

FANUC Robot series

R-30*i*A Mate CONTROLLER

LR HANDLING TOOL

(For series 7DA4/7DA5/7DA7)

OPERATOR'S MANUAL

MAROTHT7708111E REV. A

This 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, Michigan 48309-3253

B-83134EN-1/01

This manual can be used with controllers labeled R-30iA or R-J3iC. If you have a controller labeled R-J3iC, you should read R-30iA as R-J3iC throughout this 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.

The descriptions and specifications contained in this manual were in effect at the time this manual was approved for printing. 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 radiofrequency 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 the company's world headquarters in Rochester Hills, Michigan. For additional information contact

FANUC Robotics America Corporation
Training Department
3900 W. Hamlin Road
Rochester Hills, Michigan 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

**Copyright ©2011 by FANUC Robotics America Corporation
All Rights Reserved**

The information illustrated or contained herein is not to be reproduced, copied, downloaded, translated into another language, 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 Corporation.

AccuStat®, ArcTool®, iRVision®, 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-J3iB Controller™, SYSTEM R-J3iC Controller™, SYSTEM R-30iA Controller™, TCP Mate™, TorchMate™, TripleARM™, TurboMove™, visLOC™, visPRO-3D™, visTRAC™, WebServer™, WebTP™, and YagTool™.

©FANUC CORPORATION 2011

- No part of this manual may be reproduced in any form.
- All specifications and designs are subject to change without notice.

Patents

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

FANUC Robotics America Corporation 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 CORPORATION 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

Conventions

WARNING

Information appearing under the "WARNING" caption concerns the protection of personnel. It is boxed and bolded to set it apart from the surrounding text.

CAUTION

Information appearing under the "CAUTION" caption concerns the protection of equipment, software, and data. It is boxed and bolded to set it apart from the surrounding text.

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

- **Original Instructions**

Before using the Robot, be sure to read the "FANUC Robot Safety Manual (B-80687EN)" and understand the content.

- No part of this manual may be reproduced in any form.
- All specifications and designs are subject to change without notice.

The products in this manual are controlled based on Japan's "Foreign Exchange and Foreign Trade Law". The export from Japan may be subject to an export license by the government of Japan.

Further, re-export to another country may be subject to the license of the government of the country from where the product is re-exported. Furthermore, the product may also be controlled by re-export regulations of the United States government.

Should you wish to export or re-export these products, please contact FANUC for advice.

In this manual we have tried as much as possible to describe all the various matters.

However, we cannot describe all the matters which must not be done, or which cannot be done, because there are so many possibilities.

Therefore, matters which are not especially described as possible in this manual should be regarded as "impossible".

SAFETY PRECAUTIONS

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 be best 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 Robotics 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 its 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 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.

ADDITIONAL SAFETY CONSIDERATIONS FOR PAINT ROBOT INSTALLATIONS

Process technicians are sometimes required to enter the paint booth, for example, during daily or routine calibration or while teaching new paths to a robot. Maintenance personnel also must work inside the paint booth periodically.

Whenever personnel are working inside the paint booth, ventilation equipment must be used. Instruction on the proper use of ventilating equipment usually is provided by the paint shop supervisor.

Although paint booth hazards have been minimized, potential dangers still exist. Therefore, today's highly automated paint booth requires that process and maintenance personnel have full awareness of the system and its capabilities. They must understand the interaction that occurs between the vehicle moving along the conveyor and the robot(s), hood/deck and door opening devices, and high-voltage electrostatic tools.

**CAUTION**

Ensure that all ground cables remain connected. Never operate the paint robot with ground provisions disconnected. Otherwise, you could injure personnel or damage equipment.

Paint robots are operated in three modes:

- Teach or manual mode
- Automatic mode, including automatic and exercise operation
- Diagnostic mode

During both teach and automatic modes, the robots in the paint booth will follow a predetermined pattern of movements. In teach mode, the process technician teaches (programs) paint paths using the teach pendant.

In automatic mode, robot operation is initiated at the System Operator Console (SOC) or Manual Control Panel (MCP), if available, and can be monitored from outside the paint booth. All personnel must remain outside of the booth or in a designated safe area within the booth whenever automatic mode is initiated at the SOC or MCP.

In automatic mode, the robots will execute the path movements they were taught during teach mode, but generally at production speeds.

When process and maintenance personnel run diagnostic routines that require them to remain in the paint booth, they must stay in a designated safe area.

Paint System Safety Features

Process technicians and maintenance personnel must become totally familiar with the equipment and its capabilities. To minimize the risk of injury when working near robots and related equipment, personnel must comply strictly with the procedures in the manuals.

This section provides information about the safety features that are included in the paint system and also explains the way the robot interacts with other equipment in the system.

The paint system includes the following safety features:

- Most paint booths have red warning beacons that illuminate when the robots are armed and ready to paint. Your booth might have other kinds of indicators. Learn what these are.

- Some paint booths have a blue beacon that, when illuminated, indicates that the electrostatic devices are enabled. Your booth might have other kinds of indicators. Learn what these are.
- EMERGENCY STOP buttons are located on the robot controller and teach pendant. Become familiar with the locations of all E–STOP buttons.
- An intrinsically safe teach pendant is used when teaching in hazardous paint atmospheres.
- A DEADMAN switch is located on each teach pendant. When this switch is held in, and the teach pendant is on, power is applied to the robot servo system. If the engaged DEADMAN switch is released or pressed harder during robot operation, power is removed from the servo system, all axis brakes are applied, and the robot comes to an EMERGENCY STOP. Safety interlocks within the system might also E–STOP other robots.



An EMERGENCY STOP will occur if the DEADMAN switch is released on a bypassed robot.

- Overtravel by robot axes is prevented by software limits. All of the major and minor axes are governed by software limits. Limit switches and hardstops also limit travel by the major axes.
- EMERGENCY STOP limit switches and photoelectric eyes might be part of your system. Limit switches, located on the entrance/exit doors of each booth, will EMERGENCY STOP all equipment in the booth if a door is opened while the system is operating in automatic or manual mode. For some systems, signals to these switches are inactive when the switch on the SOC is in teach mode.
- When present, photoelectric eyes are sometimes used to monitor unauthorized intrusion through the entrance/exit silhouette openings.
- System status is monitored by computer. Severe conditions result in automatic system shutdown.

Staying Safe While Operating the Paint Robot

When you work in or near the paint booth, observe the following rules, in addition to all rules for safe operation that apply to all robot systems.



Observe all safety rules and guidelines to avoid injury.

 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.

 WARNING

Enclosures shall not be opened unless the area is known to be nonhazardous or all power has been removed from devices within the enclosure. Power shall not be restored after the enclosure has been opened until all combustible dusts have been removed from the interior of the enclosure and the enclosure purged. Refer to the Purge chapter for the required purge time.

- Know the work area of the entire paint station (workcell).
- Know the work envelope of the robot and hood/deck and door opening devices.
- Be aware of overlapping work envelopes of adjacent robots.
- Know where all red, mushroom-shaped EMERGENCY STOP buttons are located.
- Know the location and status of all switches, sensors, and/or control signals that might cause the robot, conveyor, and opening devices to move.
- Make sure that the work area near the robot is clean and free of water, oil, and debris. Report unsafe conditions to your supervisor.
- Become familiar with the complete task the robot will perform BEFORE starting automatic mode.
- Make sure all personnel are outside the paint booth before you turn on power to the robot servo system.
- Never enter the work envelope or paint booth before you turn off power to the robot servo system.
- Never enter the work envelope during automatic operation unless a safe area has been designated.
- Never wear watches, rings, neckties, scarves, or loose clothing that could get caught in moving machinery.
- Remove all metallic objects, such as rings, watches, and belts, before entering a booth when the electrostatic devices are enabled.
- Stay out of areas where you might get trapped between a moving robot, conveyor, or opening device and another object.
- Be aware of signals and/or operations that could result in the triggering of guns or bells.
- Be aware of all safety precautions when dispensing of paint is required.
- Follow the procedures described in this manual.

Special Precautions for Combustible Dusts (Powder Paint)

When the robot is used in a location where combustible dusts are found, such as the application of powder paint, the following special precautions are required to insure that there are no combustible dusts inside the robot.

- Purge maintenance air should be maintained at all times, even when the robot power is off. This will insure that dust can not enter the robot.
- A purge cycle will not remove accumulated dusts. Therefore, if the robot is exposed to dust when maintenance air is not present, it will be necessary to remove the covers and clean out any accumulated dust. Do not energize the robot until you have performed the following steps.
 1. Before covers are removed, the exterior of the robot should be cleaned to remove accumulated dust.
 2. When cleaning and removing accumulated dust, either on the outside or inside of the robot, be sure to use methods appropriate for the type of dust that exists. Usually lint free rags dampened with water are acceptable. Do not use a vacuum cleaner to remove dust as it can generate static electricity and cause an explosion unless special precautions are taken.
 3. Thoroughly clean the interior of the robot with a lint free rag to remove any accumulated dust.
 4. When the dust has been removed, the covers must be replaced immediately.
 5. Immediately after the covers are replaced, run a complete purge cycle. The robot can now be energized.

Staying Safe While Operating Paint Application Equipment

When you work with paint application equipment, observe the following rules, in addition to all rules for safe operation that apply to all robot systems.



WARNING

When working with electrostatic paint equipment, follow all national and local codes as well as all safety guidelines within your organization. Also reference the following standards: NFPA 33 Standards for Spray Application Using Flammable or Combustible Materials, and NFPA 70 National Electrical Code.

- **Grounding:** All electrically conductive objects in the spray area must be grounded. This includes the spray booth, robots, conveyors, workstations, part carriers, hooks, paint pressure pots, as well as solvent containers. Grounding is defined as the object or objects shall be electrically connected to ground with a resistance of not more than 1 megohms.
- **High Voltage:** High voltage should only be on during actual spray operations. Voltage should be off when the painting process is completed. Never leave high voltage on during a cap cleaning process.
- Avoid any accumulation of combustible vapors or coating matter.
- Follow all manufacturer recommended cleaning procedures.
- Make sure all interlocks are operational.

- No smoking.
- Post all warning signs regarding the electrostatic equipment and operation of electrostatic equipment according to NFPA 33 Standard for Spray Application Using Flammable or Combustible Material.
- Disable all air and paint pressure to bell.
- Verify that the lines are not under pressure.

Staying Safe During Maintenance

When you perform maintenance on the painter system, observe the following rules, and all other maintenance safety rules that apply to all robot installations. Only qualified, trained service or maintenance personnel should perform repair work on a robot.

- Paint robots operate in a potentially explosive environment. Use caution when working with electric tools.
- When a maintenance technician is repairing or adjusting a robot, the work area is under the control of that technician. All personnel not participating in the maintenance must stay out of the area.
- For some maintenance procedures, station a second person at the control panel within reach of the EMERGENCY STOP button. This person must understand the robot and associated potential hazards.
- Be sure all covers and inspection plates are in good repair and in place.
- Always return the robot to the “home” position before you disarm it.
- Never use machine power to aid in removing any component from the robot.
- During robot operations, be aware of the robot’s movements. Excess vibration, unusual sounds, and so forth, can alert you to potential problems.
- Whenever possible, turn off the main electrical disconnect before you clean the robot.
- When using vinyl resin observe the following:
 - Wear eye protection and protective gloves during application and removal.
 - Adequate ventilation is required. Overexposure could cause drowsiness or skin and eye irritation.
 - If there is contact with the skin, wash with water.
 - Follow the Original Equipment Manufacturer’s Material Safety Data Sheets.
- When using paint remover observe the following:
 - Eye protection, protective rubber gloves, boots, and apron are required during booth cleaning.
 - Adequate ventilation is required. Overexposure could cause drowsiness.
 - If there is contact with the skin or eyes, rinse with water for at least 15 minutes. Then seek medical attention as soon as possible.
 - Follow the Original Equipment Manufacturer’s Material Safety Data Sheets.

SAFETY PRECAUTIONS

For the safety of the operator and the system, follow all safety precautions when operating a robot and its peripheral devices installed in a work cell.

In addition, refer to the “FANUC Robot SAFETY HANDBOOK (B-80687EN)”.

1 WORKING PERSON

The personnel can be classified as follows.

Operator:

- Turns robot controller power ON/OFF
- Starts robot program from operator's panel

Programmer or teaching operator:

- Operates the robot
- Teaches robot inside the safety fence

Maintenance engineer:

- Operates the robot
- Teaches robot inside the safety fence
- Maintenance (adjustment, replacement)

- An operator cannot work inside the safety fence.
- A programmer, teaching operator, and maintenance engineer can work inside the safety fence. The working activities inside the safety fence include lifting, setting, teaching, adjusting, maintenance, etc..
- To work inside the fence, the person must be trained on proper robot operation.

During the operation, programming, and maintenance of your robotic system, the programmer, teaching operator, and maintenance engineer should take additional care of their safety by using the following safety precautions.

- Use adequate clothing or uniforms during system operation
- Wear safety shoes
- Use helmet

2 WORKING PERSON SAFETY

Working person safety is the primary safety consideration. Because it is very dangerous to enter the operating space of the robot during automatic operation, adequate safety precautions must be observed. The following lists the general safety precautions. Careful consideration must be made to ensure working person safety.

- (1) Have the robot system working persons attend the training courses held by FANUC.

FANUC provides various training courses. Contact our sales office for details.

- (2) Even when the robot is stationary, it is possible that the robot is still in a ready to move state, and is waiting for a signal. In this state, the robot is regarded as still in motion. To ensure working person safety, provide the system with an alarm to indicate visually or aurally that the robot is in motion.
- (3) Install a safety fence with a gate so that no working person can enter the work area without passing through the gate. Install an interlocking device, a safety plug, and so forth in the safety gate so that the robot is stopped as the safety gate is opened.

The controller is designed to receive this interlocking signal of the door switch. When the gate is opened and this signal received, the controller stops the robot (Please refer to "STOP TYPE OF ROBOT" in SAFETY for detail of stop type). For connection, see Fig.2 (a) and Fig.2 (b).

- (4) Provide the peripheral devices with appropriate grounding (Class A, Class B, Class C, and Class D).
- (5) Try to install the peripheral devices outside the work area.
- (6) Draw an outline on the floor, clearly indicating the range of the robot motion, including the tools such as a hand.
- (7) Install a mat switch or photoelectric switch on the floor with an interlock to a visual or aural alarm that stops the robot when a working person enters the work area.
- (8) If necessary, install a safety lock so that no one except the working person in charge can turn on the power of the robot.

The circuit breaker installed in the controller is designed to disable anyone from turning it on when it is locked with a padlock.

- (9) When adjusting each peripheral device independently, be sure to turn off the power of the robot

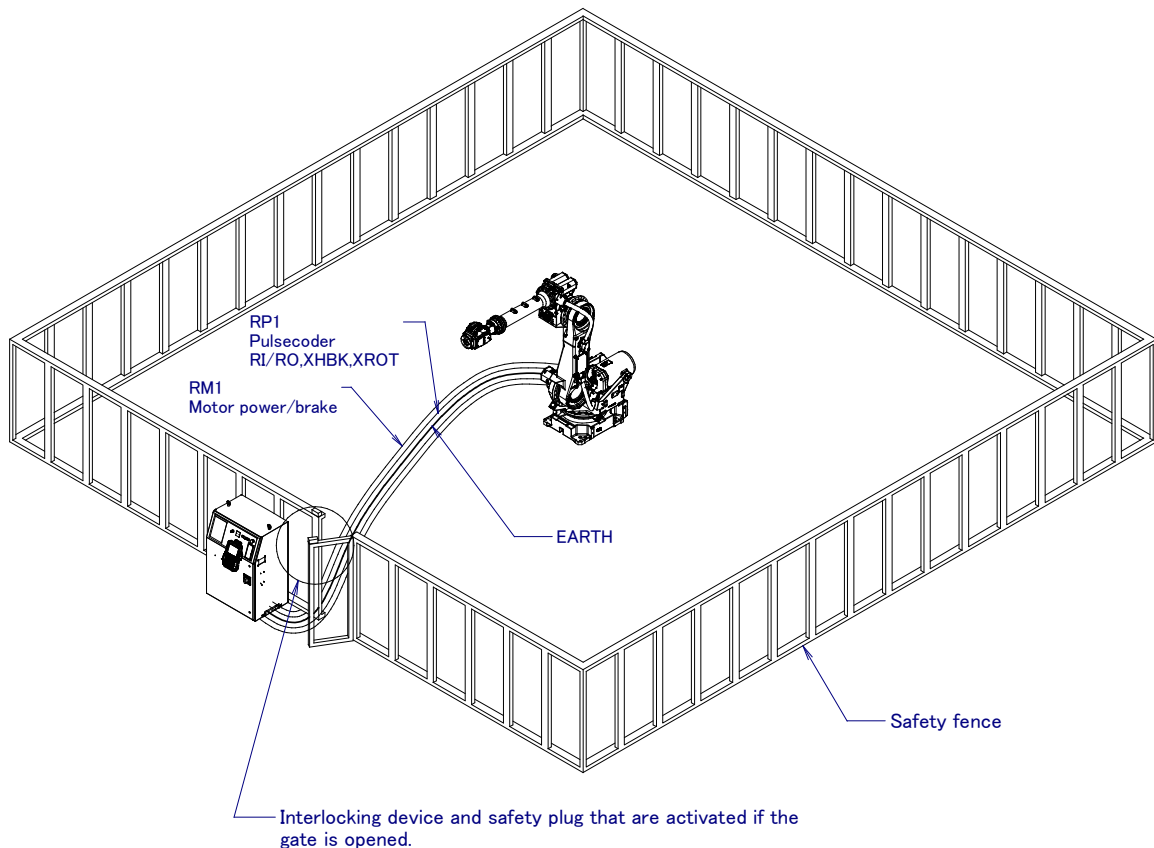


Fig. 2 (a) Safety Fence and Safety gate

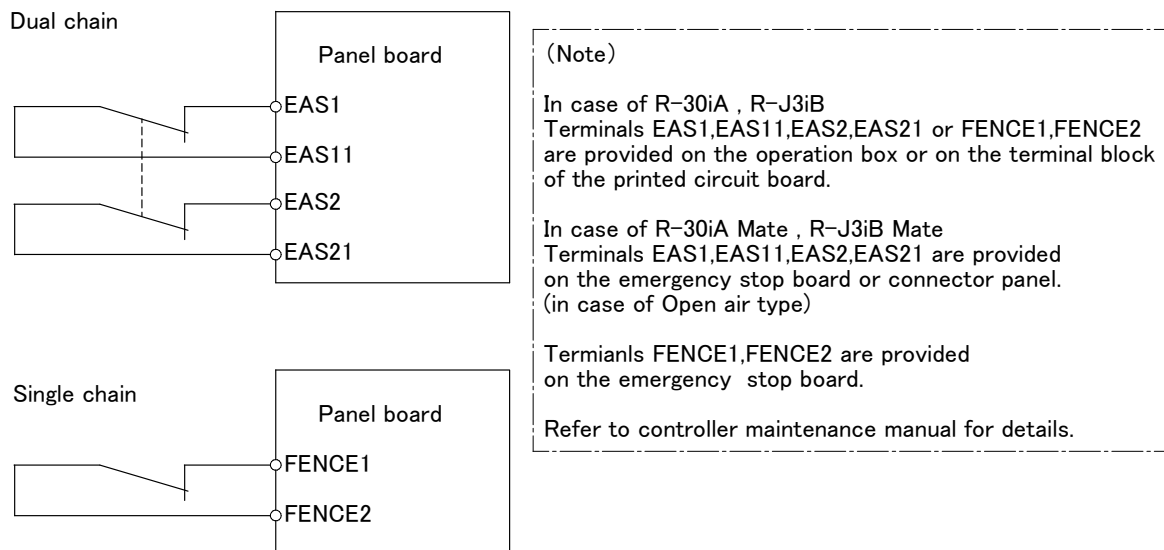


Fig. 2 (b) Limit switch circuit diagram of the safety fence (For R-J3iB CONTROLLER)

2.1 OPERATOR SAFETY

The operator is a person who operates the robot system. In this sense, a worker who operates the teach pendant is also an operator. However, this section does not apply to teach pendant operators.

