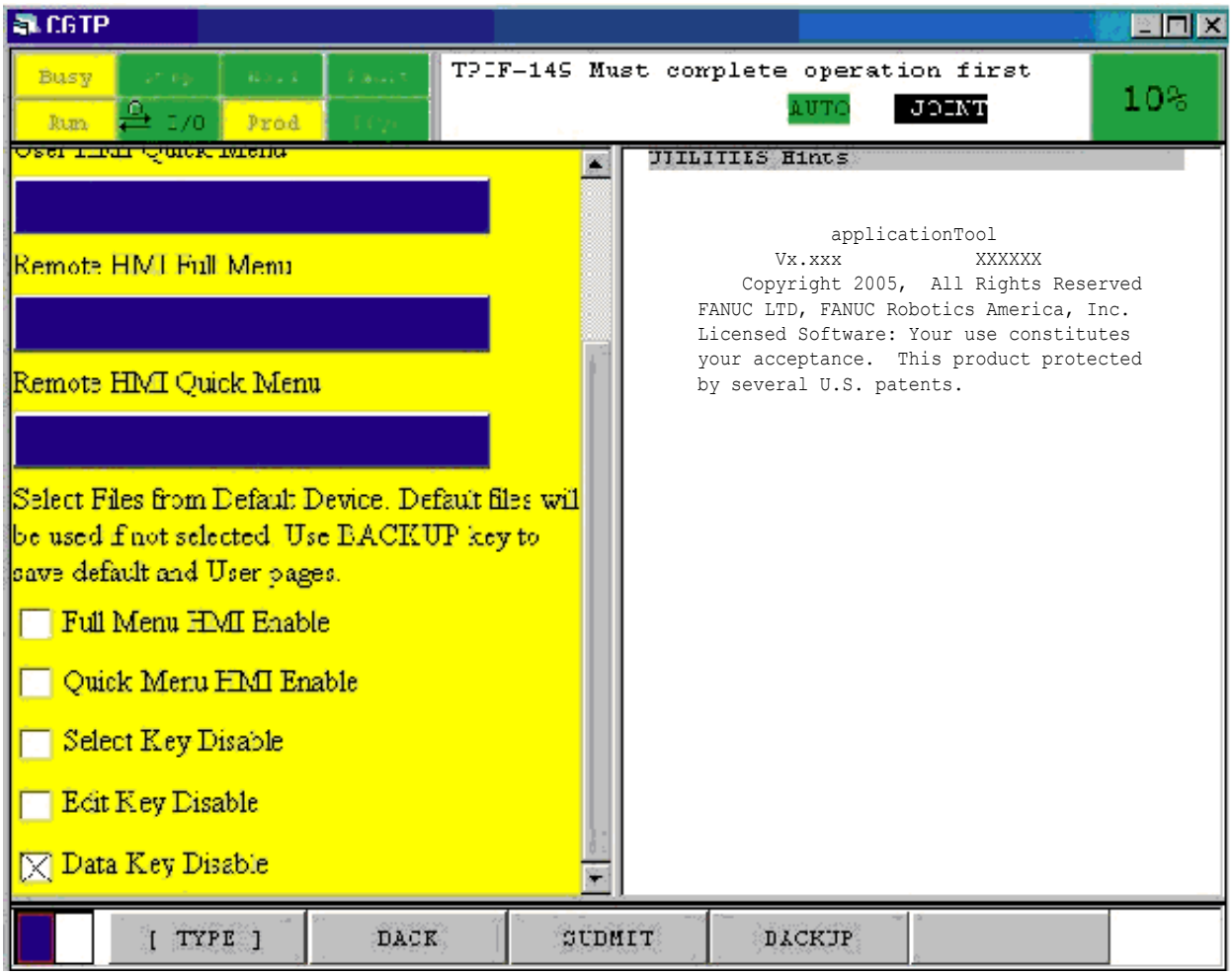
If you have set the HMI Quick Menus, when you press FCTN and select QUICK/FULL MENUS, you will see a screen similar to the following.

Figure F–17. *i*Pendant HMI QUICK Menu



8. **To deactivate the SELECT, EDIT, or DATA key during HMI mode,** scroll down then click the box next to the desired key name. See the following screen for an example.

Figure F–18. *i*Pendant HMI Setup Screen Disabling the DATA Key



When you have finished clicking the desired key, press F3, SUBMIT. The check box will disappear after you press F3 to show you that the selection has been made.

Note Each time the HMI Setup menu is displayed, check marks will be displayed showing you the current selection.

9. **To back up all HMI files to the default device,** press F4, BACKUP. This will copy all files from FRH:\CGTP\ to the default device using the overwrite function.
10. **To display the previous menu**, press F2, BACK.

F.5.5 Displaying Web Page Macros

HandlingTool provides a macro called DSP_WEBP.MR.

Users can call this macro with a parameter in their teach pendant programs.

```
Usage: CALL DSP_WEBP(1)  -- Displays the Favorites menu
CALL DSP_WEBP(2)  -- Displays the first user created web page in the [TYPE] pull-up
```

Parameters 2-11 are for displaying user-created web pages (created with or without the Panel Wizard).

The menu will be displayed only if you are using an *i* Pendant. It will always be displayed on the active pane on the *i* Pendant.

F.5.6 Customizing User Menus

F.5.6.1 Overview

The Custom User Menu Function allows you to add additional menus in most of the MENUS categories. You can use the Custom User Menu Function to:

- Run a teach pendant program
- Display a menu using a KAREL program
- Display a custom web page on the *i*Pendant

F.5.6.2 Running a Teach Pendant Program

To run a teach pendant program from a customized menu, you must set the following system variables:

- $CUSTOMMENU[*n*].$TITLE
- $CUSTOMMENU[*n*].$PROG_NAME

For example, to run a program called PERCHMOV by selecting an item on the EDIT menu called "Move to Perch," you would set the system variables as follows:

```
$CUSTOMMENU[3].$TITLE :            "Move to Perch"
$CUSTOMMENU[3].$PROG_NAME :        "PERCHMOV"
```

Note For more information on displaying and setting system variables, refer to Section 8.7 . Refer to Section F.5.6.5 for the system variable settings used for teach pendant programs.

F.5.6.3 Displaying a Menu Using a KAREL Program

You can use the Custom Menu Function to display a menu programmed using KAREL. The KAREL program must include the following processes:

- DISCTRL_FORM or DISCTRL_TBL can be used to display a form or table.
- The term mask should contain kc_f1_type. This will allow the KAREL form or table to process the F1, [TYPE] menu selection. The mask is defined in klevkmsk.kl.
- The F1 key should be labeled [TYPE].
- The KAREL program can remain in the form or table until the user selects a new menu. In this case the KAREL program should exit.
- Another option is for the KAREL program to call FORCE_SPMENU to display another menu.

You must also set the following system variables:

- $CUSTOMMENU[*n*].$TITLE
- $CUSTOMMENU[*n*].$PROG_NAME
- $CUSTOMMENU[n].$OPTION = SPI_SFMISC

SPI_SFMISC is 31

If running the KAREL program from the EDIT menu, you must call FORCE_SPMENU(device_stat, SPI_TPUSER2, 1) before writing to the screen. This forces the teach pendant into the USER2 menu.

For example, to run a program called PERCHSET by selecting an item from the UTILITY menu called “Set up Perch,” you would set the system variables as follows:

```
$CUSTOMMENU[10].$TITLE : "Set up Perch"
$CUSTOMMENU[10].$PROG_NAME : "PERCHSET"
$CUSTOMMENU[10].$OPTION: 31
```

Note For more information on displaying and setting system variables, refer to Section 8.7 . Refer to Section F.5.6.5 for the system variable settings used for KAREL programs.

F.5.6.4 Displaying a Custom Web Page on the *i*Pendant

You can use the Custom Menu Function to display a custom web page on the *i*Pendant. The 8.3 file extension of the web page is stored in $PROG_NAME. If $PROG_NAME contains a .htm or .stm, then the system will look for the file on the FR: device (customer), or on FRH:CGTP (application).

If it exists, then the *i*Pendant will display the web page. The title specified in the web page within the title tags (<title> </title>) will be displayed in the focus bar on the *i*Pendant. The number of title characters displayed on the focus bar is limited to 33. If the specified title is longer than 33 characters, it will be truncated.

When the web page is displayed, it will run within its own menu so it has a unique sofpart ID and screen ID. The [TYPE] key will be available and you will remain in the MENU category. The web page will be displayed until a new MENU is selected.

For example, if you set the following system variables:

```
$CUSTOMMENU[10].$TITLE :       "Utility 1"
$CUSTOMMENU[10].$PROG_NAME :   "custom.stm"
```

then, in the UTILITIES Menu, "UTILITY 1" is displayed as a [TYPE] menu choice. If you select this item and the file custom.stm exists on the FR: device, then the web page will be displayed.

Note The monochrome teach pendant cannot be used to display custom Web pages. For more information on displaying and setting system variables, refer to Section 8.7 . Refer to Section F.5.6.5 for the system variable settings used for custom Web pages.

F.5.6.5 System Variable Settings

Table F–3 lists the system variable settings that can be used to set up custom user menus to run teach pendant programs, run KAREL programs, and display custom Web pages.

Table F–3. System Variable Settings

System Variable	Top Level Menu	Softpart ID	Screen ID
Used for Teach Pendant or KAREL programs Only			
$CUSTOMMENU[1]	EDIT [INST]	64	1
$CUSTOMMENU[2]	EDIT [INST]	64	1
$CUSTOMMENU[3]	EDIT F2	64	1
$CUSTOMMENU[4]	EDIT F3	64	1
$CUSTOMMENU[5]	EDIT F4	64	1
Used for teach pendant, KAREL programs, or for Custom Web Pages			
$CUSTOMMENU[6]	FCTN	31	246
$CUSTOMMENU[7]	FCTN	31	247

Table F–3. System Variable Settings (Cont'd)

System Variable	Top Level Menu	Softpart ID	Screen ID
$CUSTOMMENU[10]	UTILITIES	31	250
$CUSTOMMENU[11]	UTILITIES	31	251
$CUSTOMMENU[12]	TEST CYCLE	31	252
$CUSTOMMENU[13]	MANUAL FCTNS	31	253
$CUSTOMMENU[14]	MANUAL FCTNS	31	254
$CUSTOMMENU[15]	ALARM	31	240
$CUSTOMMENU[16]	ALARM	31	239
$CUSTOMMENU[17]	I/O	31	238
$CUSTOMMENU[18]	I/O	31	237
$CUSTOMMENU[19]	SETUP	31	236
$CUSTOMMENU[20]	SETUP	31	235
$CUSTOMMENU[21]	FILE	31	234
$CUSTOMMENU[22]	DATA	31	233
$CUSTOMMENU[23]	DATA	31	232
$CUSTOMMENU[24]	STATUS	31	231
$CUSTOMMENU[25]	STATUS	31	230
$CUSTOMMENU[26]	STATUS	31	229
$CUSTOMMENU[27]	STATUS	31	228
$CUSTOMMENU[28]	SYSTEM	31	227
Used for Custom Web Pages Only			
$TX_SCREEN[1]	BROWSER	381	10
$TX_SCREEN[2]	BROWSER	381	11
$TX_SCREEN[3]	BROWSER	381	12
$TX_SCREEN[4]	BROWSER	381	13
$TX_SCREEN[5]	BROWSER	381	14
$TX_SCREEN[6]	BROWSER	381	15
$TX_SCREEN[7]	BROWSER	381	16
$TX_SCREEN[8]	BROWSER	381	17
$TX_SCREEN[9]	BROWSER	381	18
$TX_SCREEN[10]	BROWSER	381	19

Glossary

A

abort

Abnormal termination of a computer program caused by hardware or software malfunction or operator cancellation.

absolute pulse code system

A positional information system for servomotors that relies on battery-backed RAM to store encoder pulse counts when the robot is turned off. This system is calibrated when it is turned on.