- (1) If you do not have to operate the robot, turn off the power of the robot controller or press the EMERGENCY STOP button, and then proceed with necessary work.
- (2) Operate the robot system at a location outside of the safety fence
- (3) Install a safety fence with a safety gate to prevent any worker other than the operator from entering the work area unexpectedly and to prevent the worker from entering a dangerous area.
- (4) Install an EMERGENCY STOP button within the operator's reach.

The robot controller is designed to be connected to an external EMERGENCY STOP button. With this connection, the controller stops the robot operation (Please refer to "STOP TYPE OF ROBOT" in SAFETY for detail of stop type), when the external EMERGENCY STOP button is pressed. See the diagram below for connection.

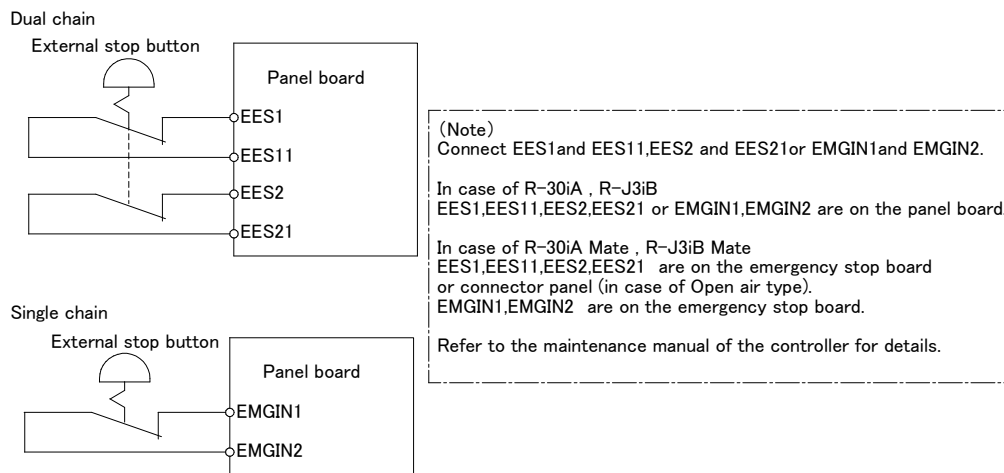


Fig. 2.1 Connection Diagram for External Emergency Stop Button

2.2 SAFETY OF THE PROGRAMMER

While teaching the robot, the operator must enter the work area of the robot. The operator must ensure the safety of the teach pendant operator especially.

- (1) Unless it is specifically necessary to enter the robot work area, carry out all tasks outside the area.
- (2) Before teaching the robot, check that the robot and its peripheral devices are all in the normal operating condition.
- (3) If it is inevitable to enter the robot work area to teach the robot, check the locations, settings, and other conditions of the safety devices (such as the EMERGENCY STOP button, the DEADMAN switch on the teach pendant) before entering the area.
- (4) The programmer must be extremely careful not to let anyone else enter the robot work area.

Our operator panel is provided with an emergency stop button and a key switch (mode switch) for selecting the automatic operation mode (AUTO) and the teach modes (T1 and T2). Before entering the inside of the safety fence for the purpose of teaching, set the switch to a teach mode, remove the key from the mode switch to prevent other people from changing the operation mode carelessly, then open the safety gate. If the safety gate is opened with the automatic operation mode set, the robot stops (Please refer to "STOP TYPE OF ROBOT" in SAFETY for detail of stop type). After the switch is set to a teach mode, the safety gate is disabled. The programmer should understand that the safety gate is disabled and is responsible for keeping other people from entering the inside of the safety fence. (In case of R-30iA Mate Controller standard specification, there is no mode switch. The automatic operation mode and the teach mode is selected by teach pendant enable switch.)

Our teach pendant is provided with a DEADMAN switch as well as an emergency stop button. These button and switch function as follows:

- (1) Emergency stop button: Causes an emergency stop (Please refer to "STOP TYPE OF ROBOT" in SAFETY for detail of stop type) when pressed.
- (2) DEADMAN switch: Functions differently depending on the mode switch setting status.
 - (a) Automatic operation mode: The DEADMAN switch is disabled.
 - (b) Teach mode: Servo power is turned off when the operator releases the DEADMAN switch or when the operator presses the switch strongly.

Note) The DEADMAN switch is provided to stop the robot when the operator releases the teach pendant or presses the pendant strongly in case of emergency. The R-30iA/ R-30iA Mate employs a 3-position DEADMAN switch, which allows the robot to operate when the 3-position DEADMAN switch is pressed to its intermediate point. When the operator releases the DEADMAN switch or presses the switch strongly, the robot stops immediately.

The operator's intention of starting teaching is determined by the controller through the dual operation of setting the teach pendant enable/disable switch to the enable position and pressing the DEADMAN switch. The operator should make sure that the robot could operate in such conditions and be responsible in carrying out tasks safely.

The teach pendant, operator panel, and peripheral device interface send each robot start signal. However the validity of each signal changes as follows depending on the mode switch and the DEADMAN switch of the operator panel, the teach pendant enable switch and the remote condition on the software.

In case of R-30iA Controller or CE or RIA specification of R-30iA Mate Controller

Mode	Teach pendant enable switch	Software remote condition	Teach pendant	Operator panel	Peripheral device
AUTO mode	On	Local	Not allowed	Not allowed	Not allowed
		Remote	Not allowed	Not allowed	Not allowed
	Off	Local	Not allowed	Allowed to start	Not allowed
		Remote	Not allowed	Not allowed	Allowed to start

Mode	Teach pendant enable switch	Software remote condition	Teach pendant	Operator panel	Peripheral device
T1, T2 mode	On	Local	Allowed to start	Not allowed	Not allowed
		Remote	Allowed to start	Not allowed	Not allowed
	Off	Local	Not allowed	Not allowed	Not allowed
		Remote	Not allowed	Not allowed	Not allowed

In case of standard specification of R-30iA Mate Controller

Teach pendant enable switch	Software remote condition	Teach pendant	Peripheral device
On	Ignored	Allowed to start	Not allowed
Off	Local	Not allowed	Not allowed
	Remote	Not allowed	Allowed to start

- (5) (Only when R-30iA Controller or CE or RIA specification of R-30iA Mate controller is selected.) To start the system using the operator's panel, make certain that nobody is in the robot work area and that there are no abnormal conditions in the robot work area.
- (6) When a program is completed, be sure to carry out a test run according to the procedure below.
 - (a) Run the program for at least one operation cycle in the single step mode at low speed.
 - (b) Run the program for at least one operation cycle in the continuous operation mode at low speed.
 - (c) Run the program for one operation cycle in the continuous operation mode at the intermediate speed and check that no abnormalities occur due to a delay in timing.
 - (d) Run the program for one operation cycle in the continuous operation mode at the normal operating speed and check that the system operates automatically without trouble.
 - (e) After checking the completeness of the program through the test run above, execute it in the automatic operation mode.
- (7) While operating the system in the automatic operation mode, the teach pendant operator should leave the robot work area.

2.3 SAFETY OF THE MAINTENANCE ENGINEER

For the safety of maintenance engineer personnel, pay utmost attention to the following.

- (1) During operation, never enter the robot work area.
- (2) Except when specifically necessary, turn off the power of the controller while carrying out maintenance. Lock the power switch, if necessary, so that no other person can turn it on.
- (3) If it becomes necessary to enter the robot operation range while the power is on, press the emergency stop button on the operator panel, or the teach pendant before entering the range. The maintenance personnel must indicate that maintenance work is in progress and be careful not to allow other people to operate the robot carelessly.
- (4) When disconnecting the pneumatic system, be sure to reduce the supply pressure.
- (5) Before the start of teaching, check that the robot and its peripheral devices are all in the normal operating condition.
- (6) Do not operate the robot in the automatic mode while anybody is in the robot work area.
- (7) When you maintain the robot alongside a wall or instrument, or when multiple workers are working nearby, make certain that their escape path is not obstructed.
- (8) When a tool is mounted on the robot, or when any moving device other than the robot is installed, such as belt conveyor, pay careful attention to its motion.
- (9) If necessary, have a worker who is familiar with the robot system stand beside the operator panel and observe the work being performed. If any danger arises, the worker should be ready to press the EMERGENCY STOP button at any time.
- (10) When replacing or reinstalling components, take care to prevent foreign matter from entering the system.

- (11) When handling each unit or printed circuit board in the controller during inspection, turn off the circuit breaker to protect against electric shock.
If there are two cabinets, turn off the both circuit breaker.
- (12) When replacing parts, be sure to use those specified by FANUC.
In particular, never use fuses or other parts of non-specified ratings. They may cause a fire or result in damage to the components in the controller.
- (13) When restarting the robot system after completing maintenance work, make sure in advance that there is no person in the work area and that the robot and the peripheral devices are not abnormal.

3 SAFETY OF THE TOOLS AND PERIPHERAL DEVICES

3.1 PRECAUTIONS IN PROGRAMMING

- (1) Use a limit switch or other sensor to detect a dangerous condition and, if necessary, design the program to stop the robot when the sensor signal is received.
- (2) Design the program to stop the robot when an abnormal condition occurs in any other robots or peripheral devices, even though the robot itself is normal.
- (3) For a system in which the robot and its peripheral devices are in synchronous motion, particular care must be taken in programming so that they do not interfere with each other.
- (4) Provide a suitable interface between the robot and its peripheral devices so that the robot can detect the states of all devices in the system and can be stopped according to the states.

3.2 PRECAUTIONS FOR MECHANISM

- (1) Keep the component cells of the robot system clean, and operate the robot in an environment free of grease, water, and dust.
- (2) Don't use unconfirmed liquid for cutting fluid and cleaning fluid.
- (3) Employ a limit switch or mechanical stopper to limit the robot motion so that the robot or cable does not strike against its peripheral devices or tools.
- (4) Observe the following precautions about the mechanical unit cables. When these attentions are not kept, unexpected troubles might occur.
 - Use mechanical unit cable that have required user interface.
 - Don't add user cable or hose to inside of mechanical unit.
 - Please do not obstruct the movement of the mechanical unit cable when cables are added to outside of mechanical unit.
 - In the case of the model that a cable is exposed, Please do not perform remodeling (Adding a protective cover and fix an outside cable more) obstructing the behavior of the outcrop of the cable.
 - Please do not interfere with the other parts of mechanical unit when install equipments in the robot.
- (5) The frequent power-off stop for the robot during operation causes the trouble of the robot. Please avoid the system construction that power-off stop would be operated routinely. (Refer to bad case example.) Please execute power-off stop after reducing the speed of the robot and stopping it by hold stop or cycle stop when it is not urgent. (Please refer to "STOP TYPE OF ROBOT" in SAFETY for detail of stop type.)
(Bad case example)
 - Whenever poor product is generated, a line stops by emergency stop.

- When alteration was necessary, safety switch is operated by opening safety fence and power-off stop is executed for the robot during operation.
 - An operator pushes the emergency stop button frequently, and a line stops.
 - An area sensor or a mat switch connected to safety signal operate routinely and power-off stop is executed for the robot.
- (6) Robot stops urgently when collision detection alarm (SV050) etc. occurs. The frequent urgent stop by alarm causes the trouble of the robot, too. So remove the causes of the alarm.

4 SAFETY OF THE ROBOT MECHANISM

4.1 PRECAUTIONS IN OPERATION

- (1) When operating the robot in the jog mode, set it at an appropriate speed so that the operator can manage the robot in any eventuality.
- (2) Before pressing the jog key, be sure you know in advance what motion the robot will perform in the jog mode.

4.2 PRECAUTIONS IN PROGRAMMING

- (1) When the work areas of robots overlap, make certain that the motions of the robots do not interfere with each other.
- (2) Be sure to specify the predetermined work origin in a motion program for the robot and program the motion so that it starts from the origin and terminates at the origin.
Make it possible for the operator to easily distinguish at a glance that the robot motion has terminated.

4.3 PRECAUTIONS FOR MECHANISMS

- (1) Keep the work areas of the robot clean, and operate the robot in an environment free of grease, water, and dust.

4.4 PROCEDURE TO MOVE ARM WITHOUT DRIVE POWER IN EMERGENCY OR ABNORMAL SITUATIONS

For emergency or abnormal situations (e.g. persons trapped in or by the robot), brake release unit can be used to move the robot axes without drive power.

Please refer to controller maintenance manual and mechanical unit operator's manual for using method of brake release unit and method of supporting robot.

5 SAFETY OF THE END EFFECTOR

5.1 PRECAUTIONS IN PROGRAMMING

- (1) To control the pneumatic, hydraulic and electric actuators, carefully consider the necessary time delay after issuing each control command up to actual motion and ensure safe control.
- (2) Provide the end effector with a limit switch, and control the robot system by monitoring the state of the end effector.

6 STOP TYPE OF ROBOT

The following three robot stop types exist:

Power-Off Stop (Category 0 following IEC 60204-1)

Servo power is turned off and the robot stops immediately. Servo power is turned off when the robot is moving, and the motion path of the deceleration is uncontrolled.

The following processing is performed at Power-Off stop.

- An alarm is generated and servo power is turned off.
- The robot operation is stopped immediately. Execution of the program is paused.

Controlled stop (Category 1 following IEC 60204-1)

The robot is decelerated until it stops, and servo power is turned off.

The following processing is performed at Controlled stop.

- The alarm "SRVO-199 Controlled stop" occurs along with a decelerated stop. Execution of the program is paused.
- An alarm is generated and servo power is turned off.

Hold (Category 2 following IEC 60204-1)

The robot is decelerated until it stops, and servo power remains on.

The following processing is performed at Hold.

- The robot operation is decelerated until it stops. Execution of the program is paused.



WARNING

The stopping distance and stopping time of Controlled stop are longer than the stopping distance and stopping time of Power-Off stop. A risk assessment for the whole robot system, which takes into consideration the increased stopping distance and stopping time, is necessary when Controlled stop is used.

When the E-Stop button is pressed or the FENCE is open, the stop type of robot is Power-Off stop or Controlled stop. The configuration of stop type for each situation is called *stop pattern*. The stop pattern is different according to the controller type or option configuration.

There are the following 3 Stop patterns.

Stop pattern	Mode	E-Stop button	External E-Stop	FENCE open	SVOFF input	Servo disconnect
A	AUTO	P-Stop	P-Stop	C-Stop	C-Stop	P-Stop
	T1	P-Stop	P-Stop	-	C-Stop	P-Stop
	T2	P-Stop	P-Stop	-	C-Stop	P-Stop
B	AUTO	P-Stop	P-Stop	P-Stop	P-Stop	P-Stop
	T1	P-Stop	P-Stop	-	P-Stop	P-Stop
	T2	P-Stop	P-Stop	-	P-Stop	P-Stop
C	AUTO	C-Stop	C-Stop	C-Stop	C-Stop	C-Stop
	T1	P-Stop	P-Stop	-	C-Stop	P-Stop
	T2	P-Stop	P-Stop	-	C-Stop	P-Stop

P-Stop: Power-Off stop

C-Stop: Controlled stop

-: Not stop

The following table indicates the Stop pattern according to the controller type or option configuration.

Option	R-30iA				R-30iA Mate		
	Standard (Single)	Standard (Dual)	RIA type	CE type	Standard	RIA type	CE type
Standard	B (*)	A	A	A	A (**)	A	A
Stop type set (Stop pattern C) (A05B-2500-J570)	N/A	N/A	C	C	N/A	C	C

(*) R-30iA standard (single) does not have servo disconnect.

(**) R-30iA Mate Standard does not have servo disconnect, and the stop type of SVOFF input is Power-Off stop.

The stop pattern of the controller is displayed in "Stop pattern" line in software version screen. Please refer "Software version" in operator's manual of controller for the detail of software version screen.

"Stop type set (Stop pattern C)" option

"Stop type set (Stop pattern C)"(A05B-2500-J570) is an optional function. When this option is loaded, the stop type of the following alarms becomes Controlled stop but only in AUTO mode. In T1 or T2 mode, the stop type is Power-Off stop which is the normal operation of the system.

Alarm	Condition
SRVO-001 Operator panel E-stop	Operator panel E-stop is pressed.
SRVO-002 Teach pendant E-stop	Teach pendant E-stop is pressed.
SRVO-007 External emergency stops	External emergency stop input (EES1-EES11, EES2-EES21) is open. (R-30iA controller)
SRVO-194 Servo disconnect	Servo disconnect input (SD4-SD41, SD5-SD51) is open. (R-30iA controller)
SRVO-218 Ext.E-stop/Servo Disconnect	External emergency stop input (EES1-EES11, EES2-EES21) is open. (R-30iA Mate controller)
SRVO-408 DCS SSO Ext Emergency Stop	In DCS Safe I/O connect function, SSO[3] is OFF.
SRVO-409 DCS SSO Servo Disconnect	In DCS Safe I/O connect function, SSO[4] is OFF.