A/D value

An analog to digital-value. Converts a multilevel analog electrical system pattern into a digital bit.

AI

Analog input.

AO

Analog output.

alarm

The difference in value between actual response and desired response in the performance of a controlled machine, system or process. Alarm=Error.

algorithm

A fixed step-by-step procedure for accomplishing a given result.

alphanumeric

Data that are both alphabetical and numeric.

AMPS

Amperage amount.

analog

The representation of numerical quantities by measurable quantities such as length, voltage or resistance. Also refers to analog type I/O blocks and distinguishes them from discrete I/O blocks. Numerical data that can vary continuously, for example, voltage levels that can vary within the range of -10 to +10 volts.

AND

An operation that places two contacts or groups of contacts in series. All contacts in series control the resulting status and also mathematical operator.

ANSI

American National Standard Institute, the U.S. government organization with responsibility for the development and announcement of technical data standards.

APC

See absolute pulse code system.

APC motor

See servomotor.

application program

The set of instructions that defines the specific intended tasks of robots and robot systems to make them reprogrammable and multifunctional. You can initiate and change these programs.

arm

A robot component consisting of an interconnecting set of links and powered joints that move and support the wrist socket and end effector.

articulated arm

A robot arm constructed to simulate the human arm, consisting of a series of rotary motions and joints, each powered by a motor.

ASCII

Abbreviation for American Standard Code for Information Interchange. An 8-level code (7 bits plus 1 parity bit) commonly used for the exchange of data.

automatic mode

The robot state in which automatic operation can be initiated.

automatic operation

The time during which robots are performing programmed tasks through unattended program execution.

axis

1. A straight line about which a robot joint rotates or moves. 2. One of the reference lines or a coordinate system. 3. A single joint on the robot arm.

B

backplane

A group of connectors mounted at the back of a controller rack to which printed circuit boards are mated.

BAR

A unit of pressure equal to 100,000 pascals.

barrier

A means of physically separating persons from the restricted work envelope; any physical boundary to a hazard or electrical device/component.

battery low alarm

A programmable value (in engineering units) against which the analog input signal automatically is compared on Genius I/O blocks. A fault is indicated if the input value is equal to or less than the low alarm value.

baud

A unit of transmission speed equal to the number of code elements (bits) per second.

big-endian

The adjectives big-endian and little-endian refer to which bytes are most significant in multi-byte data types and describe the order in which a sequence of bytes is stored in a computer's memory. In a big-endian system, the most significant value in the sequence is stored at the lowest storage address (i.e., first). In a little-endian system, the least significant value in the sequence is stored first.

binary

A numbering system that uses only 0 and 1.

bit

Contraction of binary digit. 1. The smallest unit of information in the binary numbering system, represented by a 0 or 1. 2. The smallest division of a programmable controller word.

bps

Bits per second.

buffer

A storage area in the computer where data is held temporarily until the computer can process it.

bus

A channel along which data can be sent.

bus controller

A Genius bus interface board for a programmable controller.

bus scan

One complete communications cycle on the serial bus.

Bus Switching Module

A device that switches a block cluster to one bus or the other of a dual bus.

byte

A sequence of binary digits that can be used to store a value from 0 to 255 and usually operated upon as a unit. Consists of eight bits used to store two numeric or one alpha character.

C

calibration

The process whereby the joint angle of each axis is calculated from a known reference point.

Cartesian coordinate system

A coordinate system whose axes (x, y, and z) are three intersecting perpendicular straight lines. The origin is the intersection of the axes.

Cartesian coordinates

A set of three numbers that defines the location of a point within a rectilinear coordinate system and consisting of three perpendicular axes (x, y, z).

cathode ray tube

A device, like a television set, for displaying information.

central processing unit

The main computer component that is made up of a control section and an arithmetic-logic section. The other basic units of a computer system are input/output units and primary storage.

channel

The device along which data flow between the input/output units of a computer and primary storage.

character

One of a set of elements that can be arranged in ordered groups to express information. Each character has two forms: 1. a man-intelligible form, the graphic, including the decimal digits 0-9, the letters A-Z, punctuation marks, and other formatting and control symbols; 2. a computer intelligible form, the code, consisting of a group of binary digits (bits).

circular

A MOTYPE option in which the robot tool center point moves in an arc defined by three points. These points can be positions or path nodes.

clear

To replace information in a storage unit by zero (or blank, in some machines).

closed loop

A control system that uses feedback. An open loop control system does not use feedback.

C-MOS RAM

Complementary metal-oxide semiconductor random-access memory. A read/write memory in which the basic memory cell is a pair of MOS (metal-oxide semiconductor) transistors. It is an implementation of S-RAM that has very low power consumption, but might be less dense than other S-RAM implementations.

coaxial cable

A transmission line in which one conductor is centered inside and insulated from an outer metal tube that serves as the second conductor. Also known as coax, coaxial line, coaxial transmission line, concentric cable, concentric line, concentric transmission line.

component

An inclusive term used to identify a raw material, ingredient, part or subassembly that goes into a higher level of assembly, compound or other item.

computer

A device capable of accepting information, applying prescribed processes to the information, and supplying the results of these processes.

configuration

The joint positions of a robot and turn number of wrist that describe the robot at a specified position. Configuration is designated by a STRING value and is included in positional data.

continuous path

A trajectory control system that enables the robot arm to move at a constant tip velocity through a series of predefined locations. A rounding effect of the path is required as the tip tries to pass through these locations.

continuous process control

The use of transducers (sensors) to monitor a process and make automatic changes in operations through the design of appropriate feedback control loops. While such devices historically have been mechanical or electromechanical, microcomputers and centralized control is now used, as well.

continuous production

A production system in which the productive equipment is organized and sequenced according to the steps involved to produce the product. Denotes that material flow is continuous during the production process. The routing of the jobs is fixed and set-ups are seldom changed.

controlled stop

A controlled stop controls robot deceleration until it stops. When a safety stop input such as a safety fence signal is opened, the robot decelerates in a controlled manner and then stops. After the robot stops, the Motor Control Contactor opens and drive power is removed.

controller

A hardware unit that contains the power supply, operator controls, control circuitry, and memory that directs the operation and motion of the robot and communications with external devices. See control unit.

controller memory

A medium in which data are retained. Primary storage refers to the internal area where the data and program instructions are stored for active use, as opposed to auxiliary or external storage (magnetic tape, disk, diskette, and so forth.)

control, open-loop

An operation where the computer applies control directly to the process without manual intervention.

control unit

The portion of a computer that directs the automatic operation of the computer, interprets computer instructions, and initiates the proper signals to the other computer circuits to execute instructions.

coordinate system

See Cartesian coordinate system.

CPU

See central processing unit.

CRT

See cathode ray tube.

cps (viscosity)

Centipoises per second.

CRT/KB

Cathode ray tube/keyboard. An optional interface device for the robot system. The CRT/KB is used for some robot operations and for entering programs. It can be a remote device that attaches to the robot via a cable.

cycle

1. A sequence of operations that is repeated regularly. The time it takes for one such sequence to occur. 2. The interval of time during which a system or process, such as seasonal demand or a manufacturing operation, periodically returns to similar initial conditions. 3. The interval of time during which an event or set of events is completed. In production control, a cycle is the length of time between the release of a manufacturing order and shipment to the customer or inventory.

cycle time

1. In industrial engineering, the time between completion of two discrete units of production. 2. In materials management, the length of time from when material enters a production facility until it exits. See throughput.

cursor

An indicator on a teach pendant or CRT display screen at which command entry or editing occurs. The indicator can be a highlighted field or an arrow (> or ^).

cylindrical

Type of work envelope that has two linear major axes and one rotational major axis. Robotic device that has a predominantly cylindrical work envelope due to its design. Typically has fewer than 6 joints and typically has only 1 linear axis.

D

D/A converter

A digital-to-analog converter. A device that transforms digital data into analog data.

D/A value

A digital-to-analog value. Converts a digital bit pattern into a multilevel analog electrical system.

daisy chain

A means of connecting devices (readers, printers, etc.) to a central processor by party-line input/output buses that join these devices by male and female connectors. The last female connector is shorted by a suitable line termination.

daisy chain configuration

A communications link formed by daisy chain connection of twisted pair wire.

data

A collection of facts, numeric and alphabetical characters, or any representation of information that is suitable for communication and processing.

data base

A data file philosophy designed to establish the independence of computer program from data files. Redundancy is minimized and data elements can be added to, or deleted from, the file designs without changing the existing computer programs.

DC

Abbreviation for direct current.

DEADMAN switch

A control switch on the teach pendant that is used to enable servo power. Pressing the DEADMAN switch while the teach pendant is on activates servo power and releases the robot brakes; releasing the switch deactivates servo power and applies the robot brakes.

debugging

The process of detecting, locating and removing mistakes from a computer program, or manufacturing control system. See diagnostic routine.

deceleration tolerance

The specification of the percentage of deceleration that must be completed before a motion is considered finished and another motion can begin.

default

The value, display, function or program automatically selected if you have not specified a choice.

deviation

Usually, the absolute difference between a number and the mean of a set of numbers, or between a forecast value and the actual data.

device

Any type of control hardware, such as an emergency-stop button, selector switch, control pendant, relay, solenoid valve, or sensor.

diagnostic routine

A test program used to detect and identify hardware/software malfunctions in the controller and its associated I/O equipment. See debugging.

diagnostics

Information that permits the identification and evaluation of robot and peripheral device conditions.

digital

A description of any data that is expressed in numerical format. Also, having the states On and Off only.

digital control

The use of a digital computer to perform processing and control tasks in a manner that is more accurate and less expensive than an analog control system.

digital signal

A single point control signal sent to or from the controller. The signal represents one of two states: ON (TRUE, 1. or OFF (FALSE, 0).

directory

A listing of the files stored on a device.

discrete

Consisting of individual, distinct entities such as bits, characters, circuits, or circuit components. Also refers to ON/OFF type I/O blocks.

disk

A secondary memory device in which information is stored on a magnetically sensitive, rotating disk.

disk memory

A non-programmable, bulk-storage, random-access memory consisting of a magnetized coating on one or both sides of a rotating thin circular plate.

drive power

The energy source or sources for the robot servomotors that produce motion.

DRAM

Dynamic Random Access Memory. A read/write memory in which the basic memory cell is a capacitor. DRAM (or D-RAM) tends to have a higher density than SRAM (or S-RAM). Due to the support circuitry required, and power consumption needs, it is generally impractical to use. A battery can be used to retain the content upon loss of power.

E

edit

1. A software mode that allows creation or alteration of a program. 2. To modify the form or format of data, for example, to insert or delete characters.

emergency stop

The operation of a circuit using hardware-based components that overrides all other robot controls, removes drive power from the actuators, and causes all moving parts of to stop. The operator panel and teach pendant are each equipped with EMERGENCY STOP buttons.

enabling device

A manually operated device that, when continuously activated, permits motion. Releasing the device stops the motion of the robot and associated equipment that might present a hazard.

encoder

1. A device within the robot that sends the controller information about where the robot is. 2. A transducer used to convert position data into electrical signals. The robot system uses an incremental optical encoder to provide position feedback for each joint. Velocity data is computed from the encoder signals and used as an additional feedback signal to assure servo stability.

end effector

An accessory device or tool specifically designed for attachment to the robot wrist or tool mounting plate to enable the robot to perform its intended tasks. Examples include gripper, spot weld gun, arc weld gun, spray paint gun, etc.

end-of-arm tooling

Any of a number of tools, such as welding guns, torches, bells, paint spraying devices, attached to the faceplate of the robot wrist. Also called end effector or EOAT.

engineering units

Units of measure as applied to a process variable, for example, psi, Degrees F., etc.

envelope, maximum

The volume of space encompassing the maximum designed movements of all robot parts including the end effector, workpiece, and attachments.

EOAT

See end of arm tooling, tool.

EPROM

Erasable Programmable Read Only Memory. Semiconductor memory that can be erased and reprogrammed. A non-volatile storage memory.

error

The difference in value between actual response and desired response in the performance of a controlled machine, system or process. Alarm=Error.

error message

A numbered message, displayed on the CRT/KB and teach pendant, that indicates a system problem or warns of a potential problem.

Ethernet

A Local Area Network (LAN) bus-oriented, hardware technology that is used to connect computers, printers, terminal concentrators (servers), and many other devices together. It consists of a master cable and connection devices at each machine on the cable that allow the various devices to "talk" to each other. Software that can access the Ethernet and cooperate with machines connected to the cable is necessary. Ethernets come in varieties such as baseband and broadband and can run on different media, such as coax, twisted pair and fiber. Ethernet is a trademark of Xerox Corporation.

execute

To perform a specific operation, such as one that would be accomplished through processing one statement or command, a series of statements or commands, or a complete program or command procedure.

extended axis

An optional, servo-controlled axis that provides extended reach capability for a robot, including in-booth rail, single- or double-link arm, also used to control motion of positioning devices.

F

faceplate

The tool mounting plate of the robot.

feedback

1. The signal or data fed back to a commanding unit from a controlled machine or process to denote its response to the command signal. The signal representing the difference between actual response and desired response that is used by the commanding unit to improve performance of the controlled machine or process. 2. The flow of information back into the control system so that actual performance can be compared with planned performance, for instance in a servo system.

field

A specified area of a record used for a particular category of data. 2. A group of related items that occupy the same space on a CRT/KB screen or teach pendant LCD screen. Field name is the name of the field; field items are the members of the group.

field devices

User-supplied devices that provide information to the PLC (inputs: push buttons, limit switches, relay contacts, an so forth) or perform PLC tasks (outputs: motor starters, solenoids, indicator lights, and so forth.)

file

1. An organized collection of records that can be stored or retrieved by name. 2. The storage device on which these records are kept, such as bubble memory or disk.

filter

A device to suppress interference that would appear as noise.

Flash File Storage

A portion of FROM memory that functions as a separate storage device. Any file can be stored on the FROM disk.

Flash ROM

Flash Read Only Memory. Flash ROM is not battery-backed memory but it is non-volatile. All data in Flash ROM is saved even after you turn off and turn on the robot.

flow chart

A systems analysis tool to graphically show a procedure in which symbols are used to represent operations, data, flow, and equipment. See block diagram, process chart.

flow control

A specific production control system that is based primarily on setting production rates and feeding work into production to meet the planned rates, then following it through production to make sure that it is moving. This concept is most successful in repetitive production.

format

To set up or prepare a memory card or floppy disk (not supported with version 7.20 and later) so it can be used to store data in a specific system.

FR

See Flash ROM.

F-ROM

See Flash ROM.

FROM disk

See Flash ROM.

G

general override stat

A percentage value that governs the maximum robot jog speed and program run speed.

Genius I/O bus

The serial bus that provides communications between blocks, controllers, and other devices in the system especially with respect to GE FANUC Genius I/O.

gripper

The "hand" of a robot that picks up, holds and releases the part or object being handled. Sometimes referred to as a manipulator. See EOAT, tool.

group signal

An input/output signal that has a variable number of digital signals, recognized and taken as a group.

gun

See applicator.

H

Hand Model.

Used in Interference Checking, the Hand Model is the set of virtual model elements (spheres and cylinders) that are used to represent the location and shape of the end of arm tooling with respect to the robot's faceplate.

hardware

1. In data processing, the mechanical, magnetic, electrical and electronic devices of which a computer, controller, robot, or panel is built. 2. In manufacturing, relatively standard items such as nuts, bolts, washers, clips, and so forth.

hard-wire

To connect electric components with solid metallic wires.

hard-wired

1. Having a fixed wired program or control system built in by the manufacturer and not subject to change by programming. 2. Interconnection of electrical and electronic devices directly through physical wiring.

hazardous motion

Unintended or unexpected robot motion that can cause injury.

hexadecimal

A numbering system having 16 as the base and represented by the digits 0 through 9, and A through F.

hold

A smoothly decelerated stopping of all robot movement and a pause of program execution. Power is maintained on the robot and program execution generally can be resumed from a hold.

HTML.

Hypertext Markup Language. A markup language that is used to create hypertext and hypermedia documents incorporating text, graphics, sound, video, and hyperlinks.

http.

Hypertext transfer protocol. The protocol used to transfer HTML files between web servers.

I

impedance

A measure of the total opposition to current flow in an electrical circuit.

incremental encoder system

A positional information system for servomotors that requires calibrating the robot by moving it to a known reference position (indicated by limit switches) each time the robot is turned on or calibration is lost due to an error condition.

index

An integer used to specify the location of information within a table or program.

index register

A memory device containing an index.

industrial robot

A reprogrammable multifunctional manipulator designed to move material, parts, tools, or specialized devices through variable programmed motions in order to perform a variety of tasks.

industrial robot system

A system that includes industrial robots, end effectors, any equipment devices and sensors required for the robot to perform its tasks, as well as communication interfaces for interlocking, sequencing, or monitoring the robot.

information

The meaning derived from data that have been arranged and displayed in a way that they relate to that which is already known. See data.

initialize

1. Setting all variable areas of a computer program or routine to their desired initial status, generally done the first time the code is executed during each run. 2. A program or hardware circuit that returns a program a system, or hardware device to an original state. See startup, initial.

input

The data supplied from an external device to a computer for processing. The device used to accomplish this transfer of data.

input device

A device such as a terminal keyboard that, through mechanical or electrical action, converts data from the form in which it has been received into electronic signals that can be interpreted by the CPU or programmable controller. Examples are limit switches, push buttons, pressure switches, digital encoders, and analog devices.

input processing time

The time required for input data to reach the microprocessor.

input/output

Information or signals transferred between devices, discreet electrical signals for external control.

input/output control

A technique for controlling capacity where the actual output from a work center is compared with the planned output developed by CRP. The input is also monitored to see if it corresponds with plans so that work centers will not be expected to generate output when jobs are not available to work on.

integrated circuit

A solid-state micro-circuit contained entirely within a chip of semiconductor material, generally silicon. Also called chip.

interactive

Refers to applications where you communicate with a computer program via a terminal by entering data and receiving responses from the computer.

interface

1. A concept that involves the specifications of the inter-connection between two equipments having different functions. 2. To connects a PLC with the application device, communications channel, and peripherals through various modules and cables. 3. The method or equipment used to communicate between devices.

interference zone

An area that falls within the work envelope of a robot, in which there is the potential for the robot motion to coincide with the motion of another robot or machine, and for a collision to occur.

interlock

An arrangement whereby the operation of one control or mechanism brings about, or prevents, the operations of another.

interrupt

A break in the normal flow of a system or program that occurs in a way that the flow can be resumed from that point at a later time. Interrupts are initiated by two types of signals: 1. signals originating within the computer system to synchronize the operation of the computer system with the outside

world; 2. signals originating exterior to the computer system to synchronize the operation of the computer system with the outside world.

I/O

Abbreviation for input/output or input/output control.

I/O block

A microprocessor-based, configurable, rugged solid state device to which field I/O devices are attached.

I/O electrical isolation

A method of separating field wiring from logic level circuitry. This is typically done through optical isolation devices.

I/O module

A printed circuit assembly that is the interface between user devices and the Series Six PLC.

I/O scan

A method by which the CPU monitors all inputs and controls all outputs within a prescribed time. A period during which each device on the bus is given a turn to send information and listen to all of the broadcast data on the bus.

ISO

The International Standards Organization that establishes the ISO interface standards.

isolation

1. The ability of a logic circuit having more than one inputs to ensure that each input signal is not affected by any of the others. 2. A method of separating field wiring circuitry from logic level circuitry, typically done optically.

item

1. A category displayed on the teach pendant on a menu. 2. A set of adjacent digits, bits, or characters that is treated as a unit and conveys a single unit of information. 3. Any unique manufactured or purchased part or assembly: end product, assembly, subassembly, component, or raw material.

J

jog coordinate systems

Coordinate systems that help you to move the robot more effectively for a specific application. These systems include JOINT, WORLD, TOOL, and USER.

JOG FRAME

A jog coordinate system you define to make the robot jog the best way possible for a specific application. This can be different from world coordinate frame.

jogging

Pressing special keys on the teach pendant to move the robot.

jog speed

Is a percentage of the maximum speed at which you can jog the robot.

joint

1. A single axis of rotation. There are up to six joints in a robot arm (P-155 swing arm has 8). 2. A jog coordinate system in which one axis is moved at a time.