Controlled stop is different from Power-Off stop as follows:

- In Controlled stop, the robot is stopped on the program path. This function is effective for a system where the robot can interfere with other devices if it deviates from the program path.
- In Controlled stop, physical impact is less than Power-Off stop. This function is effective for systems where the physical impact to the mechanical unit or EOAT (End Of Arm Tool) should be minimized.

- The stopping distance and stopping time of Controlled stop is longer than the stopping distance and stopping time of Power-Off stop, depending on the robot model and axis. Please refer the operator's manual of a particular robot model for the data of stopping distance and stopping time.

This function is available only in CE or RIA type hardware.

When this option is loaded, this function can not be disabled.

The stop type of DCS Position and Speed Check functions is not affected by the loading of this option.

**WARNING**

The stopping distance and stopping time of Controlled stop are longer than the stopping distance and stopping time of Power-Off stop. A risk assessment for the whole robot system, which takes into consideration the increased stopping distance and stopping time, is necessary when this option is loaded.

TABLE OF CONTENTS

SAFETY PRECAUTIONS	s-1
1 INTRODUCTION	1
1.1 MANUAL PLAN	1
2 OVERVIEW	4
2.1 APPLICATION TOOL SOFTWARE	5
2.1.1 System Setting	5
2.1.2 Jog Feed of the Robot	5
2.1.3 Program	5
2.1.4 Test Operation (Test Execution)	6
2.1.5 Automatic Operation (Operation Execution)	6
2.2 ROBOT	7
2.3 CONTROLLER	8
2.3.1 Teach Pendant	9
2.3.1.1 Display Screen of the Teach Pendant	14
2.3.2 Operator Panel	17
2.3.3 Remote Controller	18
2.3.4 CRT/KB	19
2.3.5 Communication	19
2.3.6 Input/Output	19
2.3.7 Peripheral I/O	19
2.3.8 Motion of the Robot	19
2.3.9 Emergency Stop Devices	20
2.3.10 Extended Axis	20
3 SETTING UP THE ROBOT SYSTEM	21
3.1 I/O	21
3.1.1 Digital I/O	30
3.1.2 Group I/O	34
3.1.3 Analog I/O	38
3.2 ROBOT I/O	41
3.3 PERIPHERAL I/O	44
3.4 OPERATOR'S PANEL I/O	51
3.5 I/O Link SCREEN	53
3.5.1 I/O Link List Screen	53
3.5.2 Model B Unit List Screen	55
3.5.3 Number of ports Setting Screen	56
3.6 I/O CONNECTION FUNCTION	57
3.7 SIMULATED INPUT SKIP FUNCTION	59
3.8 SETTING AUTOMATIC OPERATION	61
3.8.1 Robot Service Request (RSR)	62
3.8.2 Program Number Selection (PNS)	65
3.8.3 STYLE	68
3.8.4 Prog Select Screen	71
3.8.5 Cell Interface I/O	77
3.9 SETTING COORDINATE SYSTEMS	81
3.9.1 Setting a Tool Coordinate System	83
3.9.2 Setting a User Coordinate System	93

3.9.3	Setting a Jog Coordinate System.....	103
3.10	SETTING A REFERENCE POSITION.....	108
3.11	JOINT OPERATING AREA	112
3.12	USER ALARM	113
3.13	VARIABLE AXIS AREAS.....	115
3.14	INTERFERENCE PREVENTION AREA FUNCTION.....	116
3.15	SYSTEM CONFIG MENU	120
3.16	SETTING THE GENERAL ITEMS	126
3.17	OTHER SETTINGS	127
4	PROGRAM STRUCTURE.....	128
4.1	PROGRAM DETAIL INFORMATION.....	130
4.1.1	Program Name.....	130
4.1.2	Program Comment.....	131
4.1.3	Subtype.....	131
4.1.4	Group Mask.....	132
4.1.5	Write Protection	132
4.1.6	Interruption Disable.....	132
4.1.7	Stack Size	133
4.2	LINE NUMBER, PROGRAM END SYMBOL, AND ARGUMENT	135
4.3	MOTION INSTRUCTIONS	136
4.3.1	Motion Format.....	137
4.3.2	Position Data	141
4.3.3	Feedrate	146
4.3.4	Positioning Path.....	148
4.3.5	Additional Motion Instructions	149
4.4	PALLETIZING INSTRUCTIONS	161
4.4.1	Palletizing Instruction.....	161
4.4.2	Palletizing Motion Instruction.....	162
4.4.3	Palletizing End Instruction	162
4.5	REGISTER INSTRUCTIONS.....	162
4.5.1	Register Instructions.....	163
4.5.2	Position Register Instructions.....	164
4.5.3	Position Register Axis Instructions	165
4.5.4	Arithmetic Palletizing Register Instructions.....	167
4.5.5	String Register, String Instructions	169
4.6	I/O INSTRUCTIONS	171
4.6.1	Digital I/O Instructions.....	172
4.6.2	Robot I/O Instructions	173
4.6.3	Analog I/O Instructions	174
4.6.4	Group I/O Instruction	175
4.7	BRANCH INSTRUCTIONS.....	175
4.7.1	Label Instruction.....	176
4.7.2	Program End Instruction	176
4.7.3	Unconditional Branch Instructions.....	176
4.7.4	Conditional Branch Instructions.....	177
4.7.5	Arguments	180
4.8	WAIT INSTRUCTIONS	188
4.8.1	Time-specified Wait Instruction.....	188
4.8.2	Conditional Wait Instructions	188
4.9	SKIP CONDITION INSTRUCTION	191

4.10	OFFSET CONDITION INSTRUCTION	193
4.11	TOOL OFFSET CONDITION INSTRUCTIONS	194
4.12	FRAME INSTRUCTIONS	194
4.13	PROGRAM CONTROL INSTRUCTIONS	196
4.13.1	Halt Instruction	196
4.13.2	Abort Instruction	196
4.14	OTHER INSTRUCTIONS	197
4.14.1	RSR Instruction	197
4.14.2	User Alarm Instruction	197
4.14.3	Timer Instruction	198
4.14.4	Override Instruction	198
4.14.5	Comment Instruction	199
4.14.6	Multi-language Comment Instruction	199
4.14.7	Message Instruction	199
4.14.8	Parameter Instruction	200
4.14.9	Maximum Speed Instructions	202
4.15	MULTIAXIS CONTROL INSTRUCTIONS	203
4.15.1	Program Execution Instruction	203
4.16	OPERATION GROUP INSTRUCTIONS	204
4.16.1	Asynchronous Operation Group Instruction	204
4.16.2	Synchronous Operation Group Instruction	204
5	PROGRAMMING	206
5.1	TIPS ON EFFECTIVE PROGRAMMING	207
5.1.1	Motion Instructions	207
5.1.2	Predefined Position	208
5.2	TURNING ON THE POWER AND JOG FEED	209
5.2.1	Turning On the Power and Turning Off the Power	209
5.2.2	Three-Mode Switch	211
5.2.3	Moving the Robot by Jog Feed	215
5.3	CREATING A PROGRAM	223
5.3.1	Registering a Program	224
5.3.2	Changing a Standard Motion Instruction	228
5.3.3	Teaching a Motion Instruction	230
5.3.4	Teaching an Supplementary Motion Instruction	232
5.3.5	Teaching a Control Instruction	235
5.3.6	TP Start Prohibition	240
5.4	CHANGING A PROGRAM	242
5.4.1	Selecting a Program	242
5.4.2	Changing a Motion Instruction	243
5.4.3	Changing a Control Instruction	252
5.4.4	Program Edit Instructions	253
5.5	PROGRAM OPERATION	269
5.5.1	Changing Program Information	269
5.6	BACKGROUND EDITING	273
5.7	SINGULAR POINT CHECK FUNCTION	285
5.8	OTHER EDITING FUNCTION	286
5.8.1	Auto Position Renumbering	286
5.8.2	Fixed Program Name	287
5.8.3	Filtered Program List	288
5.8.3.1	The way to make original program name of program creation screen into initial strings	288

5.8.3.2	The way to register over six initial strings.....	288
6	EXECUTING A PROGRAM.....	290
6.1	PROGRAM HALT AND RECOVERY	290
6.1.1	Halt by an Emergency Stop and Recovery	291
6.1.2	Halt by a Hold and Recovery	292
6.1.3	Halt Caused by an Alarm	293
6.2	EXECUTING A PROGRAM	296
6.2.1	Starting a Program.....	296
6.2.2	Robot Motion	297
6.2.3	Resuming a Program	299
6.3	TESTING	303
6.3.1	Specifying Test Execution.....	304
6.3.2	Step Test.....	305
6.3.3	Continuous Test.....	308
6.3.4	Program Look/Monitor.....	309
6.4	MANUAL I/O CONTROL	310
6.4.1	Forced Output.....	311
6.4.2	Simulated I/O	312
6.4.3	Wait Release.....	313
6.5	OPERATING THE HAND MANUALLY	314
6.6	AUTOMATIC OPERATION	315
6.6.1	Automatic Operation by Robot Start Request (RSR).....	316
6.6.2	Automatic Operation with Program Number Selection (PNS).....	318
6.6.3	External Override Selection Function	320
6.7	ONLINE POSITION MODIFICATION	321
7	STATUS DISPLAY	328
7.1	LEDS ON THE TEACH PENDANT	328
7.2	USER SCREEN.....	329
7.3	REGISTERS	329
7.4	POSITION REGISTERS.....	331
7.5	PALLETIZING REGISTERS	334
7.6	STRING REGISTERS	335
7.7	CURRENT POSITION	336
7.8	SYSTEM VARIABLES	338
7.9	PROGRAM TIMER	340
7.10	SYSTEM TIMER.....	342
7.11	EXECUTION HISTORY	342
7.12	MEMORY USE STATUS DISPLAY	344
7.13	SAFETY SIGNALS	345
7.14	PROGRAM STATUS	346
8	FILE INPUT/OUTPUT	348
8.1	FILE INPUT/OUTPUT UNITS	348
8.1.1	Memory Card	351
8.2	SETTING A COMMUNICATION PORT	353
8.3	FILES.....	357
8.3.1	Program File.....	357
8.3.2	Default Logic File	357

8.3.3	System File/Application File	358
8.3.4	Data File	358
8.3.5	ASCII File	358
8.4	SAVING FILES	358
8.4.1	Saving with Program List Screen	359
8.4.2	Saving all the Program Files Using the File Screen	360
8.4.3	Saving with a Function Menu	364
8.4.4	File Manipulation	366
8.4.5	ASCII save	368
8.5	LOADING FILES.....	369
8.5.1	Loading Using the Program List Screen	370
8.5.2	Loading a Specified Program File Using the File Screen	371
8.6	PRINTING FILES.....	376
8.7	Subdirectories.....	379
8.8	AUTOMATIC BACKUP	381
8.8.1	Overview of Automatic Backup	381
8.8.2	Usable Memory Cards.....	382
8.8.3	Setting of Automatic Backup	382
8.8.4	Perform Automatic Backup (Prior to V7.40)	383
8.8.5	Perform Automatic Backup (V7.40 or later).....	384
8.8.6	Version Management.....	386
8.8.7	Restore the Backup.....	386
8.9	IMAGE BACKUP FUNCTION	387
8.10	ASCII UPLOAD FUNCTION	394
8.10.1	Overview	394
8.10.2	Loading an ASCII Teach Pendant Program from the Teach Pendant	394
8.10.3	Viewing ASCII Upload Errors	395
8.10.4	Example ASCII File	397
8.11	FILE MEMORY	398
9	UTILITY	399
9.1	MACRO INSTRUCTION	400
9.1.1	Setting Macro Instructions	401
9.1.2	Executing Macro Instructions	405
9.2	SHIFT FUNCTIONS	409
9.2.1	Program Shift Function	410
9.2.2	Mirror Shift Function	416
9.2.3	Angle-input Shift Function.....	419
9.3	COORDINATE SYSTEM CHANGE SHIFT FUNCTIONS.....	424
9.4	SOFT FLOAT FUNCTION	428
9.5	CONTINUOUS ROTATION FUNCTION.....	434
9.6	POSITION REGISTER LOOK-AHEAD EXECUTION FUNCTION.....	439
9.7	OPERATION GROUP DO OUTPUT FUNCTION	441
9.8	TIME BEFORE FUNCTION	442
9.9	DISTANCE BEFORE FUNCTION	447
9.9.1	Overview	447
9.9.2	Specification.....	447
9.9.3	Configuration	448
9.9.4	Instruction.....	448
9.9.5	Entering Distance before	455
9.9.6	Caution and Limitations	458

9.10	STATE MONITORING FUNCTION	458
9.11	AUTOMATIC ERROR RECOVERY FUNCTION	466
9.11.1	Overview	466
9.11.2	Outline of the Automatic Error Recovery Function	467
9.11.3	Defining a Resume Program	469
9.11.4	Teaching the RETURN_PATH_DSBL Instruction	470
9.11.5	Setting the Automatic Error Recovery Function	470
9.11.6	Flowchart for Resuming a Suspended Program	477
9.11.7	Manual Operation Screen of the Automatic Error Recovery Function	478
9.11.8	Execution of the Resume Program from The Teach Pendant and Test Mode.....	480
9.11.9	Changing Conditions for Executing the Resume Program.....	480
9.11.10	Other Specifications and Restrictions	481
9.11.11	Warnings	482
9.12	REMOTE TCP FUNCTION.....	482
9.12.1	Summary	482
9.12.2	Setup.....	485
9.13	HIGH-SENSITIVITY COLLISION DETECTION	487
9.13.1	Overview	487
9.13.2	Specification.....	487
9.13.3	Settings	487
9.13.4	Collision Guard Setup Screen	488
9.13.5	Program Instructions	489
9.13.5.1	COL DETECT ON / COL DETECT OFF.....	489
9.13.5.2	COL GUARD ADJUST	490
9.13.6	Cautions.....	492
9.14	LOAD SETTING	493
9.14.1	Overview	493
9.14.2	Motion Performance Screens	493
9.14.3	Program Instructions	494
9.15	LOAD ESTIMATION	496
9.15.1	Overview	496
9.15.2	Operating Procedure.....	496
9.15.3	Load Estimation Procedure (for 6-Axis Robots).....	496
9.15.4	Calibration Procedure (for 6-Axis Robots)	499
9.15.5	Other Related Matters.....	502
9.16	COLLISION DETECTION for AUXILIARY AXIS	503
9.16.1	General	503
9.16.2	Caution	503
9.16.3	Initial Setting.....	503
9.16.4	Tuning Procedure	503
9.17	GRAVITY COMPENSATION	505
9.17.1	System Variables.....	505
9.17.2	MOTION Screen	505
9.17.3	Mastering.....	506
9.18	PASSWORD FUNCTION	506
9.18.1	Overview of the Password Function.....	506
9.18.2	Password Operations by the Install User.....	508
9.18.3	Disabling the Password Function	510
9.18.4	Password Operations by Program Users and Setup Users	511
9.18.5	Password Configuration File	515
9.18.5.1	Overview	515
9.18.6	XML file for Password function.....	517
9.18.6.1	Software parts ID, Screen ID.....	517

9.18.6.2	Basic tag	518
9.18.6.3	Comment tag.....	519
9.18.6.4	Level name tag.....	519
9.18.6.5	Local Labels tag.....	520
9.18.6.6	Screen Access tag	520
9.18.6.7	Features tag.....	521
9.18.6.8	Default Menu Type tag.....	521
9.18.6.9	Default Screen tag.....	522
9.18.6.10	FCTN/Display Menu Access tag	522
9.18.6.11	EDCMD Access tag.....	523
9.18.7	Password Log	524
9.18.8	Screen Restrictions According to Password Level.....	525
9.18.9	Password Auto Login Function.....	527
9.19	MIXED LOGIC INSTRUCTION.....	528
9.20	PMC MONITOR FUNCTION	543
9.21	PMC EDIT FUNCTION	545
9.22	OPERATION LOG BOOK.....	547
9.22.1	Overview	547
9.22.2	Recorded Events.....	549
9.22.3	Setting Up Book	553
9.22.4	Operations	553
9.22.5	Extended Alarm Log	557
9.22.5.1	Setup	557
9.22.5.2	How to display alarm log.....	558
9.23	ORIGINAL PATH RESUME.....	560
9.24	PROGRAM TOOLBOX.....	564
9.24.1	Soft Limit Setting	564
9.25	ADVANCED CONSTANT PATH.....	567
9.25.1	Overview	567
9.25.2	Linear Distance.....	567
9.25.2.1	How it Works.....	568
9.25.2.2	Limitations.....	570
9.25.2.3	How to use linear distance.....	570
9.25.3	Corner Region Termination Type (CRy)	571
9.25.3.1	CRy versus segment distance	572
9.25.3.2	Teaching techniques	573
9.25.3.3	Constant path with respect to program speed changes	573
9.25.3.4	Compatibilities and limitations	573
9.25.3.5	Process speed.....	575
9.25.3.6	Limitations.....	577
9.25.4	Max Speed.....	578
9.25.4.1	Limitations.....	578
9.26	HIGH PERFORMANCE SOFTFLOAT FUNCTION	580
9.26.1	Outline	580
9.26.2	Relation to Current Softfloat Function	580
9.26.3	Procedure of operation	580
9.26.4	Knowhow of Softfloat	583
9.26.5	Knowhow of Cartesian Softfloat	584
9.27	MULTITASKING FUNCTION.....	587
9.27.1	Overview	587
9.27.2	Functions	587
9.27.2.1	Notes on program creation.....	587
9.27.2.2	How to start programs in multitasking	587
9.27.2.3	Behavior of main program and subprogram	588

	9.27.2.4	Notes on main program and subprogram.....	589
	9.27.3	Monitor.....	589
	9.27.4	Halt and Forced Termination.....	590
	9.27.5	Cycle Stop Signal (CSTOPI).....	591
9.28		ERROR SEVERITY TABLE.....	592
	9.28.1	Error Severity Table Overview	592
	9.28.2	Modifying Error Severity	592
9.29		DIAGNOSTIC LOG.....	596
9.30		AUTOMATIC ERROR RECOVERY FUNCTION FOR MULTI-TASK	597
	9.30.1	Overview	597
	9.30.2	Automatic Error Recovery Function	597
	9.30.3	Resume Program Function	597
	9.30.4	Fast Exit/entry Feature	598
	9.30.5	RESUME_PROG Instruction.....	599
	9.30.6	RETURN_PATH_DSBL Instruction	601
	9.30.7	MAINT_PROG Instruction	602
	9.30.8	Setting of the Automatic Error Recovery Function.....	605
	9.30.9	Manual Operation Screen of the Resume Program Function	616
	9.30.10	Execution of the Resume Program from the Teach Pendant and Test Mode	618
	9.30.11	Changing Conditions for Executing the Resume Program.....	619
	9.30.12	Other Specifications and Restrictions	619
	9.30.13	Warnings (Be sure to read this section for safety.).....	620
9.31		PATH SWITCHING FUNCTION	621
9.32		AUXILIARY AXIS SERVO OFF (LOCAL STOP) FUNCTION	626
	9.32.1	Introduction	626
	9.32.2	Specification.....	627
	9.32.3	Constraints.....	629
	9.32.4	Settings	629
	9.32.5	Attention.....	633
	9.32.6	Programming	633
9.33		STITCH FUNCTION	637
	9.33.1	Overview	637
	9.33.2	Specification.....	637
	9.33.2.1	Instruction.....	637
	9.33.2.2	Stitch condition.....	637
	9.33.2.3	Flow of Stitch process	639
	9.33.2.4	Other specifications	640
	9.33.3	Adjustment	641
	9.33.4	Limitations.....	642
	9.33.5	Appendix	643
9.34		DUAL MOTOR DRIVE	643
	9.34.1	Introduction	643
	9.34.2	Setup.....	644
9.35		MULTI ROBOT CONTROL FUNCTION	645
	9.35.1	Introduction	645
	9.35.1.1	Specification	646
	9.35.1.2	Software Options	647
	9.35.2	Standard Operator Panel.....	647
	9.35.2.1	BATTERY ALARM	648
	9.35.2.2	TP ENABLED.....	648
	9.35.2.3	ALARM.....	648
	9.35.2.4	REMOTE.....	648
	9.35.2.5	REMOTE/LOCAL Key.....	648
	9.35.2.6	ALARM RESET.....	648

	9.35.2.7	START.....	648
	9.35.2.8	HOLD.....	648
	9.35.2.9	USER1/USER2.....	649
	9.35.2.10	CONNECT/ISOLATE Key.....	649
9.35.3		User Operator Panel(UOP).....	649
	9.35.3.1	Modification and Addition of Signal.....	650
	9.35.3.2	HOLD#1, #2.....	650
	9.35.3.3	START#1, #2.....	651
	9.35.3.4	RSR.....	651
	9.35.3.5	PNSTROBE#1, #2, PROD_START#1, #2.....	651
	9.35.3.6	CMDENBL#1, #2.....	651
	9.35.3.7	SYRSY#1, #2.....	652
	9.35.3.8	ATPERCH.....	652
	9.35.3.9	PROGRUN#1, #2, PAUSED#1, #2.....	652
	9.35.3.10	HELD#1, #2.....	653
	9.35.3.11	FAULT#1, #2.....	653
9.35.4		ROBOT I/O.....	653
9.35.5		Toggling robot.....	654
9.35.6		Select program.....	654
9.35.7		Teaching and modifying position.....	656
9.35.8		Run program.....	656
	9.35.8.1	Run program by Teach Pendant.....	656
	9.35.8.2	Run program by operator panel.....	657
	9.35.8.3	Run two programs at the same time.....	657
	9.35.8.4	Select program at running or paused.....	658
	9.35.8.5	Robot Service Request (RSR).....	658
	9.35.8.6	Program Number Select (PNS).....	658
9.35.9		Stop program.....	659
	9.35.9.1	Pause program by Teach Pendant.....	659
	9.35.9.2	Pause program by Operator Panel.....	659
	9.35.9.3	Pause program by UOP.....	659
	9.35.9.4	Abort program by UOP.....	660
9.35.10		System variables.....	661
9.36		Robot Isolation function.....	661
	9.36.1	Introduction.....	661
	9.36.2	Method of operating.....	662
	9.36.2.1	Operation panel.....	662
	9.36.2.2	SOP.....	662
	9.36.3	System variable: \$ROBOT_ISOLC.....	662
	9.36.4	Teaching to motion instruction.....	663
9.37		Multi UOP Interface Function.....	663
	9.37.1	Introduction.....	663
	9.37.1.1	Specification.....	664
	9.37.1.2	Software Options.....	664
	9.37.1.3	The difference from Multi Robot Control function about UOP.....	664
	9.37.2	Peripheral I/O.....	665
	9.37.2.1	Setting of the number of UOP set.....	666
	9.37.2.2	Modification and Addition of Signal.....	667
	9.37.2.3	PMC.....	668
	9.37.2.4	HOLD#1~#N.....	669
	9.37.2.5	CSTOPI#1~#N.....	669
	9.37.2.6	START#1~#N.....	670
	9.37.2.7	RSR.....	670
	9.37.2.8	PNSTROBE#1~#N,PROD_START#1~#N.....	671
	9.37.2.9	CMDENBL#1~#N.....	671
	9.37.2.10	SYSRDY#1~#N.....	671

	9.37.2.11	PROGRUN#1~#N,PAUSED#1~#N.....	671
	9.37.2.12	HELD#1~#N	672
	9.37.2.13	FAULT#1~#N	672
	9.37.2.14	ATPERCH.....	672
9.37.3		Select program.....	672
	9.37.3.1	Setting of motion groups intended by each UOP set.....	673
	9.37.3.2	Select program at running or paused	674
	9.37.3.3	Program edit screen display when operating motion group is changed.....	675
9.37.4		Run program.....	676
	9.37.4.1	Robot Service Request (RSR)	676
	9.37.4.2	Program Number Select (PNS).....	676
9.37.5		Stop program	676
	9.37.5.1	Pause program by UOP	676
	9.37.5.2	Abort program by UOP	676
	9.37.5.3	Other	677
9.37.6		System variables.....	677
9.38		Error Code Output function.....	679
	9.38.1	Introduction	679
	9.38.2	Specification.....	679
	9.38.2.1	Types of Alarms	679
	9.38.2.2	Input and Output signals.....	679
9.38.3		Meaning of Alarm Code.....	680
	9.38.3.1	Severity of Alarm	681
	9.38.3.2	Alarm ID.....	682
	9.38.3.3	Alarm Number	682
9.39		DATA MONITOR	683
	9.39.1	Data Monitor Setup	685
	9.39.2	Data Monitor Schedule.....	690
	9.39.3	Programming	693
	9.39.4	Data Monitor Chart	694
9.40		BRAKE CHECK FUNCTION.....	695
	9.40.1	Overview	695
	9.40.2	Initial setting before use	695
	9.40.3	Start brake check	696
	9.40.4	Results of brake check.....	697
	9.40.5	Limitations.....	697
	9.40.6	Caution	697
9.41		PANEL WIZARD.....	698
	9.41.1	Overview	698
	9.41.2	Software Requirements and Explanation	698
	9.41.3	Hardware Requirements	698
	9.41.4	Setting up.....	698
	9.41.4.1	Overview	698
	9.41.4.2	Available iPendant Controls	701
	9.41.4.3	Setting up Fast Label	702
	9.41.4.4	Setting up fast lamp	703
	9.41.4.5	Setting up fast switch.....	705
	9.41.4.6	Addition of button change control.....	706
	9.41.4.7	Addition of command button control.....	710
	9.41.4.8	Addition of edit box control	711
	9.41.4.9	Addition of label control.....	712
	9.41.4.10	Addition of toggle button control	712
	9.41.4.11	Addition of toggle lamp control	714
	9.41.4.12	Modification of panel	715
	9.41.4.13	Modification of control.....	716

	9.41.4.14 Delete of control	717
	9.41.4.15 Cut/Copy paste of control	718
	9.41.4.16 Modification of page	719
	9.41.4.17 Re-creation of panel	720
9.41.5	Run KAREL Program by Panel	721
	9.41.5.1 Caution for creation of KAREL program	722
	9.41.5.2 Creation of run button	722
9.41.6	Usage	723
	9.41.6.1 Display of panel	723
	9.41.6.2 Backup/restore	724
9.42	ENHANCED MIRROR IMAGE	724
	9.42.1 Parallel Mirror Image	726
	9.42.2 Rotational Mirror Image	728
	9.42.3 Mirror Image using Existing Frames and Mirror Planes, with Controlled Orientation	729
	9.42.4 Mirror Image of Extended Axes	730
9.43	KAREL USE SUPPORT FUNCTION	738
	9.43.1 Overview	738
	9.43.2 KAREL Config	739
	9.43.2.1 Start the KAREL Config screen	739
	9.43.2.2 Use KAREL Config screen	740
	9.43.2.3 Run KAREL program	741
	9.43.2.4 Abort the program	742
	9.43.2.5 Start mode config	744
	9.43.2.6 Detail of KAREL Config	745
	9.43.2.7 Limitation and Caution of KAREL Config	746
	9.43.2.8 Cycle Power	747
	9.43.3 Custom Menu	748
	9.43.3.1 Starting custom menu	748
	9.43.3.2 Set custom menu	749
	9.43.3.3 Delete set	751
9.44	KAREL PROGRAM EXECUTION HISTORY RECORD	752
	9.44.1 Overview	752
	9.44.2 Hardware and Software	753
	9.44.2.1 Hardware and software requirements	753
	9.44.2.2 Hardware	753
	9.44.2.3 Software	753
	9.44.2.4 Performance	754
	9.44.3 Setup and Operations	754
	9.44.3.1 Setting up the KAREL program execution history record	754
	9.44.3.2 Dump selections screen	754
	9.44.3.3 Task selection screen	755
	9.44.3.4 Stop Logging Tasks Screen	757
	9.44.3.5 List selected tasks screen	758
	9.44.3.6 Event class selection screen	758
	9.44.3.7 Event detail selection screen	760
	9.44.3.8 Enable or disable all event logging	761
	9.44.4 Logging Events	762
	9.44.4.1 Setting up events	762
	9.44.4.2 Logging events to an ASCII file	763
	9.44.4.3 ASCII file general event information	763
	9.44.4.4 ASCII file specific event information	763
	9.44.5 Examples	767
	9.44.5.1 Overview	767
	9.44.5.2 KAREL program example	767
	9.44.5.3 Teach Pendant Program Example	768

9.44.5.4	ASCII File Example	768
9.45	TORQUE LIMIT FUNCTION.....	770
9.45.1	Introduction	770
9.45.2	Torque limit function feature.....	770
9.45.3	Torque Limit Multi-axis Setup Function.....	772
9.45.3.1	Torque limit multi-axis setup function	772
9.45.3.2	Error code of torque limit multi-axis setup function	772
9.45.4	Limitations.....	773
9.45.5	Caution	773
9.46	TCP SPEED OUTPUT.....	773
9.46.1	Overview	773
9.46.2	Limitations.....	774
9.46.3	Setting up TCP speed output	775
9.46.4	TCP speed output instruction	776
9.47	TP DRAM/FILE STORAGE FUNCTION	778
9.47.1	Storages	778
9.47.1.1	CMOS programs.....	778
9.47.1.2	SHADOW programs.....	778
9.47.1.3	SHADOW ONDEMAND programs	779
9.47.1.4	FILE programs.....	779
9.47.2	Storage Configuration	779
9.47.3	Save / Load programs.....	783
9.47.3.1	Save / Load TP files.....	783
9.47.3.2	Copy programs	783
9.47.3.3	Save / Load LS files.....	783
9.47.3.4	Make backup of programs	783
9.47.4	Loading process in program execution.....	784
9.47.5	Program exchange function without enough memory space.....	784
9.47.6	Precaution.....	785
9.47.6.1	Cause and Remedy for alarm occurrence	785
9.47.6.2	Effect of alarm occurrence.....	786
9.48	CYCLE TIME TRACKING.....	786
9.48.1	Overview	786
9.48.1.1	Cycle Time Categories.....	786
9.48.1.2	Cycle Time Screen Items	787
9.48.2	Display Modes of Cycle Time.....	788
9.48.2.1	Cycle Mode.....	788
9.48.2.2	Hourly Mode.....	789
9.48.2.3	Line-by-Line Mode.....	789
9.48.3	Tracking Cycle Time.....	790
9.48.3.1	Change Display Format of Cycle Time Data.....	790
9.48.3.2	Save Cycle Time Data	790
9.48.3.3	Update Cycle Time Data.....	791
9.48.3.4	Display Target Cycle Time Line.....	791
9.48.3.5	Display Reference Cycle Time Line	792
9.48.4	Execution example of Spot program	793
9.49	ROBOT TOOLS OF ROBOT HOMEPAGE	794
9.49.1	Setup for ROBOT TOOLS.....	796
9.49.1.1	Setup TCP/IP	796
9.49.1.2	Setup HTTP AUTHENTICATION.....	799
9.49.2	Usage of ROBOT TOOLS	802
9.50	LINE REMARK INSTRUCTION	805
9.50.1	Overview	805
9.50.2	Function.....	805
9.50.3	Teach LINE REMARK instruction	806

9.50.4	Edit REMARKED instructions	808
9.50.4.1	[EDCMD] with LINE REMARK	808
9.50.4.2	ASCII program with LINE REMARK	809
9.50.5	Consideration.....	809
9.51	FINISHING FUNCTION PACKAGE	809
9.51.1	Special Jog Sequence (J559)	809
9.51.1.1	Usage	809
9.51.1.2	Setup	810
9.51.1.3	Application	810
9.51.2	Motion Instruction Enhanced Editing (J560)	810
9.51.2.1	Motion Instruction Insert and Instruction delete function	810
9.51.2.2	Data Offset function	812
9.52	FOR/ENDFOR STATEMENT	820
9.52.1	Overview	820
9.52.2	FOR/ENDFOR statement specification.....	820
9.52.2.1	Form of FOR statement	820
9.52.2.2	FOR statement execution.....	821
9.52.2.3	Form of ENDFOR statement	821
9.52.2.4	ENDFOR statement execution.....	821
9.52.3	Teach FOR/ENDFOR statement	821
9.52.3.1	How to teach FOR/ENDFOR statement.....	821
9.52.3.2	FOR/ENDFOR statement Combination	824
9.52.4	Execution examples.....	826
9.52.4.1	Backward execution.....	826
9.52.4.2	Examples	826
9.52.5	Alarms	829
9.53	MATH FUNCTION INSTRUCTION.....	831
9.53.1	Type of Math functions	831
9.53.2	Instruction format of math function	832
9.53.2.1	Instruction format of assignment statements	832
9.53.2.2	Instruction format of relational statements	832
9.53.2.3	Instruction format of wait command statements.....	833
9.53.3	Function specification of Math functions.....	833
9.53.3.1	Square root (SQRT)	833
9.53.3.2	Trigonometric function (SIN)	833
9.53.3.3	Trigonometric function (COS)	834
9.53.3.4	Trigonometric function (TAN)	834
9.53.3.5	Inverse trigonometric function (ASIN)	834
9.53.3.6	Inverse trigonometric function (ACOS)	835
9.53.3.7	Inverse trigonometric function (ATAN2)	835
9.53.3.8	Inverse trigonometric function (ATAN)	835
9.53.3.9	Exponent.....	836
9.53.3.10	Natural Logarithm	836
9.53.3.11	Absolute (ABS)	837
9.53.3.12	Truncate (TRUNC)	837
9.53.3.13	Round off (ROUND)	837
9.53.4	Background operation of Math function	838
9.53.5	Teach Math function instruction	838
9.53.6	Restriction of teaching math function	840
9.53.7	Exceptions and Restriction.....	840
9.54	GROUP MASK EXCHANGE	841
10	PALLETIZING FUNCTION.....	845
10.1	PALLETIZING FUNCTION	845

10.2	PALLETIZING INSTRUCTIONS	847
10.3	TEACHING THE PALLETIZING FUNCTION	849
10.3.1	Selecting a Palletizing Instruction	849
10.3.2	Inputting Initial Data	850
10.3.3	Teaching a Stacking Pattern	856
10.3.4	Setting Path Pattern Conditions	862
10.3.5	Teaching a Path Pattern	865
10.3.6	Notes on Teaching the Palletizing Function	868
10.4	EXECUTING THE PALLETIZING FUNCTION	869
10.4.1	Palletizing Register	870
10.4.2	Controlling the Palletizing Function by a Palletizing Register	871
10.5	MODIFYING THE PALLETIZING FUNCTION	872
10.6	PALLETIZING FUNCTION WITH EXTENDED AXES	873
10.7	PALLETIZING ALL-POINT TEACHING	874
11	FANUC <i>i</i>Pendant	877
11.1	OVERVIEW	877
11.2	APPEARANCE AND OPERATIONS	878
11.2.1	Appearance and Switches	878
11.2.2	Key Switches	880
11.2.3	Touch Panel	881
11.2.4	Status Window	882
11.2.5	Splitting the Screen	883
11.2.6	Changing the Operation Target Screen	885
11.2.7	Internet Browser Screen	886
11.2.8	Screen Selection Menu and Screen Menus on the Edit Screen	889
11.2.9	Status Sub-window	890
11.2.9.1	Current position display	891
11.2.9.2	Operator panel status display	891
11.2.9.3	Safety signal status display	892
11.2.10	Color Display According to the Alarm Severity	892
11.2.11	Backlight automatic blanking	893
11.2.12	Operational Restrictions	894
11.2.13	USB Port (Soft option, New <i>i</i> Pendant only)	894
11.2.14	Operational improvement in 7DA5 or later series	895
11.2.14.1	ICON menu	896
11.2.14.2	ICON editor	900
11.2.14.3	Cursor positioning	903
11.2.14.4	List for CALL instruction edit	905
11.2.14.5	List for MACRO instruction edit	905
11.2.14.6	Current edit line display	906
11.2.14.7	Sub program display	906
11.2.14.8	Wide screen mode	907
11.2.14.9	Screen maximization	907
11.2.15	Operational improvement in 7DA7 or later series	908
11.3	SETTING UP <i>i</i> Pendant	910
11.4	OPERATION PANEL	920
11.4.1	Summary	920
11.4.2	Basic operation	921
11.4.3	User Authentication	923
11.4.4	Panel Setup	924
11.4.4.1	Operation panel setting screen	924
11.4.4.2	Item select screen	926

	11.4.4.3	Item setting	927
	11.4.4.4	Panel enable condition setting	929
	11.4.4.5	Finish setting.....	930
	11.4.4.6	Save and restore setting value.....	930
	11.4.4.7	Register panel	931
	11.4.5	Caution	934
11.5		RESTRICTIONS	934