JOINT

A motion type in which the robot moves the appropriate combination of axes independently to reach a point most efficiently. (Point to point, non-linear motion).

joint interpolated motion

A method of coordinating the movement of the joints so all joints arrive at the desired location at the same time. This method of servo control produces a predictable path regardless of speed and results in the fastest cycle time for a particular move. Also called joint motion.

K

K

Abbreviation for kilo, or exactly 1024 in computer jargon. Related to 1024 words of memory.

KAREL

The programming language developed for robots by the FANUC Robotics America, Inc.

L

label

An ordered set of characters used to symbolically identify an instruction, a program, a quantity, or a data area.

LCD

See liquid crystal display.

lead time

The span of time needed to perform an activity. In the production and inventory control context, this activity is normally the procurement of materials and/or products either from an outside supplier or from one's own manufacturing facility. Components of lead time can include order preparation time, queue time, move or transportation time, receiving and inspection time.

LED

See Light Emitting Diode.

LED display

An alphanumeric display that consists of an array of LEDs.

Light Emitting Diode

A solid-state device that lights to indicate a signal on electronic equipment.

limiting device

A device that restricts the work envelope by stopping or causing to stop all robot motion and that is independent of the control program and the application programs.

limit switch

A switch that is actuated by some part or motion of a machine or equipment to alter the electrical circuit associated with it. It can be used for position detection.

linear

A motion type in which the appropriate combination of axes move in order to move the robot TCP in a straight line while maintaining tool center point orientation.

liquid crystal display

A digital display on the teach pendant that consists of two sheets of glass separated by a sealed-in, normally transparent, liquid crystal material. Abbreviated LCD.

little-endian

The adjectives big-endian and little-endian refer to which bytes are most significant in multi-byte data types and describe the order in which a sequence of bytes is stored in a computer's memory. In a big-endian system, the most significant value in the sequence is stored at the lowest storage address (i.e., first). In a little-endian system, the least significant value in the sequence is stored first.

load

1. The weight (force) applied to the end of the robot arm. 2. A device intentionally placed in a circuit or connected to a machine or apparatus to absorb power and convert it into the desired useful form. 3. To copy programs or data into memory storage.

location

1. A storage position in memory uniquely specified by an address. 2. The coordinates of an object used in describing its x, y, and z position in a Cartesian coordinate system.

lockout/tagout

The placement of a lock and/or tag on the energy isolating device (power disconnecting device) in the off or open position. This indicates that the energy isolating device or the equipment being controlled will not be operated until the lock/tag is removed.

log

A record of values and/or action for a given function.

logic

A fixed set of responses (outputs) to various external conditions (inputs). Also referred to as the program.

loop

The repeated execution of a series of instructions for a fixed number of times, or until interrupted by the operator.

M

mA

See milliampere.

machine language

A language written in a series of bits that are understandable by, and therefore instruct, a computer. This is a "first level" computer language, as compared to a "second level" assembly language, or a "third level" compiler language.

machine lock

A test run option that allows the operator to run a program without having the robot move.

macro

A source language instruction from which many machine-language instructions can be generated.

magnetic disk

A metal or plastic floppy disk (not supported on version 7.10 and later) that looks like a phonograph record whose surface can store data in the form of magnetized spots.

magnetic disk storage

A storage device or system consisting of magnetically coated metal disks.

magnetic tape

Plastic tape, like that used in tape recorder, on which data is stored in the form of magnetized spots.

maintenance

Keeping the robots and system in their proper operating condition.

MC

See memory card.

mechanical unit

The robot arm, including auxiliary axis, and hood/deck and door openers.

medium

plural **media** . The physical substance upon which data is recorded, such as a memory card (or floppy disk which is not supported on version 7.10 and later).

memory

A device or media used to store information in a form that can be retrieved and is understood by the computer or controller hardware. Memory on the controller includes C-MOS RAM, Flash ROM and D-RAM.

memory card

A C-MOS RAM memory card or a flash disk-based PC card.

menu

A list of options displayed on the teach pendant screen.

message

A group of words, variable in length, transporting an item of information.

microprocessor

A single integrated circuit that contains the arithmetic, logic, register, control and memory elements of a computer.

microsecond

One millionth (0.000001) of a second

milliampere

One one-thousandth of an ampere. Abbreviated mA.

millisecond

One thousandth of a second. Abbreviated msec.

module

A distinct and identifiable unit of computer program for such purposes as compiling, loading, and linkage editing. It is eventually combined with other units to form a complete program.

motion type

A feature that allows you to select how you want the robot to move from one point to the next. MOTYPES include joint, linear, and circular.

mode

1. One of several alternative conditions or methods of operation of a device. 2. The most common or frequent value in a group of values.

N

network

1. The interconnection of a number of devices by data communication facilities. "Local networking" is the communications network internal to a robot."Global networking" is the ability to provide communications connections outside of the robot's internal system. 2. Connection of geographically separated computers and/or terminals over communications lines. The control of transmission is managed by a standard protocol.

non-volatile memory

Memory capable of retaining its stored information when power is turned off.

O

Obstacle Model.
Used in Interference Checking, the Obstacle Model is the set of virtual model elements (spheres, cylinders, and planes) that are used to represent the shape and the location of a given obstacle in space.

off-line
Equipment or devices that are not directly connected to a communications line.

off-line operations
Data processing operations that are handled outside of the regular computer program. For example, the computer might generate a report off-line while the computer was doing another job.

off-line programming
The development of programs on a computer system that is independent of the "on-board" control of the robot. The resulting programs can be copied into the robot controller memory.

offset
The count value output from a A/D converter resulting from a zero input analog voltage. Used to correct subsequent non-zero measurements also incremental position or frame adjustment value.

on-line
A term to describe equipment or devices that are connected to the communications line.

on-line processing
A data processing approach where transactions are entered into the computer directly, as they occur.

operating system
Lowest level system monitor program.

operating work envelope
The portion of the restricted work envelope that is actually used by the robot while it is performing its programmed motion. This includes the maximum the end-effector, the workpiece, and the robot itself.

operator
A person designated to start, monitor, and stop the intended productive operation of a robot or robot system.

operator box
A control panel that is separate from the robot and is designed as part of the robot system. It consists of the buttons, switches, and indicator lights needed to operate the system.

operator panel
A control panel designed as part of the robot system and consisting of the buttons, switches, and indicator lights needed to operate the system.

optional features
Additional capabilities available at a cost above the base price.

OR

An operation that places two contacts or groups of contacts in parallel. Any of the contacts can control the resultant status, also a mathematical operation.

orientation

The attitude of an object in space. Commonly described by three angles: rotation about x (w), rotation about y (p), and rotation about z (r).

origin

The point in a Cartesian coordinate system where axes intersect; the reference point that defines the location of a frame.

OT

See overtravel.

output

Information that is transferred from the CPU for control of external devices or processes.

output device

A device, such as starter motors, solenoids, that receive data from the programmable controller.

output module

An I/O module that converts logic levels within the CPU to a usable output signal for controlling a machine or process .

outputs

Signals, typically on or off, that controls external devices based upon commands from the CPU.

override

See general override.

overtravel

A condition that occurs when the motion of a robot axis exceeds its prescribed limits.

overwrite

To replace the contents of one file with the contents of another file when copying.

P

parity

The anticipated state, odd or even, of a set of binary digits.

parity bit

A binary digit added to an array of bits to make the sum of all bits always odd or always even.

parity check

A check that tests whether the number of ones (or zeros) in an array of binary digits is odd or even.

parity error

A condition that occurs when a computed parity check does not agree with the parity bit.

part

A material item that is used as a component and is not an assembly or subassembly.

pascal

A unit of pressure in the meter-kilogram-second system equivalent to one newton per square meter.

path

1. A variable type available in the KAREL system that consists of a list of positions. Each node includes positional information and associated data. 2. The trajectory followed by the TCP in a move.

PCB

See printed circuit board.

PC Interface

The PC Interface software uses Ethernet connections to provide file transfer protocol (FTP) functions, PC send macros, telnet interface, TCP/IP interface web server functions, and host communications.

pendant

See teach pendant.

PLC

See programmable logic controller or cell controller.

PMC

The programmable machine controller (PMC) functions provide a ladder logic programming environment to create PMC functions. This provides the capability to use the robot I/O system to run PLC programs in the background of normal robot operations. This function can be used to control bulk supply systems, fixed automation that is part of the robot workcell, or other devices that would normally require basic PLC controls.

printed circuit board

A flat board whose front contains slots for integrated circuit chips and connections for a variety of electronic components, and whose back is printed with electrically conductive pathways between the components.

production mode

See automatic mode.

program

1. A plan for the solution of a problem. A complete program includes plans for the transcription of data, coding for the computer, and plans for the absorption of the results into the system. 2. A sequence of instructions to be executed by the computer or controller to control a robot/robot system. 3. To furnish a computer with a code of instructions. 4. To teach a robot system a specific set of movements and instructions to do a task.

programmable controller

See programmable logic controller or cell controller.

programmable logic controller

A solid-state industrial control device that receives inputs from user-supplied control devices, such as switches and sensors, implements them in a precise pattern determined by ladder diagram-based programs stored in the user memory, and provides outputs for control of processes or user-supplied devices such as relays and motor starters.

Program ToolBox

The Program ToolBox software provides programming utilities such as mirror image and flip wrist editing capabilities.

protocol

A set of hardware and software interfaces in a terminal or computer that allows it to transmit over a communications network, and that collectively forms a communications language.

psi

Pounds per square inch.

Q

queue.

1. Waiting lines resulting from temporary delays in providing service. 2. The amount of time a job waits at a work center before set-up or work is performed on the job. See also job queue.

R

RAM

See Random Access Memory.

random access

A term that describes files that do not have to be searched sequentially to find a particular record but can be addressed directly.

Random Access Memory

1. Volatile, solid-state memory used for storage of programs and locations; battery backup is required. 2. The working memory of the controller. Programs and variable data must be loaded into RAM before the program can execute or the data can be accessed by the program.

range

1. A characterization of a variable or function. All the values that a function can possess. 2. In statistics, the spread in a series of observations. 3. A programmable voltage or current spectrum of values to which input or output analog signals can be limited.

RI

Robot input.

RO

Robot output.

read

To copy, usually from one form of storage to another, particularly from external or secondary storage to internal storage. To sense the meaning of arrangements of hardware. To sense the presence of information on a recording medium.

Read Only Memory

A digital memory containing a fixed pattern of bits that you cannot alter.

record

To store the current set or sets of information on a storage device.

recovery

The restoration of normal processing after a hardware or software malfunction through detailed procedures for file backup, file restoration, and transaction logging.

register

1. A special section of primary storage in a computer where data is held while it is being worked on.
2. A memory device capable of containing one or more computer bits or words.

remote/local

A device connection to a given computer, with remote devices being attached over communications lines and local devices attached directly to a computer channel; in a network, the computer can be a remote device to the CPU controlling the network.

repair

To restore robots and robot systems to operating condition after damage, malfunction, or wear.

repeatability

The closeness of agreement among the number of consecutive movements made by the robot arm to a specific point.

reset

To return a register or storage location to zero or to a specified initial condition.

restricted work envelope

That portion of the work envelope to which a robot is restricted by limiting devices that establish limits that will not be exceeded in the event of any reasonably foreseeable failure of the robot or its controls. The maximum distance the robot can travel after the limited device is actuated defines the restricted work envelope of the robot.