APPENDIX

A		SCREEN/PROGRAM INSTRUCTIONS	937
A.1		LIST OF MENUS	937
A.2		TYPES OF SCREENS.....	943
A.3		LIST OF PROGRAM INSTRUCTIONS.....	962
A.4		PROGRAM INSTRUCTIONS	966
	A.4.1	Motion Instructions	966
	A.4.2	Additional Motion Instructions	966
	A.4.3	Register and I/O Instructions.....	967
	A.4.4	Conditional Branch Instructions.....	969
	A.4.5	Wait Instruction.....	970
	A.4.6	Unconditional Branch Instructions.....	970
	A.4.7	Program Control Instructions	971
	A.4.8	Other Instructions	971
	A.4.9	Skip and Offset Condition Instruction.....	972
	A.4.10	Frame Setup Instruction	972
	A.4.11	Macro Instruction	973
	A.4.12	Multiaxis Control Instructions.....	973
	A.4.13	Position Register Look-ahead Execution Instruction	973
	A.4.14	Soft Float Instruction.....	974
	A.4.15	Status Monitoring Instructions	974
	A.4.16	Motion Group Instructions	974
	A.4.17	Palletizing Instructions	975
B		SPECIAL OPERATION.....	976
B.1		START MODE	976
	B.1.1	Start Up Methods.....	976
	B.1.2	Initial Start.....	977
	B.1.3	Controlled Start	978
	B.1.4	Cold Start.....	980
	B.1.5	Hot Start	981
B.2		MASTERING	982
	B.2.1	Jig Mastering	984
	B.2.2	Mastering at the Zero-degree Positions.....	985
	B.2.3	Quick Mastering	987
	B.2.4	Single Axis Mastering.....	989
	B.2.5	Setting Mastering Data	992
B.3		SOFTWARE VERSION	994
B.4		ROBOT AXIS STATUS.....	996
B.5		DIAGNOSIS SCREEN	1001
	B.5.1	Outline	1001
	B.5.2	About Reducer Diagnosis.....	1001
	B.5.3	Procedure.....	1002
	B.5.4	Each Item.....	1003

B.6	WORLD FRAME ORIGIN	1005
B.7	FSSB LINE SETUP	1006
B.7.1	Definition of FSSB line	1006
B.7.2	Setting 1 (FSSB line)	1007
B.7.3	Setting 2 (Number of total axes on FSSB line 1)	1007
B.7.4	Setting 3 (Hardware start axis)	1007
B.7.5	Setup Examples	1008
B.7.5.1	Example 1	1008
B.7.5.2	Example 2	1009
B.7.5.3	Example 3	1009
B.8	POSITIONER SETUP	1010
B.9	EXTENDED AXIS SETUP	1016
B.10	INDEPENDENT ADDITIONAL AXIS SETUP	1021
C	SYSTEM VARIABLES	1025
C.1	FORMAT OF A SYSTEM VARIABLE TABLE	1025
C.2	SYSTEM VARIABLES	1027
D	SAVING RESEARCH DATA	1043
D.1	DIAGNOSTIC LOG	1045
D.2	MAINTENANCE DATA	1046
D.3	All BACKUP	1047
D.4	IMAGE BACKUP	1049

1 INTRODUCTION

This chapter explains the manual plan.

Contents of this chapter

1.1 MANUAL PLAN

1.1 MANUAL PLAN

About this manual

FANUC Robot series (R-30iA Mate CONTROLLER) Operator's Manual.

This manual describes how to operate the FANUC Robot, an all-purpose compact robot. It is controlled by the FANUC R-30iA Mate controller (called the robot controller hereinafter) containing the FANUC Robot software.

This manual describes the following items for manipulating workpieces with the robot:

- Setting the system for manipulating workpieces
- Operating the robot
- Creating and changing a program
- Executing a program
- Status indications
- Backup and restore robot programs.

Using this manual

Each chapter of the manual describes a single operation of the robot. The user can select and read chapters describing required operations.

Chapter 1 Introduction	Describes how to use this manual and the safety precautions that must be observed in working with the robot. All users must read the safety precautions.
Chapter 2 Overview	Gives a basic knowledge of the robot. It describes the basic configuration of the robot and the system for Robot.
Chapter 3 Setting the System for Robot	Describes the procedure for setting the system for Robot including input/output, coordinate system, and reference position.
Chapter 4 Program Structure	Describes the program structure and the syntax of program instructions.
Chapter 5 Creating a Program	Describes how to design, create, change, delete, and copy a program. It also describes the procedures for turning the power on and moving the robot by jog feed.
Chapter 6 Executing a Program	Describes how to execute and stop a program. It also describes the test operation, automatic operation, and recovery from the alarm state.
Chapter 7 Status Indications	Describes how to check the operating status of the robot, using the status indicator LEDs.
Chapter 8 File Input/Output	Describes how to store and read, a program file or system file.
Chapter 9 Utility	Describes additional utility functions, macro functions, program shift and mirror shift.
Chapter 10 Palletizing Function	Describes the setting and executing of palletizing function.
Chapter 11 FANUC iPendant	Describes the FANUC iPendant.
Appendix	Describes lists of the menus, screens, program instructions and detail of program, special operations, lists of the system variables, saving research data.

Identification

For editions and order files of software, read the following sections:

Item to be checked	Section
Edition of your software	B.3 Software Version
Order No. of your software	A.1 List of Menus

Specifications of products

For memory statuses or software option list, see the following sections:

Item to be checked	Section
Memory status	7.12 Memory Use Status Display
Software option list	A.1 List of Menus
Menu displayed when an option is selected	A.1 List of Menus
Program instruction that can be used when an option is selected	A.3 List of Program Instructions

Related manuals

The following manuals are available:

Robot controller	OPERATOR'S MANUAL (This manual)	Intended readers: Operators responsible for designing, introducing, operating, and adjusting the robot system at the work site. Topics: Functions, operations and the procedure for operating the robot. Programming procedure, interface and alarm. Use: Guide to teaching, introducing, and adjusting the robot at the work site, and application designing.
	OPERATOR'S MANUAL (Alarm code list)	Topics: Error code listings, causes, and remedies. Use: Installing and activating the system, connecting the mechanical unit to the peripheral device and maintenance the robot.
	MAINTENANCE MANUAL	Topics: Installing and activating the system, connecting the mechanical unit to the peripheral device and maintenance the robot.
Mechanical unit	OPERATOR'S MANUAL	Topics: Installing and activating the robot, connecting the mechanical unit to the controller, maintaining the robot. Use: Guide to installation, activation, connection, and maintenance.

Notation

This manual contains safety precautions against injury and property damage. Those precautions are labeled "Warning" or "Caution," according to the degree of importance. Supplementary explanation is given under "Note." Before starting to use a robot, carefully read the "Warning," "Caution," and "Note."



WARNING

Failure to follow the instruction given under "Warning" can cause fatal or serious injury to the user. This information is indicated in bold type in a box so that it can be easily distinguished from the main body of this manual.

**CAUTION**

Failure to follow the instruction given under "Caution" can cause injury to the user or property damage. This information is indicated in a box so that it can be easily distinguished from the main body of this manual.

NOTE

The information given under "Note" is a supplementary explanation, which is neither a warning nor a caution.

Carefully read and save this manual.

2 OVERVIEW

This chapter shows the basic configuration of the FANUC Robot System and briefly describes the functions of each component.

Contents of this chapter

- 2.1 APPLICATION TOOL SOFTWARE
- 2.2 ROBOT
- 2.3 CONTROLLER

A FANUC robot system consists of the tool software for manipulating workpieces, the mechanical unit of the robot itself (FANUC Robot series), and the Robot controller.

The FANUC robot offers outstanding performance when handling or welding.

Tool software for application

The tool software for application is a software package for all kinds of Robot's manipulations installed on the Robot controller. Any work can be performed by specifying menus and instructions from the teach pendant. The tool software for manipulating workpieces contains instructions for controlling the robot, hands, remote controllers, and other peripheral devices.

An additional axis or the I/O between controller and another peripheral device can be controlled. Other peripheral devices include cell controllers or sensors.

Robot

Robot has a hand or another end effector interface for control to do work. The FANUC robot is ideal for manipulating workpieces.

Controller

The Robot controller supplies power to drive the mechanical unit.

The tool software for manipulating workpieces is installed on the Robot controller to control the teach pendant, operator's panel, and external peripheral devices.

Peripheral devices, including remote controllers, are required to configure a system for manipulating workpieces.

- The remote controllers are used to control the Robot controller.
- The hands, sensors, and other devices are operated using I/O and serial communication units.

Fig. 2 shows a typical robot system for manipulating workpieces. The system consists of a robot, the Robot controller, and peripheral devices.

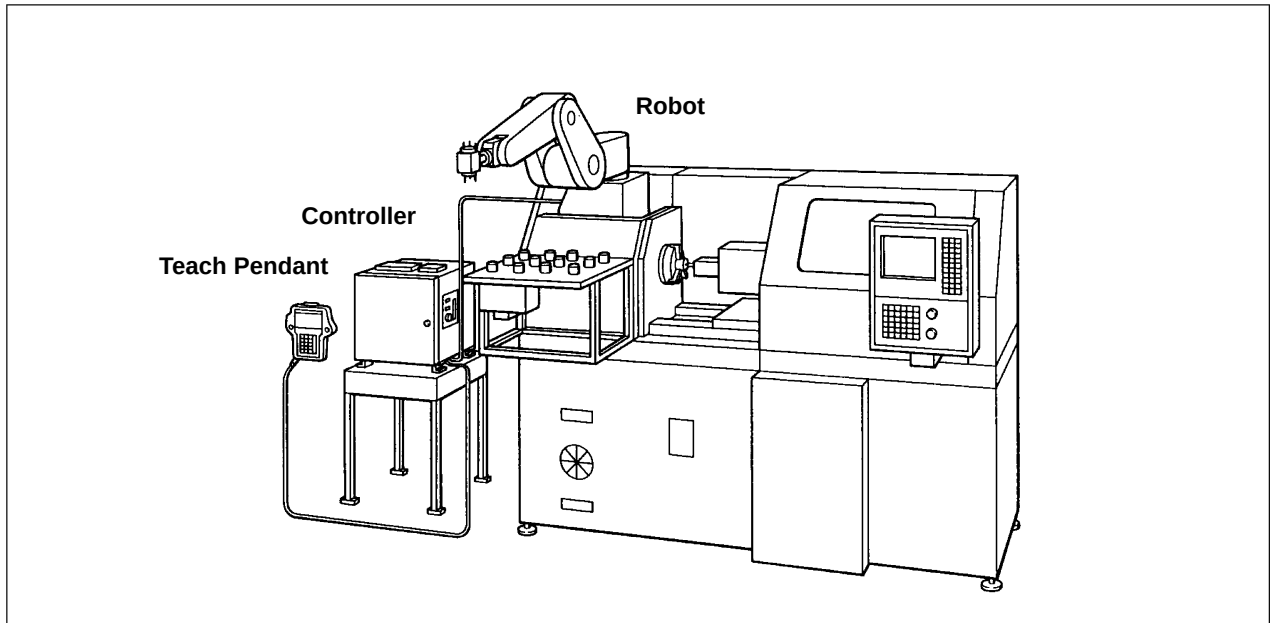


Fig. 2 Handling system

2.1 APPLICATION TOOL SOFTWARE

The Application tool software has been specially designed to perform manipulating workpieces operations. The Application tool software is contained in the Robot and enables the following:

- Setting up the system for Robot applications
- Creating a program
- Performing the test operation of a program
- Performing the automatic operation
- Status display or monitoring

When optional functions are incorporated, the system can be expanded and the management functions can be enhanced.

2.1.1 System Setting

The Application tool software has an interface for specifying parameters of operation of the manipulation system. (For how to set the Robot system, see Chapter 3.)

With the Application tool software, the Handling remote controller, and other external units can be controlled. Before the manipulation is started, the following must be specified: input from and output to the hand and other peripheral units, the coordinate system, communication, and automatic operation.

2.1.2 Jog Feed of the Robot

Jog feed of the robot is the operation of moving the robot as desired by manually entering commands on the teach pendant. When a motion instruction of a program is taught, the robot is moved to the target position by jog feed, then the position is recorded. (For the jog feed of the robot, see Subsection 5.2.3.)

2.1.3 Program

A program contains motion instructions, input/output instructions, register instructions, and branch instructions. (For the program structure, see Chapter 4.) Each instruction is assigned a statement number. The target work is accomplished by sequentially executing the instructions.

The teach pendant is used to create or correct a program. (For creation of a program, see Chapter 5.) The program contains the following instructions. Fig. 2.1.3 shows a basic program for manipulating workpieces.

- **Motion instruction:**
Moves the tool to the target position within the operating range.
- **Additional motion instruction:**
Performs an additional (special) operation during a motion.
- **Register instruction:**
Places (loads) numerical data into a register.
- **Position register instruction:**
Places (loads) position data into a register.
- **Input/output instruction:**
Sends or receives a signal to or from a peripheral unit.
- **Branch instruction:** Changes the flow of a program.
- **Wait instruction:**
Holds execution of the program until the specified conditions are satisfied.
- **Routine call instruction:** Calls and executes a subprogram.
- **Macro instruction:**
Calls a specified program and executes it.
- **Palletizing instruction:** Palletizes workpieces.
- **Program end instruction:** Terminates execution of a program.
- **Comment instruction:** Adds a comment to a program.
- **Other instructions**

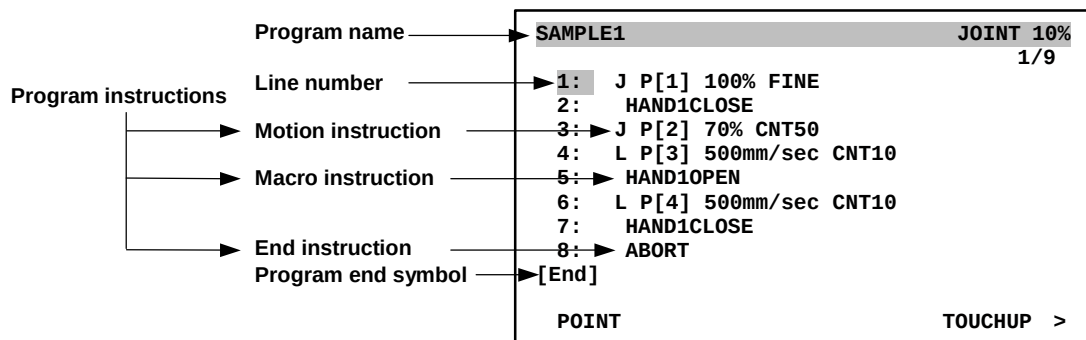


Fig. 2.1.3 Robot Program

2.1.4 Test Operation (Test Execution)

After the system is set and a program is created, perform the test operation in the test execution mode to check the program for normal operation. (For the test operation, see Section 6.3.)

The test execution of the program is one of the important steps in creating a good program. Before starting automatic operation, execute the test program.

2.1.5 Automatic Operation (Operation Execution)

Automatic operation (operation execution) is the final step in executing programs. In automatic operation, the following processing is executed:

- Specified programs are started one after another. (For automatic operation, see Sections 3.8 and 6.6.)
- During automatic operation, position data can be corrected (online position correction Section 6.7).
- The processing is halted, then aborted or resumed. (For halting a program, see Section 6.1.)

2.2 ROBOT

A robot is a mechanical unit consisting of axes and arms driven by servo motors. A place at which arms are connected is a joint, or an axis.

J1, J2, and J3 are main axes. The basic configuration of the robot depends on whether each main axis functions as a linear axis or rotation axis.

The wrist axes are used to move an end effector (tool) mounted on the wrist flange. The wrist itself can be rotated about one wrist axis and the end effector rotated about the other wrist axis.

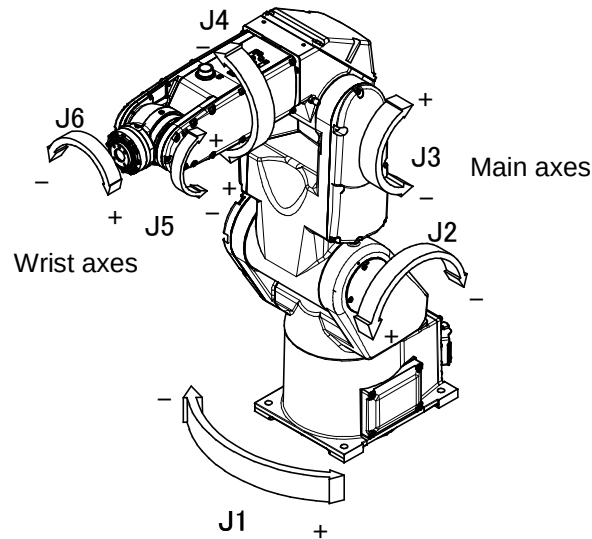


Fig. 2.2 (a) Main axes and wrist axes

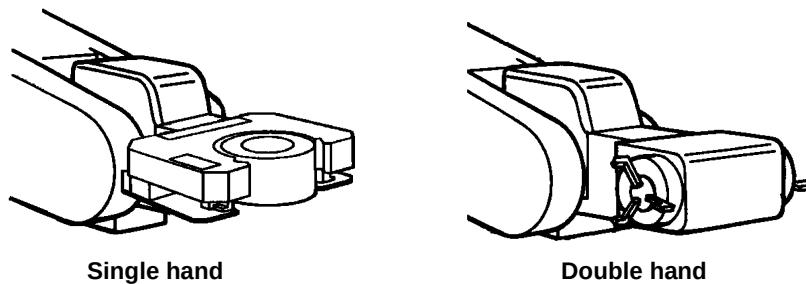


Fig. 2.2 (b) Hand with Fingers

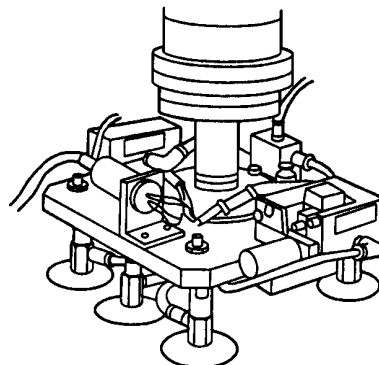


Fig. 2.2 (c) Hand with Suction Cups and No Fingers

2.3 CONTROLLER

Robot controller includes a power unit, user interface circuit, motion controlling circuit, memory circuit, and input/output circuit.

There are switches and connectors on the operator's panel of the robot controller.

- A circuit protector (for turning on or off the power) and emergency stop button are provided.
- A memory card port is provided.

The user should use a teach pendant to operate the controller.

The operation control circuit controls the servo amplifier which moves all the robot axes, including any additional axes, via the main CPU printed circuit board.

The memory circuit can store programs and data set by the user in the C-MOS RAM on the main CPU printed circuit board.

The input/output (I/O) circuit interfaces the controller with the peripheral units by receiving and sending signals via the I/O link cable and peripheral connecting cable. The remote input/output signal is used for communication with the remote controller.

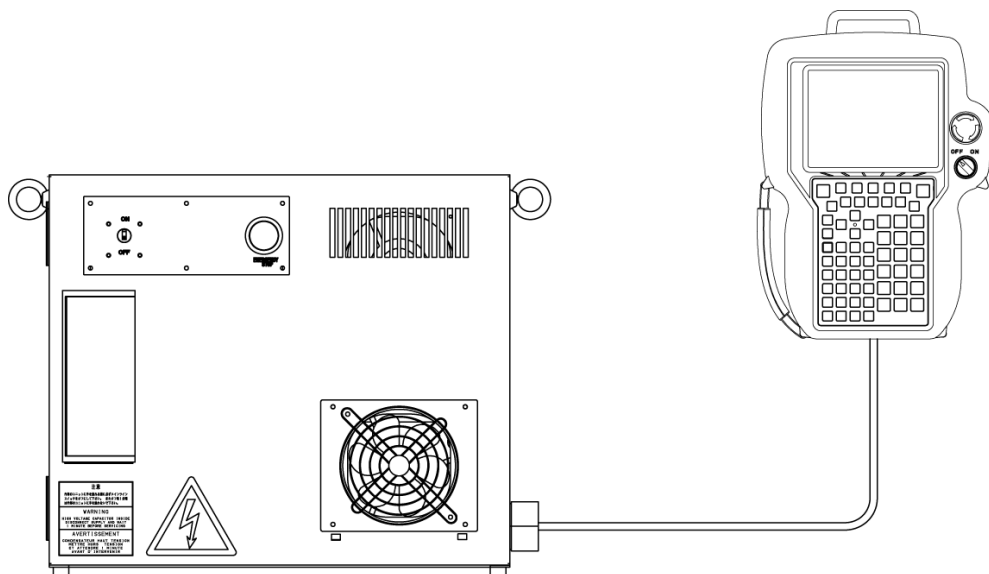


Fig. 2.3 (a) Robot controller (Standard)

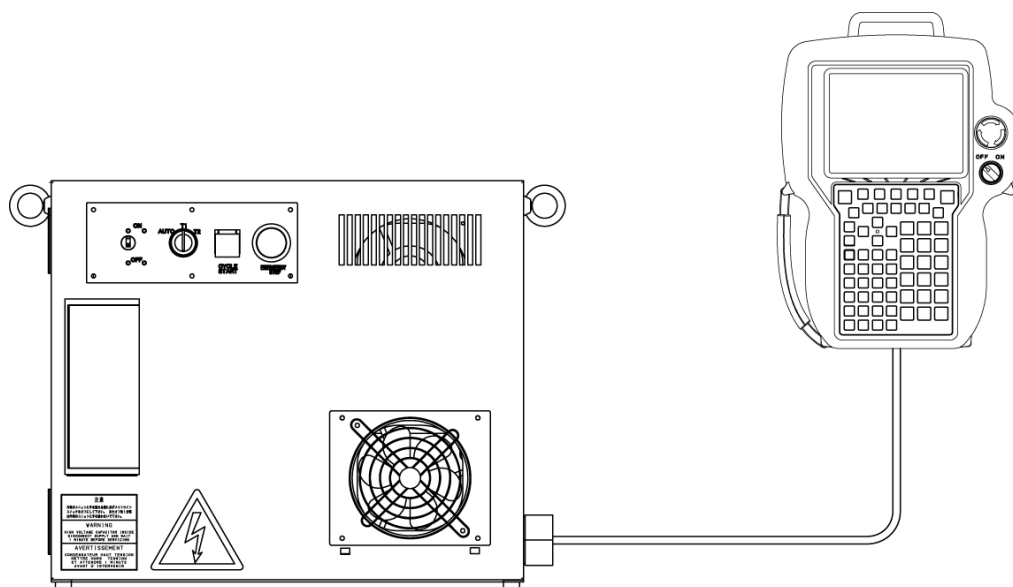


Fig. 2.3 (b) Robot controller (RIA specifications)

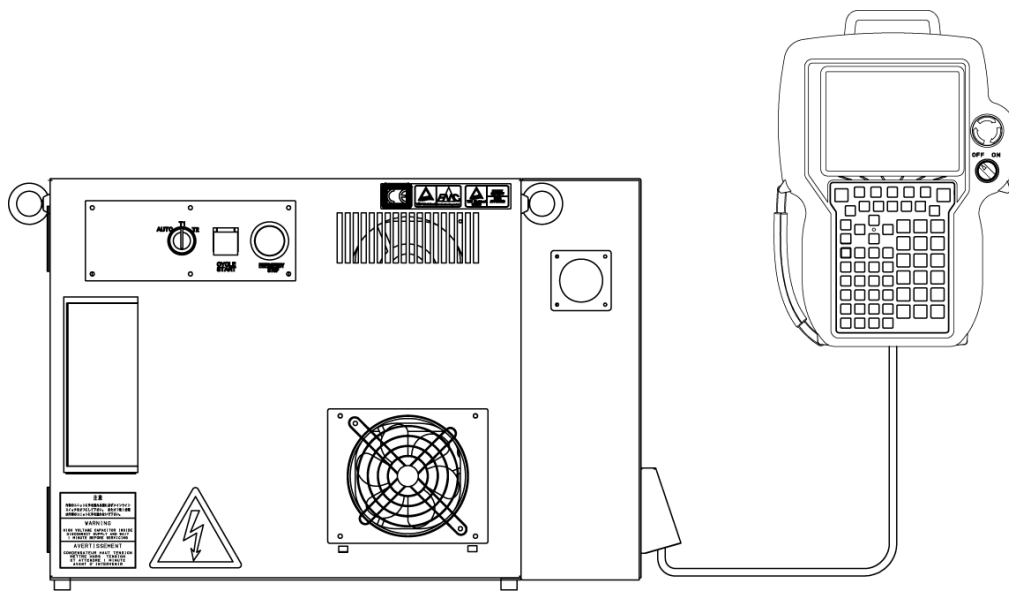


Fig. 2.3 (c) Robot controller(CE specifications)

The circuitry of the controller depends on the robot and the system it controls. For details, refer to the maintenance manual.

2.3.1 Teach Pendant

The teach pendant provides an interface between the Application tool software and the operator. The teach pendant is connected to the printed circuit board in the controller by a cable.

The following operations can be performed using the teach pendant:

- Jog feed of the robot
- Program generation
- Test execution
- Actual work
- Status check

The teach pendant includes the following:

- Liquid crystal display of 40 characters by 16 lines
- 11 LEDs
- 61 keys (Four keys of them are for exclusive use of each application)



CAUTION

The operator of the teach pendant should use gloves that would not cause any operation error.

The following switches are also provided:

Teach pendant enable switch	This switch enables or disables the teach pendant. When the teach pendant is disabled, a jog feed, program generation, or test execution cannot be carried out.
Deadman switch	DEADMAN SWITCH is used as an enabling device. When the teach pendant is enabled, this switch allows robot motion only while the deadman switch is gripped. If you release this switch, the robot stops immediately.
Emergency stop button	When pressed, the emergency stop button immediately stops the robot (Please refer to "STOP TYPE OF ROBOT" in SAFETY for detail of stop type).

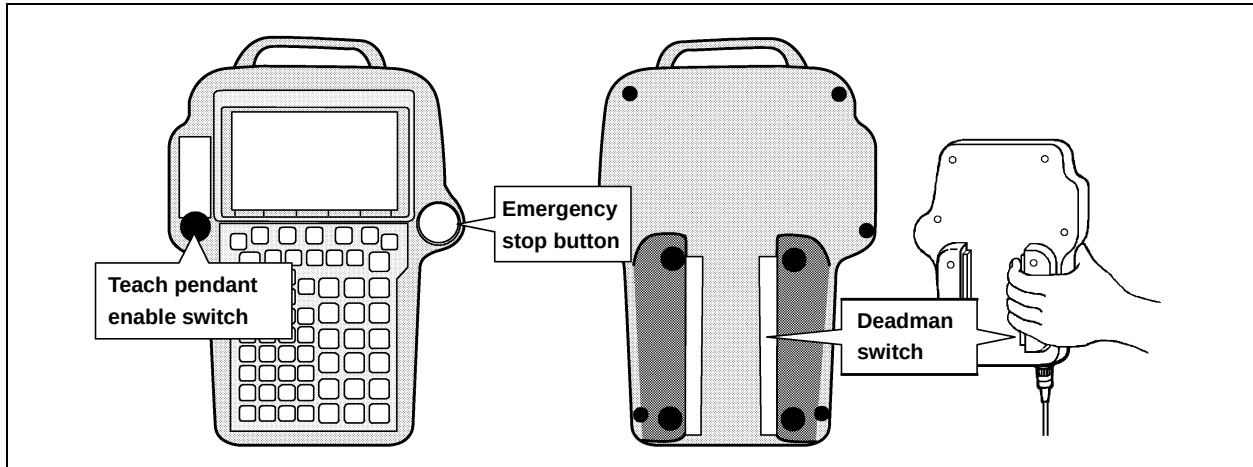


Fig. 2.3.1 (a) Switches on the Teach Pendant

Fig. 2.3.1 (b) shows the teach pendant.

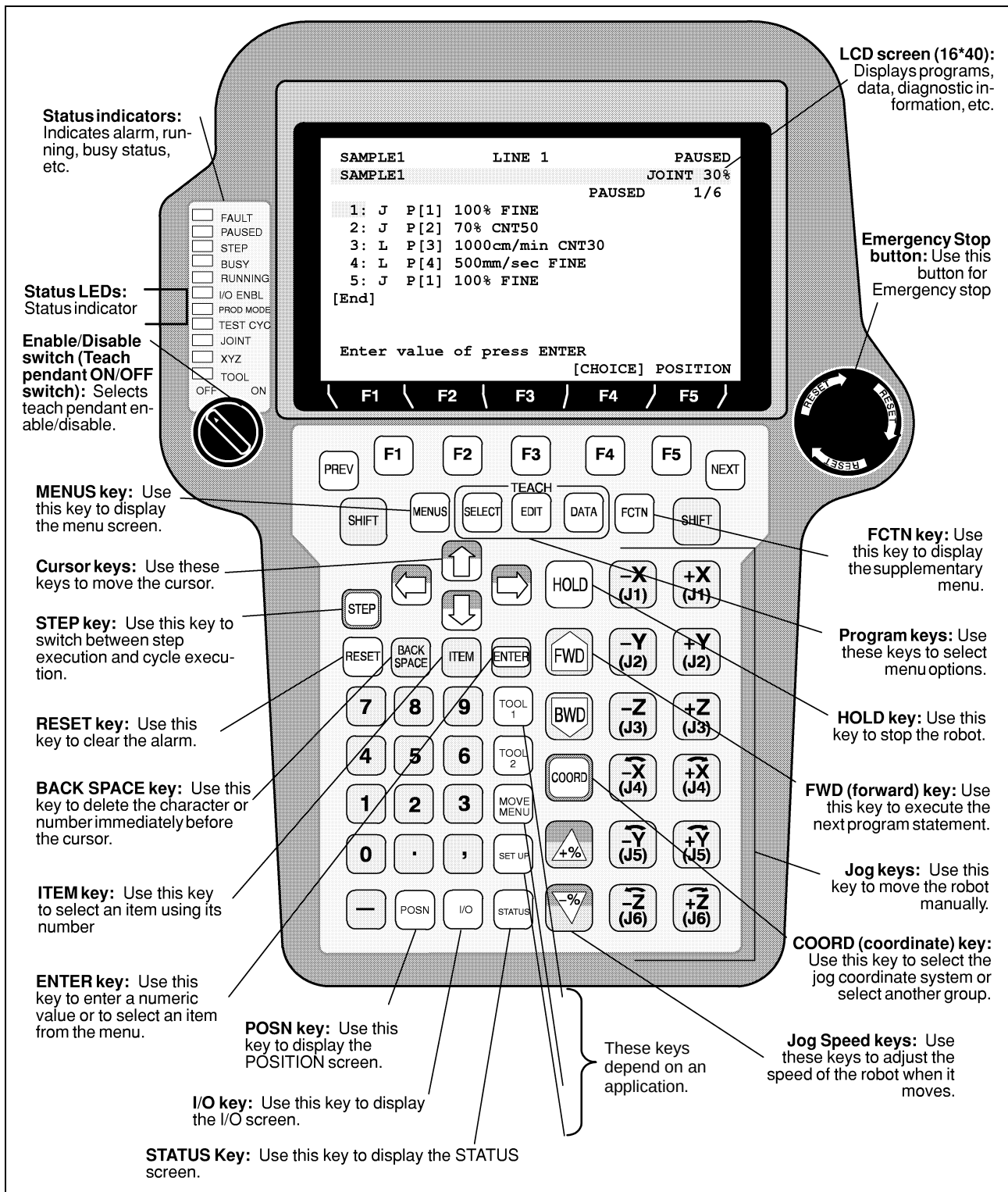


Fig. 2.3.1 (b) Teach Pendant

Keys on the teach pendant

The teach pendant has the following keys:

- Keys related to menus
- Keys related to jog feed
- Keys related to execution
- Keys related to editing

Table 2.3.1 (a) Keys related to menus

Key	Function
    	The function (F) key to select a function key menu at the bottom line of the screen.
	The next page key to switch the function key menu on the next page.
 	The MENUS key to display the screen menu. The FCTN key to display the function menu.
  	The SELECT key to display the program selection screen. The EDIT key to display the program edit screen. The DATA key to display the data screen.
 	The MOVE MENU key moves the robot to the reference position. Create a program which moves the robot to the reference position and assign this program to a macro instruction so that this can be started by this MOVE MENU key. The SET UP key displays the setup screen.
  	The STATUS key displays the status screen. The I/O key displays the I/O screen. The POSN key displays the current position screen.

Each of the keys TOOL 1, TOOL 2, and MOVE MENU is an application-dedicated key on the teach pendant for handling tools.

Application-dedicated keys differ depending on the application.

Table 2.3.1 (b) Keys related to jog feed

Key	Function
	The SHIFT key is used to execute a jog feed of the robot, teach the position data, and start a program. The right and left Shift keys have the same function.
     	The jog keys are effective while a Shift key is held down. They are used for jog feed.
     	

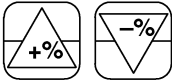
Key	Function
	The COORD key selects a manual-feed coordinate system (jog type). Each time the COORD key is pressed, it selects the next jog type in the order: JOINT, JGFRM, World frame, TOOL, USER. When this key is pressed while a Shift key is held down, a jog menu for changing the coordinate system appears.
	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%, 5%, 50%, 100%. (changing amount 1% for 5% or less and changing amount 5% for 5% or more.)

Table 2.3.1 (c) Keys related to execution

Key	Function
 	The FWD key or BWD key (+ SHIFT key) starts a program. When the shift key is released during the program execution, the program halts.
	The HOLD key causes a program to halt.
	The STEP key selects step or continuous test operation.

Table 2.3.1 (d) Keys related to editing

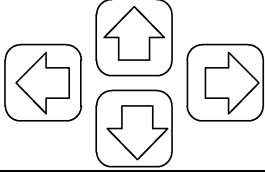
Key	Function
	The PREV key restores the most recent state. In some cases, the screen may not return to the immediately preceding status.
	The ENTER key enters a numeral or selects a menu.
	The BACK SPACE key deletes the character or numeral immediately before the cursor.
	The cursor key moves the cursor. The cursor is the highlighted part which can move on the teach pendant screen. This part becomes the object of operation (input or change of the value or contents) from the teach pendant key.
	The ITEM key moves the cursor to a line whose number is specified.

Table 2.3.1 (e) Keys related to handling

Key	Function
 	TOOL 1 or 2 key displays the tool 1 or 2 screen.

LEDs on the teach pendant

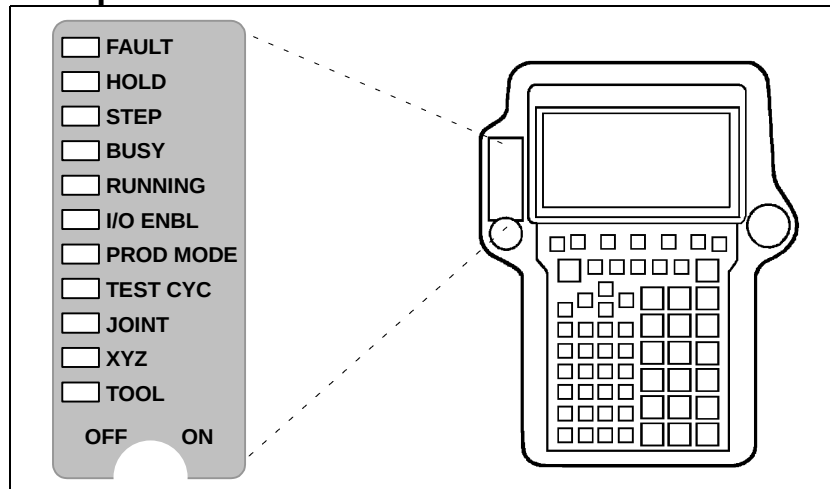


Fig. 2.3.1 (c) LEDs on the teach pendant

Table 2.3.1 (f) LEDs on the teach pendant

LED	Function
FAULT	The FAULT LED indicates that an alarm has occurred.
HOLD	The HOLD LED indicates that the HOLD button is being pressed or HOLD signal is being input.
STEP	The STEP LED indicates that it is under step operation mode.
BUSY	The BUSY LED is lit while the robot is working. It is also lit when a program is executed.
RUNNING	The RUNNING LED indicates that the program is being executed.
JOINT	The JOINT LED is lit when joint jog is selected as the manual-feed coordinate system (jog type).
XYZ	The XYZ LED is lit when Cartesian jog (JGFRM World frame or USER) is selected as the manual-feed coordinate system (jog type).
TOOL	The TOOL LED is lit when tool jog (TOOL) is selected as the manual-feed coordinate system (jog type).
I/O ENBL	This LED indicates that I/O is enabled.
PROD MODE	This LED indicates that the system is in production mode and that the program will start when an automatic operation start signal (production start) is received.
TEST CYC	This LED indicates that the remote/local setting is set to local and that the program is under test execution.