RIA

Robotic Industries Association Subcommittee of the American National Standards Institute, Inc.

robot

A reprogrammable multifunctional manipulator designed to move material, parts, tools, or specialized devices, through variable programmed motions for the performance of a variety of tasks.

Robot Model.

Used in Interference Checking, the Robot Model is the set of virtual model elements (sphere and cylinders) that are used to represent the location and shape of the robot arm with respect to the robot's base. Generally, the structure of a six axes robot can be accurately modeled as a series of cylinders and spheres. Each model element represents a link or part of the robot arm.

ROM

See Read Only Memory.

routine

1. A list of coded instructions in a program. 2. A series of computer instructions that performs a specific task and can be executed as often as needed during program execution.

S

saving data.

Storing program data in Flash ROM, to a floppy disk (not supported on version 7.10 and later), or memory card.

scfm

Standard cubic feet per minute.

scratch start

Allows you to enable and disable the automatic recovery function.

sensor

A device that responds to physical stimuli, such as heat, light, sound pressure, magnetism, or motion, and transmits the resulting signal or data for providing a measurement, operating a control or both. Also a device that is used to measure or adjust differences in voltage in order to control sophisticated machinery dynamically.

serial communication

A method of data transfer within a PLC whereby the bits are handled sequentially rather than simultaneously as in parallel transmission.

serial interface

A method of data transmission that permits transmitting a single bit at a time through a single line. Used where high speed input is not necessary.

Server Side Include (SSI)

A method of calling or "including" code into a web page.

servomotor
An electric motor that is controlled to produce precision motion. Also called a "smart" motor.

SI
System input.

signal
The event, phenomenon, or electrical quantity that conveys information from one point to another.

significant bit
A bit that contributes to the precision of a number. These are counted starting with the bit that contributes the most value, of "most significant bit", and ending with the bit that contributes the least value, or "least significant bit".

singulating
Separating parts into a single layer.

slip sheet
A sheet of material placed between certain layers of a unit load. Also known as tier sheet.

SO
System output.

specific gravity
The ratio of a mass of solid or liquid to the mass of an equal volume of water at 45C. You must know the specific gravity of the dispensing material to perform volume signal calibration. The specific gravity of a dispensing material is listed on the MSDS for that material.

SRAM
A read/write memory in which the basic memory cell is a transistor. SRAM (or S-RAM) tends to have a lower density than DRAM. A battery can be used to retain the content upon loss of power.

slpm
Standard liters per minute.

Standard Operator Panel (SOP).
A panel that is made up of buttons, keyswitches, and connector ports.

state
The on or off condition of current to and from and input or output device.

statement
See instruction.

storage device
Any device that can accept, retain, and read back one or more times. The available storage devices are SRAM, Flash ROM (FROM or F-ROM), floppy disks (not available on version 7.10 and later), memory cards, or a USB memory stick.

system variable

An element that stores data used by the controller to indicate such things as robot specifications, application requirements, and the current status of the system.

T

Tare

The difference between the gross weight of an object and its contents, and the object itself. The weight of an object without its contents.

TCP

See tool center point.

teaching

Generating and storing a series of positional data points effected by moving the robot arm through a path of intended motions.

teach mode

1. The mode of operation in which a robot is instructed in its motions, usually by guiding it through these motions using a teach pendant. 2. The generation and storage of positional data. Positional data can be taught using the teach pendant to move the robot through a series of positions and recording those positions for use by an application program.

teach pendant

1. A hand-held device used to instruct a robot, specifying the character and types of motions it is to undertake. Also known as teach box, teach gun. 2. A portable device, consisting of an LCD display and a keypad, that serves as a user interface to the KAREL system and attaches to the operator box or operator panel via a cable. The teach pendant is used for robot operations such as jogging the robot, teaching and recording positions, and testing and debugging programs.

telemetry

The method of transmission of measurements made by an instrument or a sensor to a remote location.

termination type

Feature that controls the blending of robot motion between segments.

tool

A term used loosely to define something mounted on the end of the robot arm, for example, a hand, gripper, or an arc welding torch.

tool center point

1. The location on the end-effector or tool of a robot hand whose position and orientation define the coordinates of the controlled object. 2. Reference point for position control, that is, the point on the tool that is used to teach positions. Abbreviated TCP.

TOOL Frame

The Cartesian coordinate system that has the position of the TCP as its origin to stet. The z-axis of the tool frame indicates the approach vector for the tool.

TP.

See teach pendant.

transducer

A device for converting energy from one form to another.

U

UOP

See user operator panel.

URL

Universal Resource Locator. A standard addressing scheme used to locate or reference files on web servers.

USB memory stick

The controller USB memory stick interface supports a USB 1.1 interface. The USB Organization specifies standards for USB 1.1 and 2.0. Most memory stick devices conform to the USB 2.0 specification for operation and electrical standards. USB 2.0 devices as defined by the USB Specification must be backward compatible with USB 1.1 devices. However, FANUC Robotics does not support any security or encryption features on USB memory sticks. The controller supports most widely-available USB Flash memory sticks from 32MB up to 1GB in size.

USER Frame

The Cartesian coordinate system that you can define for a specific application. The default value of the User Frame is the World Frame. All positional data is recorded relative to User Frame.

User Operator Panel

User-supplied control device used in place of or in parallel with the operator panel or operator box supplied with the controller. Abbreviated UOP .

V

variable

A quantity that can assume any of a given set of values.

variance

The difference between the expected (or planned) and the actual, also statistics definitions.

vision system

A device that collects data and forms an image that can be interpreted by a robot computer to determine the position or to “see” an object.

volatile memory

Memory that will lose the information stored in it if power is removed from the memory circuit device.

W

web server

An application that allows you to access files on the robot using a standard web browser.

warning device

An audible or visible device used to alert personnel to potential safety hazards.

work envelope

The volume of space that encloses the maximum designed reach of the robot manipulator including the end effector, the workpiece, and the robot itself. The work envelope can be reduced or restricted by limiting devices. The maximum distance the robot can travel after the limit device is actuated is considered the basis for defining the restricted work envelope.

write

To deliver data to a medium such as storage.

Index

A

Abort all programs by CSTOPI
  setting, 3–229
ABORT instruction, 6–183
accuracy enhancement, 15–3
  overview, 15–3
    See also ircalibration options
adaptive process control (APC), 10–7
  adding APC instructions to a program, 10–16
  APC instructions, 10–16
  overview, 10–7
  setting up input parameters, 10–9
  setting up map functions, 10–13
  setting up options, 10–15
  setting up output parameters, 10–10
adjusting a program, 7–44
  adjusting a program or a schedule, 7–44
  error handling, 7–52
  overview, 7–44
advanced iPendant functions
  iPendant controls installation, F–2
  overview, F–2
  remote monitoring, F–3
    limitations, F–9
    operation, F–6
    overview, F–3
    setup, F–3
  remote operation, F–11
    limitations, F–19
    overivew, F–11
    remote iPendant operation, F–12
    setup, F–11
  screen customization, F–20
    custom screen setup, F–20
    overview, F–20
alarm
  history, A–5
alarm history
  display, A–5
alarm log
  extended, A–3
all-point teaching for palletizing, 6–232
Allow Chg. Ovrd in Auto Mode
  setting, 3–231
Allow Force I/O in Auto Mode
  setting, 3–231
angle entry shift function, 10–17
application program, 1–66
arm load information, 3–209
ASCII save, 9–74
ASCII upload, 9–89
  example ASCII file, 9–95
  loading an ASCII teach pendant program from KCL, 9–93
  loading an ASCII teach pendant program from the teach pendant, 9–90
  overview, 9–89
  viewing ASCII upload errors, 9–94
AUTO, 1–14
Auto Display of Alarm Menu
  setting, 3–231
auto login function, 3–172
auto normal, *see* easy normal
auto singularity avoidance, 10–23
  cautions, 10–27
  compatibility, 10–26
  how it works, 10–25
  how to use, 10–30
  limitations, 10–26
  overview, 10–23
  system variable setup, 10–37
  troubleshooting, 10–37
Autoexec Program for Cold Start
  setting, 3–227
Autoexec program for Hot Start

setting, 3–227
automatic backup, 9–75
automatic collision recovery
collision recovery, automatic, 10–40
automatic mode, 1–14
automatic software update, 9–65
axes, 1–4
extended, 1–6, 1–21
extendend
sub groups, 2–13
major, 1–4
minor, 1–4
axis limits setup, 3–119

B

background editing, 5–48
modifying a program in the background, 5–52
overview, 5–48
process flow, 5–49
troubleshooting, 5–54
using iPendant, 5–51
background logic, 6–247
edge detection, counter, timer, 6–253
setting up, 6–250
backplane, 1–22
backup, C–22
automatic, 9–75
Basic Diagnostic Resources
cause and remedy, A–8
binary save, 9–74
boot software, C–28
brake on hold setup, 3–127
brake timers setup, 3–122
branching instructions, 6–139
conditional branching instructions, 6–142
label definition instruction LBL[x], 6–140
overview, 6–139
unconditional branching instructions, 6–140
browser requirements, 1–61
bump box, 10–59
overview, 10–59
schedules, 10–60
setting up I/O, 10–67

C

Cal-plate, 15–78, 15–109
calibration options, 15–3
*i*RCalibration TCP Set, 15–78
See also TCPCal
*i*RCalibration TCP Shift, 15–109
See also TCPCal
cause and remedy
Basic Diagnostic Resources screen, A–8
cell interface I/O signals, 4–84
input signals, 4–84
output signals, 4–85
overview, 4–83
setting up I/O signals, 4–87
center finder, 10–82
overview, 10–82
troubleshooting, 10–91
using, 10–82
circular motion, 1–20
clock status, 8–26
cold start, C–6
collision avoid, 16–3
collision guard, 10–91
collision guard instructions, 6–196
collision skip, 10–98
operation, 10–99
overview, 10–98
requirements, 10–98
common characters
truncating, 6–14, 9–31
communications, 1–17
serial, 1–17
condition monitor function, 10–103
condition handler program, 10–107
condition menu, 10–108
conditions, 10–107
monitor instructions, 10–106
monitor state, 10–105
monitors, 10–104
overview, 10–103
restrictions, 10–111
condition monitor instructions, 6–190
confirm
payload, 10–348
confirmation
position, 6–29
Constant Path, 6–72

controlled start, C–10
controller, 1–7
overview, 1–7
controller backplane, 1–22
controller backup, C–22
as images, C–22
restoring images, C–25
controller backup and restore
overview, 9–75
controller restore, C–22
controlling I/O, 4–68
forcing outputs, 4–68
simulating inputs and outputs, 4–69
conversion
PC files, 9–64
coordinate systems, 2–7
robot
jogging, 2–7
coordinates offset function, 10–117
how to use, 10–117
tool frame offset function, 10–121
user frame offset function, 10–125
creating and modifying a program
macro program, 5–47
creating and writing a new program, 5–22
CRT/KB, 1–17
keys, B–6
menus, B–5
overview, B–2
setup, B–3
configure terminal parameters, B–4
connect controller port, B–3
controller port, B–3
overview, B–3
program key codes for function keys, B–4
set communications parameters, B–3
CSTOPI for ABORT
setting, 3–228
current language setup, 3–129
custom I/O, 4–27

D

data
log file, 10–162
data monitor, 10–139
overview, 10–139
programming, 10–153
schedule, 10–148
setup, 10–142
DEADMAN switch, 1–29
Default Logical Command
setting, 3–230
Detect Fault_Reset Signal
setting, 3–229
detection schedule
iRCalibration Master Set, 15–14
*i*RCalibration Mastering, 15–49
development tools, C–42
diagnostic log, C–42
diagnostics, 1–48
See also help
iPendant, 1–48
direct entry method
jog frame, 3–69
tool frame, 3–22
user frame, 3–45
display
motion status, 6–30
displaying DVM, 10–162
displaying DVM logs, 10–162
distance before, 10–165
action, 10–169
adding distance before motion option, 10–177
alarms posted, 10–172
cautions and limitations, 10–180
distance value, 10–167
hold and resume, 10–173
overview, 10–165
power failure recovery, 10–177
program instruction, 10–167
resume after jogging, 10–176
single step execution, 10–173
specification, 10–165
system variables, 10–181
trigger condition, 10–170
distributed (model B) I/O
basic digital I/O units, 4–34
DIP switches, 4–31
overview, 4–29
user I/O, 4–36
DRAM, 1–23

DVM
Diagnostic Video Monitoring, 10–154
DVM logs, 10–162
dynamic-uframe, 15–158, 15–191

E

E-stop button, 1–30
easy normal utility, 10–183
execution, 10–189
overview, 10–183
using the easy normal utility, 10–184
Easy normal utility
setup, 10–187
EMERGENCY STOP, 7–3
EMERGENCY STOP button, 1–30
emergency stop devices, 1–17
Enable UI Signals
setting, 3–228
End-of-Arm Tooling, 1–5
enhanced mirror image option, 10–190
EOAT, 1–5
error code output setup overview, 3–179
error code properties, A–3
error message text, A–15
facility name and code, A–8
overview, A–3
severity descriptions, A–12
error codes
overview, A–2
error log, *see* alarm log
error recovery, 10–209
error recovery manual function, 10–224
features, 10–210
I/O interface, 10–212
I/O timing sequence, 10–227
limitations, 10–211
overview, 10–209
programming, 10–220
setup, 10–214
testing, 10–224
error recovery procedures, A–17
chain failure detection, A–22
hand breakage recovery, A–19
overtravel release, A–17
overview, A–17
pulse coder alarm recovery, A–20
error severity table, 3–174
modifying error severity, 3–174
overview, 3–174
Ethernet Device, 9–3, 9–7
executing macro commands, 3–113
executing multiple programs (multi-tasking), 10–232
dedicated I/O signals, 10–234
guidelines, 10–233
overview, 10–232
program number select (PNS) execution, 10–235
RUN program instruction execution, 10–237
single step program execution, 10–237
standard operator panel (SOP) cycle start execution, 10–234
synchronizing the execution of multiple programs, 10–234
execution history, 8–40
extended axes, 1–21, 2–13, 10–359
Extended axes, 1–6
extended error log, *see* alarm log
extended user frames, 10–239
jog menu, 10–240
External I/O
setting, 3–234

F

family members, 6–22
FANUC sensor interface
communication protocol, 12–2
data format, 12–3
flow control mechanism, 12–2
handshaking, 12–5
hardware connections, 12–9
overview, 12–2
programming, 12–8
sensor setup, 12–9
fault and incident reporting, *see* reporting log
filtered memory device, 9–3, 9–8
FIRP, *see* reporting log
flag, 6–252
flash file storage disk, 1–23
Flash File Storage disk (FR), 9–3, 9–5
Flash Rom (F-ROM), 1–23

FMD device, *see* filtered memory device
focus
windows, 1–41
Force Message
setting, 3–231
four point method
user frame setup, 3–39
frame data
saving, 3–74
frames
extended user, 10–239
frames setup, 3–3
overview, 3–3
tool frame, 3–6

G

general setup, 3–3
group mask exchange, 10–244
overview, 10–244
setting up, 10–245
troubleshooting, 10–246
guidelines
iPendant
editing, 1–41
operational, 1–60

H

Hand Broken
setting, 3–233
hardware diagnosis
hot start, C–16
head of family, 6–22
help, 1–48
See also diagnostics
iPendant, 1–48
help and diagnostics, 1–49
operations, 1–49
HELP/DIAG key, A–8
high-speed skip, 6–162
history,
alarm, A–5
HMI
iPendant screen customizations, F–25
HOLD, 7–5
hot start, C–3
HOT START done signal
setting, 3–228

I

I/O, 1–18, 4–10 to 4–11
See also I/O setup ; I/O setup ; I/O setup
controlling, 4–68
controlling outputs, 4–10
custom, 4–27
memory image port assignment, 4–11
model B, 4–29
robot I/O, 4–37
simulating I/O, 4–10
SOP I/O, 4–41
touch sensing, 18–4
valve, 13–17
I/O instructions, 6–134
analog input and output instructions, 6–137
digital input and output instructions, 6–134
group input and output instructions, 6–138
overview, 6–134
robot digital input and output instructions, 6–136
I/O interconnect
setup, 4–62
I/O link connection, 4–50
FANUC, 4–52
I/O link device screen, 4–47
I/O link devices, 4–45
model A I/O, 4–46
model B I/O, 4–47
overview, 4–45
ports, 4–60
process I/O, 4–45
I/O Power Fail Recovery
setting, 3–226
I/O setup
comments, 4–10
complementary output signals and polarity, 4–10
configuring I/O, 4–6
hardware, 4–3
kinds of I/O, 4–3
overview, 4–3
ignore offset setup, 3–129
ignore tool offset setup, 3–130
init start, C–15

Input, 1–18
input skip, 10–440
input/output instructions, *see* I/O instructions
interference check
TP instructions, 16–58
interference checking
collision avoid, 16–3
collision checking, 16–36
combination pairs, 16–37
overview, 16–36
DualARM single controller, 16–6
DualARM single controller interference checking, 16–57
modeling, 16–7
obstacle modeling, 16–26
overview, 16–7
robot arm modeling, 16–13
teaching a box obstacle, 16–35
teaching a plane obstacle, 16–32
user tool modeling, 16–18
multiple controllers, 16–5
multiple robot applications, 16–39
auto setup of default interference zone between robot pairs, 16–47
combination check pairs, 16–43
host robot, 16–39
overview, 16–39
priority list, 16–39
overview, 16–3
robot arm, 16–36
robot parameters, 16–61
static obstacles, 16–36
status and recovery, 16–59
overview, 16–59
recovery, 16–60
iPendant, 1–27
features, 1–27
guidelines, 1–41
editing, 1–41
operational, 1–60
help and diagnostics, 1–48
HMI screen customizations, F–25
multiple window display, 1–39
screen navigation, 1–39
status window, 1–43
windows, 1–39
focus, 1–41
popup menus, 1–43
status bar displays, 1–45
iPendant Firmware, C–31
*i*RCalibration Frame Set, 15–139
create program, 15–156
displaying UFrame log, 15–158
mastering a frame
creating a frame calibration TP program, 15–150
mastering a frame search TP program, 15–156
preliminary setup, 15–140
recovering a frame, 15–156
run program, 15–156
running the program, 15–156
setting up
uframe calibration schedule, 15–141
setup, 15–140
teaching points, 15–151
troubleshooting, 15–160
iRCalibration Frame Shift, 15–139, 15–163
create program, 15–188
displaying UFrame log, 15–191
mastering a frame
creating a frame calibration teach pendant program, 15–178
mastering a frame search teach pendant program, 15–188
overview, 15–139, 15–163
preliminary setup, 15–164
recovering a frame, 15–188
run program, 15–188
running the program, 15–188
setting up
uframe calibration schedule, 15–168
setup, 15–167
teaching points, 15–179
troubleshooting, 15–193
iRCalibration Master Set
calibration log, 15–38, 15–74
overview, 15–8
setting up, 15–18
troubleshooting, 15–40
updating master count, 15–37
iRCalibration Mastering

overview, 15–42
setting up, 15–53
troubleshooting, 15–76
updating master count, 15–73
ircalibration options, 15–3
overview
electrical continuity sensing, 15–5
torque observer sensor, 15–5
*i*RCalibration TCP Set
displaying UTool log, 15–105
error recovery, 15–107
fit errors, 15–107
redoing adjustment, 15–107
overview, 15–78
preliminary setup, 15–78
setting up
touch plate installation, 15–80
setup, 15–78, 15–83
touch plate, 15–80
troubleshooting, 15–108
*i*RCalibration TCP Shift
displaying UTool log, 15–135
error recovery, 15–137
fit errors, 15–137
redoing adjustment, 15–137
overview, 15–109
preliminary setup, 15–109
setting up
touch plate installation, 15–111
setup, 15–109, 15–113
touch plate, 15–111
troubleshooting, 15–138

J

J502
order number, 12–2
J523
order number, 10–129
j527, 3–179
J527
order number, 3–179
J535
order number, B–2
J536
order number, 18–3
J541
order number, 3–137
J553
order number, 10–183
J609
order number, 10–470
j611, 10–518
J611
order number, 10–518
J612
order number, 10–446
J624
order number, 3–5
J627
order number, 6–162
J644
order number, 10–424
J649
order number, 10–242
J664
order number, 10–209
J666
order number, 10–98
J669
order number, 3–188
J671
order number, 11–3
J675
order number, 10–139
J695
order number, 10–330
J831
order number, 10–248
J878
order number, 10–348
J887
order number, 4–89
J951
order number, 10–56
J957 option, 9–5
jog frame, 3–61
direct entry method, 3–69
selecting, 3–73
setting up, 3–61
three point method, 3–64
jog menu, 2–18
extended user frames, 10–240
jog speed, 2–6

robot
jogging, 2–6
jogging, 2–2
extended axes, 2–13
motion groups, 2–12
robot, 2–5
coordinate systems, 2–7
jog speed, 2–6
overview, 2–5
remote TCP jogging, 2–11
wrist jogging, 2–10
sub groups, 2–13
jogginh
jog menu, 2–18
joint motion, 1–20