Each of the LEDs I/O ENBL, PROD MODE, and TEST CYC is an application-dedicated LED on the teach pendant for handling tools. Application-dedicated LEDs differ depending on the application tool.

2.3.1.1 Display Screen of the Teach Pendant

The liquid crystal display screen (liquid crystal display) displays various application tool software screens shown in Fig.2.3.1.1 (a). To operate the robot, select a screen corresponding to a desired function. The screen is selected by the screen menus shown in Fig.2.3.1.1 (b).

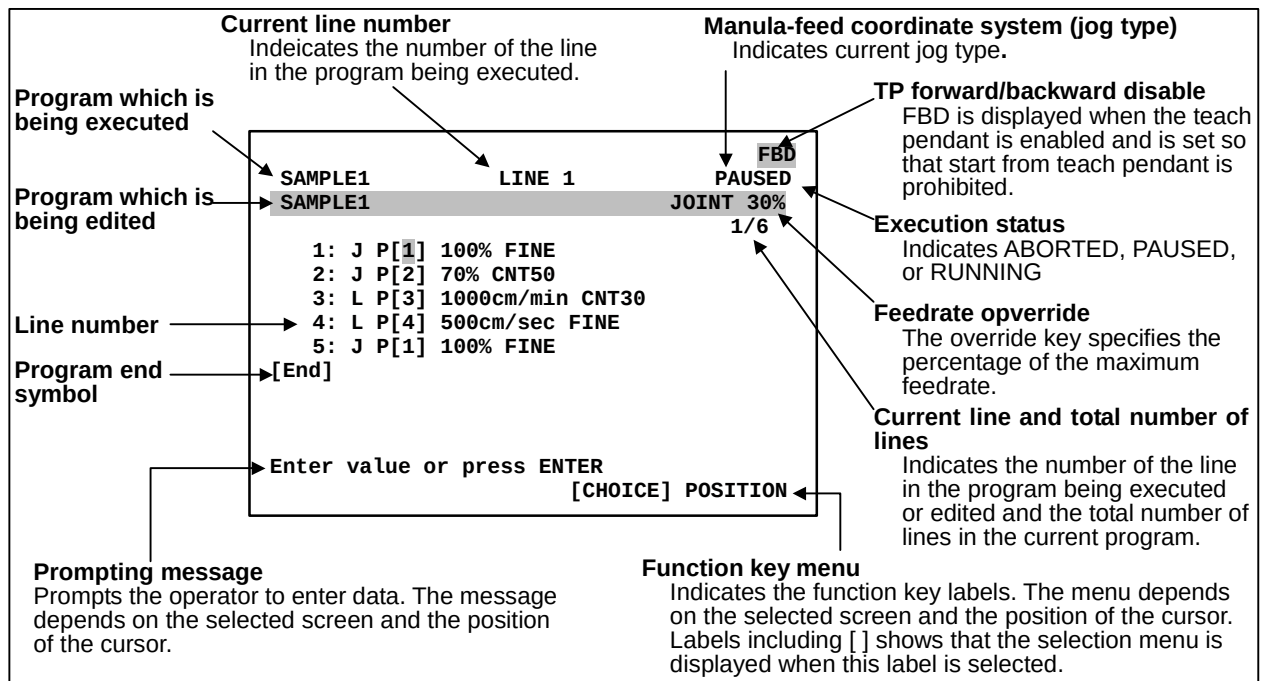


Fig. 2.3.1.1 (a) Program Edit Screen

Screen menu and function menu

Menus are used to operate the teach pendant. The screen menu is selected by the MENUS key and the function menu is selected by the FCTN key.

Fig. 2.3.1.1 (b), Fig. 2.3.1.1 (c), and Fig. 2.3.1.1 (d) show the screen menu, function menu, and quick menu respectively.

- Screen menu

The screen menu is used to select a screen. The screen menu lists the following options. (For the list of menus, see Appendix A.1. For the screen type, see Appendix A.2.) To display the screen menu, press the MENUS key on the teach pendant.

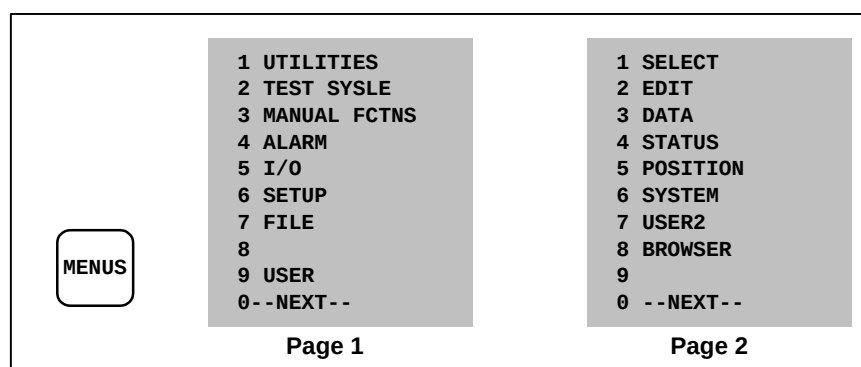


Fig. 2.3.1.1 (b) Screen menu

Table 2.3.1.1 (a) Screen menu

LED	Function
UTILITIES	The utility screen is used to execute various functions.
TEST CYCLE	The test cycle screen is used to specify the data for test operation.
MANUAL FCTNS	The manual operation screen is used to execute macro instructions.
ALARM	The alarm history screen shows the history and details of alarms.
I/O	The I/O screen is used to display and set manual output, simulated input/output, and assign of signals.

LED	Function
SETUP	The setting screen is used to set the system.
FILE	The file screen is used to read or store files.
USER	The user screen shows user messages.
SELECT	The program selection screen is used to list or create the programs.
EDIT	The program edit screen is used to correct and execute a program.
DATA	The data screen shows the values in registers, position registers, and pallet register.
STATUS	The status screen shows the system status.
POSITION	The current position screen shows the current position of the robot.
SYSTEM	The system screen is used to set system variables and mastering.
USER2	This screen displays messages output from KAREL programs.
BROWSER	This screen is used to brows Web pages on the network. (Displayed only when the <i>i</i> Pendant is used as the teach pendant.)

- Function menu

The function menu is used to execute a miscellaneous function. (For the list of menus, see Appendix A.1.)
To display the function menu, press the FCTN key on the teach pendant.

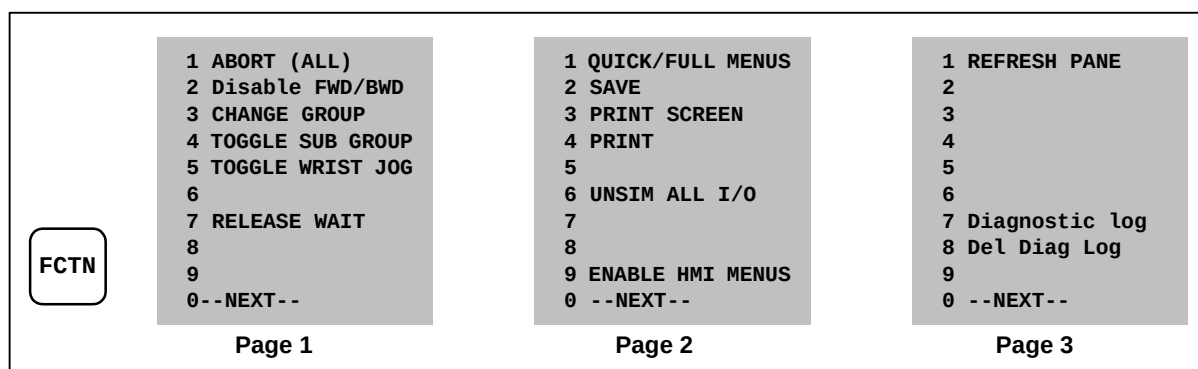


Fig. 2.3.1.1 (c) Function menu

Table 2.3.1.1 (b) Function menu

LED	Function
ABORT (ALL)	ABORT forces a program which is being executed or temporarily halted to terminate.
Disable FWD/BWD	Disable FWD/BWD enables or disables starting a program with a teach pendant
CHANGE GROUP	Changes the operation group for jog feed. Displayed only when multiple groups are set.
TOGGLE SUB GROUP	TOGGLE SUB GROUP toggles jog between robot standard axes and extended axes.
TOGGLE WRIST JOG	TOGGLE WRIST JOG toggles jog between the attitude control feed and the wrist joint feed which does not maintain the wrist attitude in linear feed.
RELEASE WAIT	Skips the wait instruction currently being executed. When the wait state is released, execution of the program stops temporarily at the line subsequent to the wait instruction.
QUICK/FULL MENUS	QUICK/FULL MENUS toggles the menu between a usual screen menu and a quick menu.
SAVE	SAVE saves the data related to the current screen on external memory device.
PRINT SCREEN	PRINT SCREEN prints the data displayed on the current screen.
PRINT	PRINT prints the data on the current screen exactly.
UNSIM ALL I/O	Cancels the simulated settings of all IO signals.
ENABLE HMI MENUS	Used to select whether to display the HMI menu when the MENUS key is pressed. (This is displayed when the <i>i</i> Pendant is used as the teach pendant.)

LED	Function
REFRESH PANE	This item refreshes the pane in the active window. (This is displayed when the <i>iPendant</i> is used as the teach pendant.)
Diagnostic log	This item saves the log data to external device when problem occur on controller. This save operation should do before controller power off.
Del Diag Log	This item deletes the saved diagnostic log data.

Quick menu

When a quick menu is selected in QUICK/FULL MENUS of FUNCTIONS, the screen that can be displayed by using the screen menu is limited to the following:

- ALARM / alarm occurrence and alarm history screen
- UTILITIES / hint screen
- TEST CYCLE screen
- DATA / register screen
- Tool 1 / Tool 2 screen
- STATUS/axis screen
- I/O/digital/group/Robot I/O screen
- POSITION screen
- SETUP / Coordinate system setup screen
- USER / USER 2 screen
- SETUP PASSWORD screen
- BROWSER screen (only when *iPendant* is used)

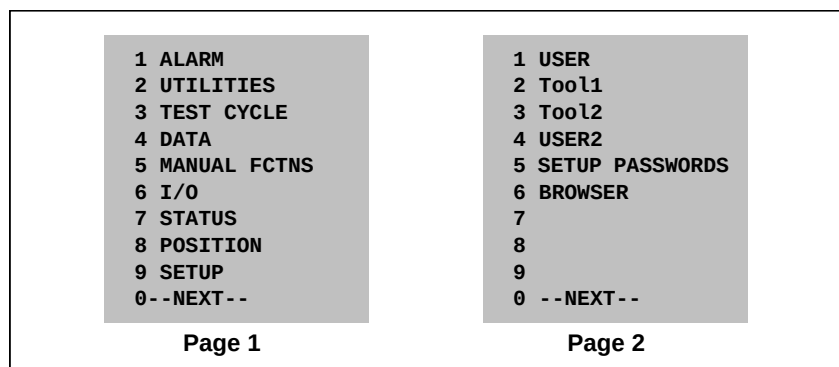


Fig. 2.3.1.1 (d) Quick menu

NOTE

- 1 The program selection screen can be displayed by the SELECT key. But the only available function is selecting a program.
- 2 The program edit screen can be displayed by the EDIT key. But the only available functions are changing position and speed values.

2.3.2 Operator Panel

The operator panel has buttons, switches, and connectors. Fig. 2.3.2(a) ~ (c) shows the operator panel on the cabinet.

The buttons on the operator panel can be used to start a program and perform other operations.



CAUTION

Do not wear gloves which would likely cause operator errors when using the operator panel.

Table 2.3.2 lists the switches on the operator's panel. The standard operator panel of the robot controller does not have the power ON/OFF button. Execute the power on or off with the breaker of the controller.

Table 2.3.2 Switches on the Operator Panel

Switch	Function
Emergency stop button	Press this button to stop the robot immediately (Please refer to "STOP TYPE OF ROBOT" in SAFETY for detail of stop type). Turn the emergency stop button clockwise to release it.
Start button	Starts the currently selected program. Lit while the program is being started.
Three mode switch	Enables the user to select operation mode suitable to the robot operation conditions or the status of its use.

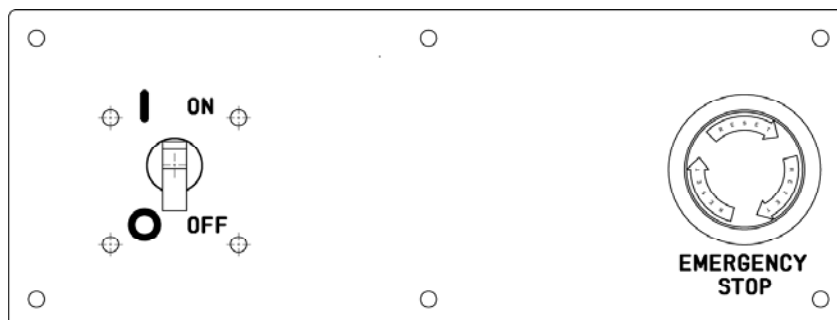


Fig. 2.3.2 (a) Operator Panel (Standard)

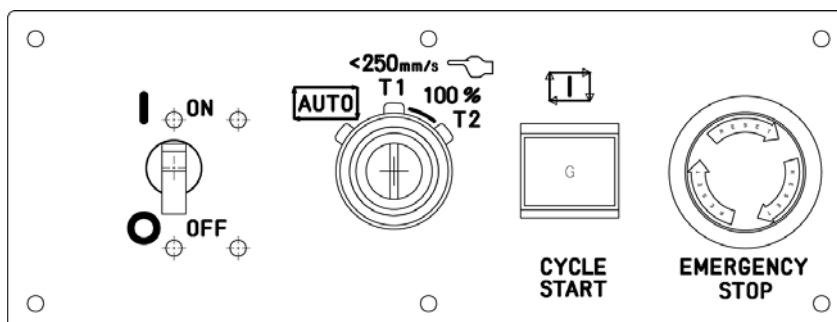


Fig. 2.3.2 (b) Operator Panel (RIA specifications)

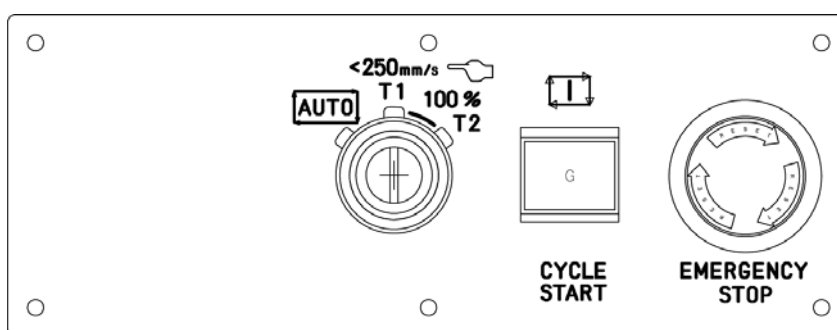


Fig. 2.3.2 (c) Operator Panel (CE specifications)

2.3.3 Remote Controller

Remote controllers are external devices connected to the Robot controller to configure a system. These are controllers for controlling the operation of the system created by the user using peripheral devices and I/O provided by the robot controller.

2.3.4 CRT/KB

The CRT/KB is an optional operation unit. An external CRT/KB is connected to the controller via an RS-232-C cable.

The CRT/KB can be used to execute almost all teach pendant functions excluding those related to robot operation. Functions related to robot operation can only be executed using the teach pendant.

2.3.5 Communication

For communications, the following interfaces are provided (communication ports Section 8.2).

- RS-232-C ports (internal 1)
- RS-422 ports (internal 1)

2.3.6 Input/Output

General-purpose and specialized input/output (I/O) signals are used to send the data of an external unit to the application tool software. The general-purpose signal (user-defined signal) is controlled by a program and is used to send or receive data to or from the external units or hand. The specialized signal (system-defined signal) is applied to a specific use.

The input/output signals include the following:

- Peripheral I/O (See Section 3.3.)
- Operator's panel I/O (See Section 3.4.)
- Robot I/O (See Section 3.2.)
- Digital I/O (See Subsection 3.1.1.)
- Group I/O (See Subsection 3.1.2.)
- Analog I/O (See Subsection 3.1.3.)

The number of the I/O signals and their types depend on the hardware of the controller and the number of selected I/O modules and their types.

2.3.7 Peripheral I/O

Peripheral I/O is a signal specialized for sending and receiving data to or from the remote controller or peripheral equipment. (See Section 3.3, "PERIPHERAL I/O").

Peripheral I/O signals perform the following:

- Select a program
- Start and stop a program
- Recover the system from the alarm state
- Others

2.3.8 Motion of the Robot

A single motion instruction specifies a motion of the robot, or a movement of the tool center point (TCP) from the current position to the target position. The Robot uses a motion control system that comprehensively controls the tool path, acceleration/deceleration, positioning, feedrate, and other factors. The Robot controller can control up to 56 axes, divided into up to eight operation groups, up to four robot arms (multiple motion function). The controller can control up to nine axes for a group. The operation groups are independent of one another, but can be synchronized to operate the robot simultaneously.

The robot moves according to a jog feed specified on the teach pendant or a motion instruction specified in a program.

To execute a jog feed of the robot, use the corresponding key on the teach pendant. In jog feed, the motion of the robot depends on the selected manual-feed coordinate system (jog type) and feedrate override. When

a motion instruction is used, the motion of the robot depends on the position data, motion format, positioning path, traveling speed, and feedrate override specified in the instruction.

One of three motion formats – Linear, Circular, and Joint – can be selected to operate the robot. When Joint is selected, the tool is moved arbitrarily between two specified points. When Linear is selected, the tool is moved along a straight line between the two specified points. When Circular is selected, the tool is moved along an arc connecting three specified points.

A positioning path can be selected from two options, Fine and Cnt.

2.3.9 Emergency Stop Devices

This robot has following emergency stop devices.

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

When an emergency stop button is pushed or the external emergency stop is input, the robot stops immediately in any cases (Please refer to "STOP TYPE OF ROBOT" in SAFETY for detail of stop type). The signal terminal of the external emergency stop is on the controller and operator's box inside.

2.3.10 Extended Axis

A maximum of three axes of one group can be added to the standard axes (usually six axes) of the robot. The Robot can control up to 56 axes (with an optional servo card). The extended axis has the following two types:

- Extended axes
This can be controlled regardless of the robot motion and can move only at the joint motion.
- Integrated axes
Controlled together with the robot during linear or circular robot operation. Use these axes to perform linear or circular robot operation.