K

KAREL paths, 3–215
KAREL positions, 3–212
KAREL variables, 3–224
teaching, 3–212
KCL
using with passwords enabled, 3–171
keys
editing, 1–35
execution, 1–34
function
favorites screen, 1–61
Handlingtool-specific, 1–36
HandlingTool-specific, 1–36
help and diagnostic (iPendant only), 1–37
navigation and data entry, 1–31
robot motion, 1–33
teach pendant, 1–31

L

Laser
Analog Input Non-Contact Touch Sensing, 18–54
limit set, 10–249
line number, 6–22
line remark instruction, 6–154
Line tracking, 1–25
linear motion, 1–20
log
alarm, A–3
diagnostic, C–42
reporting, A–7
Log Book, Operation, 10–330
log file data, 10–162
log viewer, 10–163

M

machine tool easy setup, 14–2
overview, 14–2
See also machine tool options
machine tool options, 14–2
macro command instruction, 6–168
macro command programs
assigning I/O and recording positions, 3–117
macro commands, 3–99, 3–118
executing, 3–113
executing from MANUAL FCTNS menu, 3–114
executing from teach pendant user keys, 3–114
overview, 3–99
setting up, 3–99
macro program, 5–47
maintenance and repair, 7–54
maintenance operations, C–28
updating boot software, C–28
updating iPendant firmware, C–31
manipulating files, 9–42
backing up files, 9–52
checking file memory, 9–74
copying files, 9–67
deleting files, 9–70
displaying text (ASCII) files, 9–66
generating a directory of files, 9–50
loading files to controller memory, 9–58
overview, 9–42
restoring files to controller memory, 9–58
saving files, 9–71
manipulating programs, 9–30
copying programs, 9–36
deleting programs, 9–37
loading programs, 9–34
overview, 9–30
printing, 9–39
saving programs, 9–32

SELECT menu, 9–31, 9–37
selecting programs, 9–31
MANUAL FCTNS screen, 7–54
marker, 6–255
mastering, D–2
resetting alarms and preparing for, D–4
saving and restoring data, D–6
single axis, D–11
to a fixture, D–8
material handling
setting up
controlling valve I/O, 13–19
valve I/O, 13–17
valve signals, 13–12
valve macro programs, 13–23
examples, 13–24
release part, 13–27
material handling tool
production, 13–41
alarms, 13–42
disable for 20 cycles, 13–41
MH tryout mode, 13–41
setting up
tool valves, 13–6
valve macro programs
set MH Tool, 13–40
material handling tooling
controlling tooling valve in manual mode, 13–22
overview, 13–2
setting up
selecting default application, 13–3
valve macro programs
check no part, 13–30
clear to proceed, 13–32
grip part, 13–25
part present, 13–29
prepare to pick, 13–31
set currentvalve, 13–39
turn OFF blowoff, 13–37
turn OFF vacuum, 13–36
turn ON vacuum, 13–33
material handling tools
setting up, 13–3
Maximum of ACC Instruction
setting, 3–231
memory, 1–23
memory card (MC), 9–3
memory device, 9–3, 9–7
filtered, 9–3, 9–8
memory device backup, 9–7
memory device binary, 9–3
memory status, 8–31
menu utility, 10–252
list menu, 10–265
operator entry menu, 10–285
overview, 10–252
prompt box message menu, 10–253
prompt box yes/no menu, 10–259
status menu, 10–273
teach pendant menus, 10–294
MF device, 9–3, 9–8
Minimum of ACC Instruction
setting, 3–231
mirror image
enhanced (option), 10–190
mirror image of extended axes, 10–197, 10–300
mirror image utility, 10–295
miscellaneous instructions, 6–151
maximum speed instruction, 6–157
message instruction, 6–155
OVERRIDE instruction, 6–152
overview, 6–151
parameter name instruction, 6–155
RSR enable/disable instruction, 6–151
timer instruction, 6–152
user alarm instruction, 6–151
mixed logic, 6–241
adding instructions, 6–245
background logic, 6–247
backup/restore, 6–260
data types, 6–241
expressions, 6–243
flag, 6–252
marker, 6–255
operators, 6–242
overview, 6–241
TC_Online, 6–257
mode
multitask, 3–235
single task, 3–235
MODE SELECT switch, 1–12
AUTO, 1–14
T1, 1–13

T2, 1–14
model A I/O, 4–46
model B I/O, 4–29, 4–47
modifying a program, 5–28
background editing, 5–52
setting up and using predefined positions, 5–45
modifying a program in the background, *see* background editing
monitoring
diagnostic video, 10–154
monitoring programs, 10–311
overview, 10–311
monochrome teach pendant, 1–27
motion, 1–20
motion group DO output function, 10–313
operations, 10–314
overview, 10–313
restrictions, 10–314
setup, 10–315
subprogram and multi-tasking execution, 10–316
motion group instructions, 6–186
defining motion group instructions, 6–187
independent motion group instructions, 6–186
overview, 6–186
simultaneous motion group instructions, 6–186
motion groups, 1–20, 2–12
jogging, 2–12
motion instruction, 6–22
frame number of positional data, 6–40
KAREL program motion, 6–117
motion options, 6–52
motion type, 6–23
overview, 6–22
positional information, 6–28
speed, 6–42
switch frame check type, 6–41
teach pendant program motion, 6–117
termination type, 6–50
motion option
Constant Path, 6–72
motion start delay detection function, 10–317
detection information and system variables, 10–318
motion start delay detection function example, 10–319
overview, 10–317
restrictions, 10–317
motion status display, 6–30
motion types
circular, 1–20
joint, 1–20
linear, 1–20
Multi Program Selection
setting, 3–235
multi-arm
family members, 6–22
head of family, 6–22
multi-lng remark instruction, 6–153
multi-tasking, 10–232
multiple applications (option), 1–22
multiple control instructions, 6–167
multiple motion group option, *see* motion group DO output function
multiple window display, 1–39

N

navigation
iPendant, 1–39
NOT RECOVER
setting, 3–227

O

offline singularity avoidance, 10–320
compatibility, 10–323
how it works, 10–321
limitations, 10–324
overview, 10–320
procedure, 10–324
troubleshooting, 10–329
offset/frame instructions, 6–163
older programs
conversion, 9–64
ON/OFF switch, 1–29
Operation Log Book, 10–330
operation panel
setup
panel wizard, F–21
operator panel, 1–10
UOP, 1–17

order number, 9–65
Original Program Name
setting, 3–230
OTHER program select mode, 3–98
Output, 1–18
output error parameters, 3–186
output errors
digital outputs, 3–180
group outputs, 3–185
Output when WAIT on input
setting, 3–233
override select setup, 3–134
overview
controller, 1–7
error code output setup, 3–179
frames setup, 3–3
mastering, D–2
robot, 1–4
jogging, 2–5
robot payload setting, 3–188
software, 1–65
system, 1–3

P

pallet register instructions, 6–235
pallet register status, 8–17
palletizing
all-point teaching, 6–232
palletizing instructions, 6–200
overview, 6–200
PALLETIZING-B instruction, 6–203
PALLETIZING-BX instruction, 6–209
PALLETIZING-E instruction, 6–216
PALLETIZING-END instruction, 6–232
PALLETIZING-EX instruction, 6–224
panel wizard
operation panel setup, F–21
parallel and rotational shift, 10–358
parallel mirror image, 10–192, 10–296
parallel shift, 10–357
parameters for program call and macro instructions, 6–169
argument registers, 6–174, 6–181
guidelines for using parameters, 6–175
including parameters, 6–177
macro instructions, 6–177
overview, 6–169
parameter instruction syntax, 6–170
program call, 6–177
string value selections, 6–172
sub-programs, 6–181
password
auto login function, 3–172
install user password operations, 3–138
password level screen permissions, 3–165
password log, 3–162
program and setup users password operation, 3–143
setup, 3–137
USB password, 3–169
using KCL with passwords enabled, 3–171
password level screen permissions, 3–165
password log, 3–162
PAUSE instruction, 6–183
payload confirm, 10–348
payload instruction, 6–194
payload schedule, 3–188
payload schedules
setting up, 3–190
PC file
conversion, 9–64
planning a program
macros, 5–7
motion, 5–3
overview, 5–3
position registers, 5–8
predefined positions, 5–7
programs, 5–8
planning and creating a program
overview, 5–3
PNS production start, 7–42
popup menus, 1–43
position confirmation, 6–29
position register instructions, 6–123
overview, 6–123
PR[i,j] position register instructions, 6–125
PR[x] position register instructions, 6–123
position register look-ahead execution
function, 10–353
execution, 10–356
overview, 10–353
program example, 10–355
program instructions, 10–354

position register look-ahead instructions, 6–189
position register status, 8–10
position status, 8–33
predefined positions, 5–45
process I/O, 4–45
process speed, 6–96
Prod_Start depend on PNSTROBE
setting, 3–229
ProductDisplayImprove, 4–89
production operation, 7–35, 7–38
overview, 7–35
program number select (PNS) and UOP
production start, 7–42
robot service request (RSR) production
start, 7–41
user operator panel start, 7–39
production operation setup, 3–76
production operation setup overview, 3–76
production status, 8–43
PROG ADJUST, *see* adjusting a program
Prog Select, 3–76
program, 1–66
conversion of older programs, 9–64
planning and creating, 5–3
running, 1–67
testing, 1–66
program control instructions, 6–182
ABORT instruction, 6–183
clear resume program instruction, 6–185
error program instruction, 6–183
maintenance program instruction, 6–184
PAUSE instruction, 6–183
resume program instruction, 6–184
return path disable instruction, 6–185
program elements
overview, 6–5
program end marker, 6–22
program examples, E–2
program header information, 6–7
application mask, 6–20
copy source, 6–13
creation date, 6–12
group mask, 6–18
ignore pause, 6–20
modification date, 6–13
overview, 6–7
positions and program size, 6–13
program comment, 6–18
program name, 6–13
sub type, 6–14
write protection, 6–19
program monitor status, 8–61
Program Number Select (PNS) setup, 3–86
program pause and recovery, 7–3
EMERGENCY STOP and recovery, 7–3
HOLD and recovery, 7–5
overview, 7–3
setting tolerance for resuming a
program, 7–6
program shift utility, 10–356
extended axes, 10–359
parallel and rotational shift, 10–358
parallel shift, 10–357
program status, 8–43
program timer status, 8–22
PSPD, 6–96

Q

quick skip, 6–162

R

R505
order number, 10–7
R507
order number, 9–89
R510
order number, 10–40
R512
order number, 10–24
R516
order number, 10–59
R518
order number, 15–163
R520
order number, 10–82
R531
order number, 1–25
R534
order number, 10–91
R542
order number, A–3
R545
order number, 9–8

R548
order number, 10–248
R553
order number, F–25
R557
order number, 16–3
R577
order number, 10–252
R578
order number, 13–2
R585
order number, 10–320
R594
order number, F–21
R596
order number, 10–249
R601
order number, 15–8, 15–42
R609
order number, 10–429
R617
order number, 15–78, 15–109
R620
order number, 10–501
R633
order number, 6–84
R634
order number, 6–100
R640
order number, 6–56
R641
order number, 10–378
R657
order number, 6–68
R658
order number, 6–84
R663
order number, 6–72
R674
order number, 10–393
R675
order number, 10–387
R692
order number, 10–154
R696
order number, 10–239
R698
order number, 10–190
R727
order number, 14–2
RAM Disk (RD), 9–3, 9–6
Receive Timeout
setting, 3–230
RECOVER ALL
setting, 3–227
RECOVER SIM
setting, 3–227
reference position utility, 10–368
register instructions, 6–118
register status, 8–8
release wait, 7–34
remark instruction, 6–153
remote I/O interfaces, 1–19
remote TCP, 6–68
remote TCP frame, 3–50
setting up, 3–50
remote TCP jogging, 2–11
robot, 2–11
Remote/Local setup
setting, 3–234
repair, 7–54
reporting log, A–7
Reset Chain Failure
setting, 3–231
restore, C–22
Restore Selected Program
setting, 3–228
restoring controller images, C–25
resuming a program, 7–6
Return to Top of Program
setting, 3–230
robot
axes, 1–4
jogging, 2–5
coordinate systems, 2–7
jog menu, 2–18
jog speed, 2–6
overview, 2–2, 2–5, 2–11
wrist jogging, 2–10
models, 1–5
overview, 1–4
turning off, 2–2
turning on, 2–2
overview, 2–2

robot axes, 1–4
robot axes status, 8–45
displaying the axis status screens, 8–60
disturbance torque screen, 8–53
overview, 8–45
pulse screen, 8–50
servo diagnosis, 8–53
status 1 screen, 8–45
status 2 screen, 8–49
torque monitor screen, 8–51
tracking screen, 8–52
robot I/O
setup, 4–37
robot payload setting
arm load information, 3–209
overview, 3–188
payload schedule, 3–188
payload schedules
setting up, 3–190
robot ready, 4–89
Robot Service Request (RSR) Setup, 3–82
robot stop variation, 1–16
robot-PC communications macros, 10–378
get data(, 10–383
overview, 10–378
request menu( ), 10–384
send data ( ), 10–380
send event ( ), 10–378
send sysvar(, 10–381
RobotCal, 15–8, 15–42
rotational mirror image, 10–194, 10–299
RS-232-C, 1–18
RS-422, 1–18
RSR production start, 7–41
running
program, 1–67
running production
overview, 7–2

S

safety signal status, 8–20
safety space, 10–385
operation, 10–386
overview, 10–385
requirements, 10–385
saving ASCII, 9–74
saving binary, 9–74
saving frame data, 3–74
schedules
setting up, 10–155
screen navigation
iPendant, 1–39
SEARCH END instruction, 6–239
SEARCH START instruction, 6–238
SELECT menu, 9–36
selecting a jog frame, 3–73
selecting a tool frame, 3–25
selecting a user frame, 3–49
servo tool, 10–387
servo tool change, 11–3
Set if Input Simulated
setting, 3–232
Set if sim. Skip Enabled
setting, 3–232
Set When Prompt Displayed
setting, 3–232
setting tolerance, 7–6
setting up DVM schedules, 10–155
setting up I/O
overview, 4–5
setting up macro commands, 3–99
setting up predefined positions, 5–45
setup, 3–3
axis limits, 3–119
brake on hold, 3–127
brake timers, 3–122
current language, 3–129
frames, 3–3
ignore offset, 3–129
ignore tool offset, 3–130
jog frame, 3–61
override select, 3–134
overview
production operation, 3–76
production operation, 3–76
remote TCP frame, 3–50
robot payload schedules, 3–190
software, 1–66
tool frame, 3–6
user alarm, 3–131
user frame, 3–27
shape generation II, 10–393
custom shape data, 10–413

overview, 10–393
program generation, 10–416
programming, 10–416
shape data setup, 10–397
shape setup, 10–395
standard shape data, 10–400
tuning guidelines, 10–426
UTILITIES, 10–416
Signal if Override=100
setting, 3–233
Signal to set if E-Stop
setting, 3–232
Signal to Set in Auto Mode
setting, 3–232
Signal to set in T1 Mode
setting, 3–232
Signal to Set in T2 Mode
setting, 3–232
Sim. Input Wait Delay
setting, 3–232
simple teach suite, 10–429
auto path smoothing, 10–430
how it works, 10–430
limitations, 10–430
overview, 10–430
overview, 10–429
simulated input skip, 10–440
single task mode, 3–235
singularity check function, 10–445
six point method
tool frame setup, 3–16
skip, 6–162
See also quick skip
simulated input, 10–440
skip instruction, 6–159
small circle accuracy, 10–424
Soft Float, 10–446
Cartesian, 10–451
joint, 10–449
pushout, 10–453
restrictions, 10–447
schedules, 10–449, 10–451, 10–453
Soft Float function
overview, 10–446
program instructions, 10–456
true position variables, 10–468
software
overview, 1–65
program, 1–66
running a program, 1–67
setup, 1–66
testing a program, 1–66
SOP cycle start production, 7–38
space check function, 10–470
SRAM, 1–23
standard operator panel, 1–10
standard operator panel (SOP) cycle start production, 7–38
standard operator panel (SOP) I/O
setup, 4–41
START for CONTINUE only
setting, 3–228
startup methods, C–2
cold start, C–6
controlled start, C–10
hot start, C–3
init start, C–15
status
production, 8–43
program, 8–43
status window, 1–43
status bar displays, 1–45
status indicators, 8–3
overview, 8–3
standard operator panel (SOP) status indicators, 8–5
teach pendant status indicators, 8–3
storage devices, 9–3
formatting devices, 9–23
overview, 9–3
setting the default device, 9–10
setting up a port, 9–12
using a memory card interface, 9–19
using a USB memory stick, 9–19
string register instructions, 6–128
overview, 6–128
string register status, 8–14
stroke limit setup, 3–210
style name setup, 3–92
sub groups, 2–13
system
overview, 1–3
system operations

startup methods, C–2
hot start, C–3
overview, C–2
system timer status, 8–25
system variable status, 8–18

T

T1, 1–13
T2, 1–14
TC_Online, 6–257
TCP prediction, 1–20
TCP speed prediction, 10–487
overview, 10–487
setting up and running TCP speed prediction, 10–488
system variables, 10–489
teach pendant, 1–10, 1–26
EMERGENCY STOP button, 1–30
iPendant, 1–27
screen navigation, 1–39
status window, 1–43
keys, 1–28, 1–31
kinds, 1–26
monochrome, 1–27
motion control buttons, 1–28
ON/OFF switch, 1–29
teach pendant keys, 1–31
teach pendant shim, 10–501
history, 10–507
overview, 10–501
previous shims, 10–507
setting up, 10–504
using, 10–504
teaching KAREL variables, 3–212
termination speed, 1–20
termination type, 1–20
test cycle, 7–16
continuous testing, 7–28
monitoring programs, 7–34
single step testing, 7–21
test cycle setup, 7–16
test cycle overview, 7–16
Test Mode 1, 1–13
Test Mode 2, 1–14
testing
program, 1–66
testing a program
overview, 7–2
test cycle, 7–16
test cycle overview, 7–16
three point method
jog frame, 3–64
tool frame setup, 3–11
user frame setup, 3–33
time before/after motion option
instruction, 10–510
execution timing, 10–513
overview, 10–510
program example, 10–516
program execution, 10–510
programming hints, 10–517
recording, 10–514
tool change
servo, 11–3
tool frame, 3–6
direct entry method, 3–22
selecting, 3–25
setup, 3–6
six point method, 3–16
three point method, 3–11
tool offset condition instruction, 6–166
torque limit, 10–518
TOUCH OFFSET instruction, 6–239
touch sense instructions, 6–237
overview, 6–237
SEARCH END instruction, 6–239
SEARCH START instruction, 6–238
TOUCH OFFSET instruction, 6–239
touch sensing
adding new positions, 18–46
assigning I/O, 18–4
overview, 18–4
assigning inputs and outputs, 18–6
enable/disable output signal, 18–5, 18–51
hardware, 18–51
overview, 18–51
I/O, 18–4
overview, 18–4
input signal, 18–5, 18–51

mastering, 18–38
adding new positions, 18–46
master flag, 18–42
offsets, 18–40
overview, 18–38
patterns, 18–41
touching up path positions, 18–44
touching up search start positions, 18–49
multiple searches, 18–47
overview, 18–3
programming, 18–28
executing a program, 18–30
instructions, 18–28
motion instructions use with touch sensing, 18–30
motion option, 18–29
programming examples, 18–32
robot position touchup, 18–31
programming overview, 18–28
remastering, 18–39
search start positions, 18–49
setting up, 18–7
overview, 18–7
search pattern, 18–16
touch frames, 18–10
touch schedule, 18–22
simple low voltage circuit, 18–52
touching up path positions, 18–44
truncating
common characters, 6–14, 9–31
turn number display, 8–36
overview, 8–36
$SCR_GRP[group].$turn_axis[i] system variable, 8–39
usual configuration, 8–36

U

ultitask mode, 3–235
UNSIMULATE
setting, 3–227
UOP, 1–17
input signals, 4–75
output signals, 4–80
signal definition, 4–72
UOP Auto Assignment
setting, 3–235
UOP production start, 7–42
UOP signal definition, 4–72
update
automatic software, 9–65
updating iPendant firmware, C–31
USB, 1–65
USB Memory Stick Device (UD1), 9–3 to 9–4
USB Memory Stick Device (UT1), 9–3
USB password, 3–169
Use HOT START
setting, 3–226
Use PPABN Signal
setting, 3–230
user alarm setup, 3–131
user frame, 3–27
direct entry method, 3–45
extended, 10–239
four point method, 3–39
selecting, 3–49
setting up, 3–27
three point method, 3–33
user frames
extended
jog menu, 10–240
user I/O
distributed (model B) I/O, 4–36
user operator panel, 1–17
user operator panel definition, 4–72
user screen status, 8–7
using predefined positions, 5–45

V

version identification status, 8–27
video loop
displaying, 10–162
viewing the log, 10–163

W

wait instructions, 6–146
Wait Timeout
setting, 3–230
window display guidelines, 1–60
windows
changing focus, 1–41
display guidelines, 1–60

focus, 1–41
help and diagnostics information display locations, 1–49
iPendant screen, 1–39
popup menus, 1–43
status bar displays, 1–45
status/single, 1–43
WJNT for Default Motion
setting, 3–231
wrist jogging, 2–10
robot
jogging, 2–10
writing and modifying a program
overview, 5–15
writing a new program, 5–17