3 SETTING UP THE ROBOT SYSTEM

The Robot system can be used after required data is specified. This chapter describes the data that can be specified.

Contents of this chapter

- 3.1 I/O
- 3.2 ROBOT I/O
- 3.3 PERIPHERAL I/O
- 3.4 OPERATOR'S PANEL I/O
- 3.5 I/O Link SCREEN
- 3.6 I/O CONNECTION FUNCTION
- 3.7 SIMULATED INPUT SKIP FUNCTION
- 3.8 SETTING AUTOMATIC OPERATION
- 3.9 SETTING COORDINATE SYSTEMS
- 3.10 SETTING A REFERENCE POSITION
- 3.11 JOINT OPERATING AREA
- 3.12 USER ALARM
- 3.13 VARIABLE AXIS AREAS
- 3.14 INTERFERENCE PREVENTION AREA FUNCTION
- 3.15 SYSTEM CONFIG MENU
- 3.16 SETTING THE GENERAL ITEMS
- 3.17 OTHER SETTINGS

3.1 I/O

Input/output signals (I/O) are electric signals that allow the controller to communicate with the robot, end effector, external equipment, and other peripheral equipment of the system. The signals are divided into two groups: general-purpose I/O and specialized I/O.

General-purpose I/O

The user can define the general-purpose I/O as required.

This group includes the following signals:

- Digital I/O: DI[i]/DO[i]
- Group I/O: GI[i]/GO[i]
- Analog I/O: AI[i]/AO[i]

[i] represents the logic number of each I/O signal and group signal.

Specialized I/O

The use of the specialized I/O has already been defined. This group includes the following signals:

- Peripheral (UOP) I/O: UI[i]/UO[i]
- Operator's panel (SOP) I/O: SI[i]/SO[i]
- Robot I/O: RI[i]/RO[i]

[i] represents the logic number of each I/O signal and group signal.

- For Digital, Group, Analog, and Peripheral I/O, the logic ports can be mapped to the physical ports. They can be redefined.
- The physical numbers of the robot I/O are always the same as the logic numbers. They cannot be redefined.

Configuring I/O

An I/O module consists of the following hardware components. For details, refer to the "Maintenance Manual".

Rack

The rack indicates the kind of hardware which composes I/O module.

- 0 = Process I/O PC board
- 1 to 16 = I/O Unit-MODEL A / B
- 32 = I/O Link slave
- 48 = Peripheral device interface (CRMA15, CRMA16)

SLOT

The slot indicates numbers of I/O module parts which compose the rack.

- When the process I/O printed circuit board is used, the first connected board is SLOT 1, the second is SLOT 2 and others are numbered sequentially as this.
- When the I/O Unit-MODEL A or B is used, SLOT is the number identifying the connected module.
- For I/O link slaves or peripheral device interfaces (CRMA15 and CRMA16), the number is always 1.

Peripheral Device Control Interfaces (CRMA15 and CRMA16)

The R-30iA Mate controller is equipped with peripheral device control interfaces, which have 28 input points and 24 output points in total. By default, the signals of the peripheral device control interfaces are allocated to DI[101-120], DO[101-120], DI[81-88], and DO[81-84]. In the LR handling tool, the peripheral device control interfaces are allocated to peripheral I/O signals.

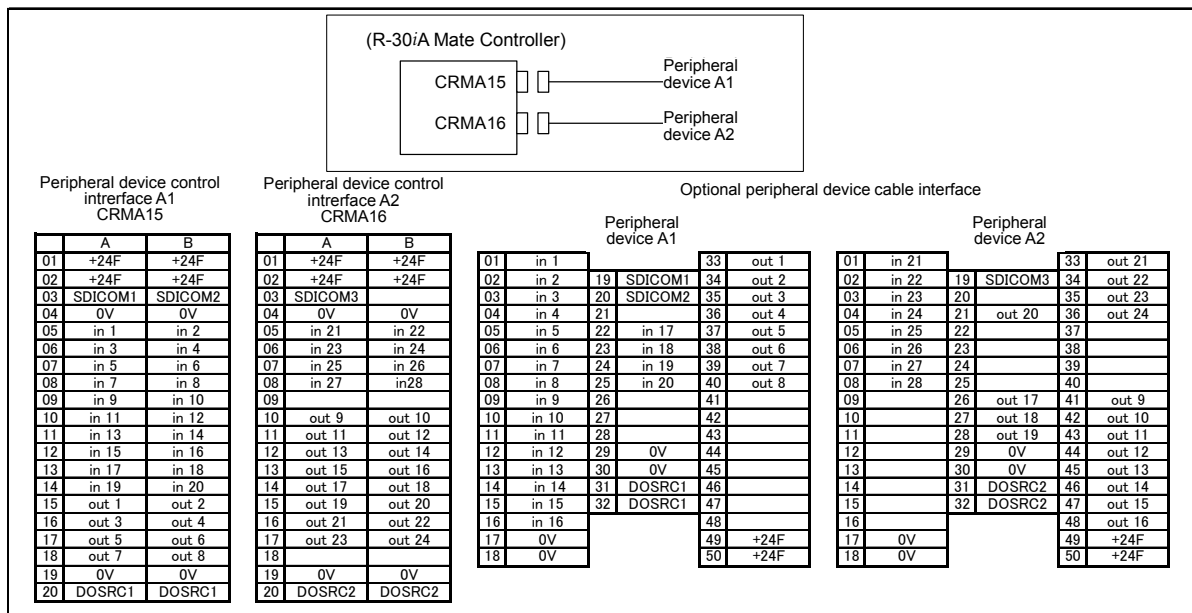


Fig. 3.1 (a) Peripheral device control interfaces

I/O link (master/slave)

The R-30iA Mate controller has two modes: I/O link master mode and I/O link slave mode.

I/O link slave mode

The robot controller operates as an I/O link slave device and connects to an I/O link master device such as the CNC.

- Connector for master side : M-JD1A/S-JD1B
- Connector for slave side : S-JD1A

I/O link master mode

The robot controller operates as an I/O link master device and connects to an I/O link slave device.

- Connector : M-JD1A/ S-JD1B

In default setting, the LR tool is set to the slave mode and the LR handling tool is set to the master mode.

To switch between the I/O link modes, change system variable \$IOMASTER and turn the power of the robot controller off and back on again.

\$IOMASTER=0: Slave mode

\$IOMASTER=1: Master mode

When UOP automatic allocation is set to "Full (Slave)" or "Simple(Slave)", the slave mode is automatically set. In other settings, the master mode is set.

I/O allocation

Signals for general I/O (DI/O, GI/O, etc.) and special I/O (UI/O, RI/O, etc.) are called logical signals. In the robot controller, logical signals are subjected to signal processing.

On the other hand, signals of actual I/O devices are called physical signals. Physical signals specify a device using a rack and slot and specify each signal using its signal number in the device.

To control the signals of I/O devices in the robot controller, establish an association between physical signals and logical signals. The association is called I/O allocation.

Generally, I/O allocation is performed automatically.

Standard I/O allocation

When I/O allocation is deleted and the power of the robot controller is turned on, the connected I/O devices are recognized and proper I/O allocation is automatically performed. The I/O allocation is called the standard I/O allocation.

There are seven types of I/O allocation as shown below and each type has different allocation of peripheral I/O signals (UOP).

Name	UOP allocation type	I/O device to which UOP is allocated
None	No allocation	None
Full	All allocation	I/O link master interface
Full (Slave)	All allocation	I/O link slave interface
Full (CRMA16)	All allocation	Peripheral device control interface (CRMA15 and CRMA16)
Simple	Simple allocation	I/O link master interface
Simple (Slave)	Simple allocation	I/O link slave interface
Simple (CRMA16)	Simple allocation	Peripheral device control interface (CRMA16)

In default setting, the LR tool is set to "Simple (Slave)" and the LR handling tool is set to "Simple (CRMA16)".

Fig. 3-1(b)~(d) shows the interfaces to which each type of standard allocation is performed. To perform standard allocation, set "Automatic UOP allocation" as described below.

Procedure for setting automatic UOP allocation

CAUTION

This operation deletes all I/O allocation settings.

- 1 Press the MENUS key to display the screen menu.
- 2 Select "0 -NEXT--" and then select "6 SYSTEM".
- 3 Press F1 "TYPE" to display the screen switching menu.
- 4 Select "Config".
- 5 Move the cursor to the "UOP auto assignment" line.

```

47 UOP auto assignment:  Simple(CRMA16)
[ TYPE ] [ CHOICE ]

```

- 6 Press F4 "CHOICE" to display choices.

```

1 None          5 Simple
2 Full          6 Simple (Slave)
3 Full (Slave)  7 Simple (CRMA16)
4 Full (CRMA16) 8
47 UOP auto assignment:  Simple(CRMA16)
[ TYPE ] [ CHOICE ]

```

- 7 Select the type of the standard I/O allocation.
(When "Full (Slave)" or "Simple (Slave)" is selected, I/O link is set to the slave mode (\$IOMASTER=0). When another type is selected, it is set to the master mode (\$IOMASTER=1).)
- 8 When "Clear ALL I/O assignment to apply this?" appears, press F4 "YES".
- 9 When "Cycle power to apply new UOP assignment" appears, turn the power of the robot controller off and back on again.

UOP allocation types

There are the following two types of peripheral I/O signal (UOP) allocation.

All allocation

All peripheral I/O signals can be used.

Eighteen input physical signals and twenty output physical signals are allocated to peripheral I/O signals.

Simple allocation

The peripheral I/O signals that the number of signals is small can be used.

Eight input physical signals and four output physical signals are allocated to peripheral I/O signals.

In simple allocation, the number of signals that can be used for general digital I/O is increased because the number of peripheral I/O signals is decreased, but the functions of peripheral I/O signals are restricted as shown in the table below.

UI[1]	IMSTP	Always ON
UI[2]	HOLD	Operable
UI[3]	SFSPD	Always ON
UI[4]	CSTOPI	Allocated to the same signal as in RESET *1
UI[5]	RESET	Operable
UI[6]	START	Operable
UI[7]	HOME	No allocation
UI[8]	ENBL	Operable
UI[9]	RSR1/PNS1	Operable as PNS1 *3
UI[10]	RSR2/PNS2	Operable as PNS2 *3
UI[11]	RSR3/PNS3	Operable as PNS3 *3

UO[1]	CMDENBL	Operable
UO[2]	SYSRDY	No allocation
UO[3]	PROGRUN	No allocation
UO[4]	PAUSED	No allocation
UO[5]	HELD	No allocation
UO[6]	FAULT	Operable
UO[7]	ATPERCH	No allocation
UO[8]	TPENBL	No allocation
UO[9]	BATALM	Operable
UO[10]	BUSY	Operable
UO[11]	ACK1/SNO1	No allocation

UI[12]	RSR4/PNS4	Operable as PNS4 *3
UI[13]	RSR5/PNS5	No allocation
UI[14]	RSR6/PNS6	No allocation
UI[15]	RSR7/PNS7	No allocation
UI[16]	RSR8/PNS8	No allocation
UI[17]	PNSTROBE	Allocated to the same signal as in START *2
UI[18]	PROD_START	No allocation

UO[12]	ACK2/SNO2	No allocation
UO[13]	ACK3/SNO3	No allocation
UO[14]	ACK4/SNO4	No allocation
UO[15]	ACK5/SNO5	No allocation
UO[16]	ACK6/SNO6	No allocation
UO[17]	ACK7/SNO7	No allocation
UO[18]	ACK8/SNO8	No allocation
UO[19]	SNACK	No allocation
UO[20]	RESERVE	No allocation

- *1 Since CSTOPI and RESET are allocated to the same signal, reset input can forcibly terminate the program if " CSTOPI for ABORT" is enabled.
- *2 Since PNSTROBE and START are allocated to the same signal, the program is selected at the rising edge (OFF→ON) of the START signal and the program is started at the falling edge (ON→OFF) of the START signal.
- *3 RSR cannot be used in simple allocation. Even if RSR is selected on the Prog Select screen, PNS is automatically selected during power-on.

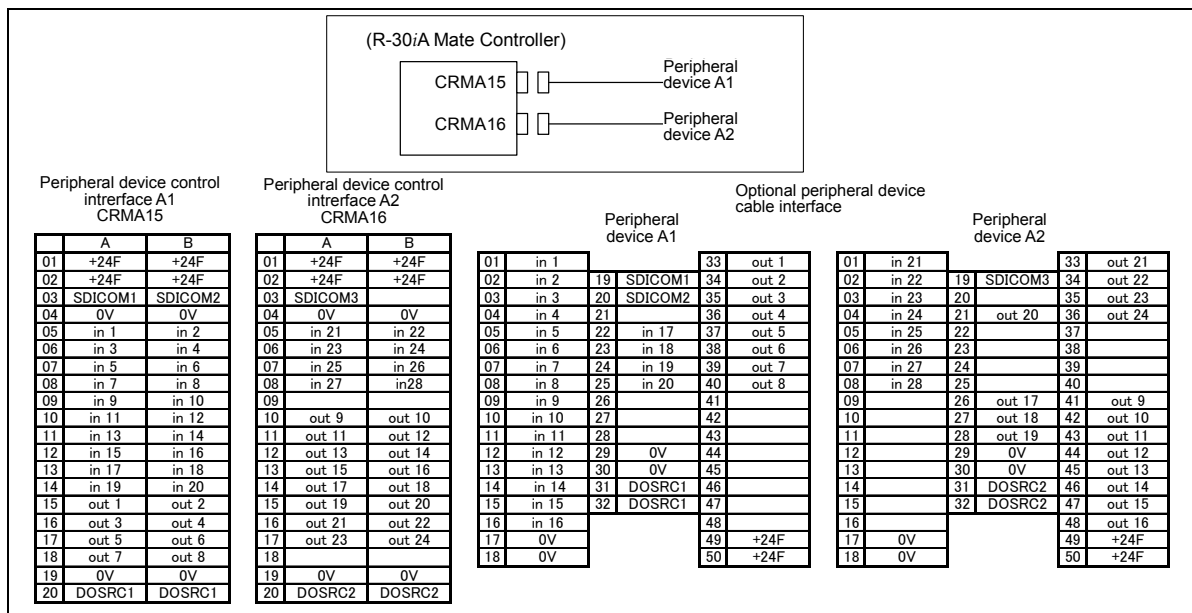
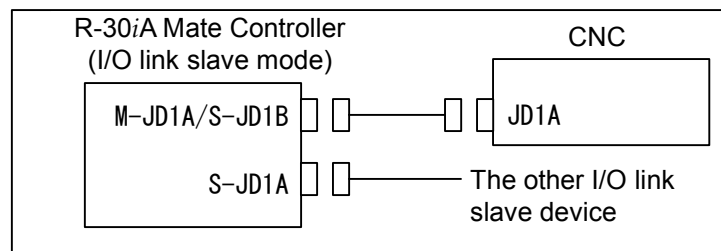


Fig. 3.1 (b) Allocation of peripheral device control interfaces (1/2)

Standard allocation of peripheral device control interfaces											
Automatic UOP allocation: Simple (CRMA16)				Automatic UOP allocation: All (CRMA16)				Automatic UOP allocation: None Automatic UOP allocation: All Automatic UOP allocation: Simple Automatic UOP allocation: All(slave) Automatic UOP allocation: Simple(slave)			
in 1	DI[101]			in 1	UI[1]	*IMSTP		in 1	DI[101]		
in 2	DI[102]			in 2	UI[2]	*HOLD		in 2	DI[102]		
in 3	DI[103]			in 3	UI[3]	*SFSPD		in 3	DI[103]		
in 4	DI[104]			in 4	UI[4]	CSTOPI		in 4	DI[104]		
in 5	DI[105]			in 5	UI[5]	RESET		in 5	DI[105]		
in 6	DI[106]			in 6	UI[6]	START		in 6	DI[106]		
in 7	DI[107]			in 7	UI[7]	HOME		in 7	DI[107]		
in 8	DI[108]			in 8	UI[8]	ENBL		in 8	DI[108]		
in 9	DI[109]			in 9	UI[9]	RSR1/PNS1		in 9	DI[109]		
in 10	DI[110]			in 10	UI[10]	RSR2/PNS2		in 10	DI[110]		
in 11	DI[111]			in 11	UI[11]	RSR3/PNS3		in 11	DI[111]		
in 12	DI[112]			in 12	UI[12]	RSR4/PNS4		in 12	DI[112]		
in 13	DI[113]			in 13	UI[13]	RSR5/PNS5		in 13	DI[113]		
in 14	DI[114]			in 14	UI[14]	RSR6/PNS6		in 14	DI[114]		
in 15	DI[115]			in 15	UI[15]	RSR7/PNS7		in 15	DI[115]		
in 16	DI[116]			in 16	UI[16]	RSR8/PNS8		in 16	DI[116]		
in 17	DI[117]			in 17	UI[17]	PNSTROBE		in 17	DI[117]		
in 18	DI[118]			in 18	UI[18]	PROD START		in 18	DI[118]		
in 19	DI[119]			in 19	DI[119]			in 19	DI[119]		
in 20	DI[120]			in 20	DI[120]			in 20	DI[120]		
in 21	UI[2]	*HOLD		in 21	DI[81]			in 21	DI[81]		
in 22	UI[5]	RESET *1		in 22	DI[82]			in 22	DI[82]		
in 23	UI[6]	START *2		in 23	DI[83]			in 23	DI[83]		
in 24	UI[8]	ENBL		in 24	DI[84]			in 24	DI[84]		
in 25	UI[9]	PNS1		in 25	DI[85]			in 25	DI[85]		
in 26	UI[10]	PNS2		in 26	DI[86]			in 26	DI[86]		
in 27	UI[11]	PNS3		in 27	DI[87]			in 27	DI[87]		
in 28	UI[12]	PNS4		in 28	DI[88]			in 28	DI[88]		
out 1	DO[101]			out 1	UO[1]	CMDENBL		out 1	DO[101]		
out 2	DO[102]			out 2	UO[2]	SYSRDY		out 2	DO[102]		
out 3	DO[103]			out 3	UO[3]	PROGRUN		out 3	DO[103]		
out 4	DO[104]			out 4	UO[4]	PAUSED		out 4	DO[104]		
out 5	DO[105]			out 5	UO[5]	HELD		out 5	DO[105]		
out 6	DO[106]			out 6	UO[6]	FAULT		out 6	DO[106]		
out 7	DO[107]			out 7	UO[7]	ATPERCH		out 7	DO[107]		
out 8	DO[108]			out 8	UO[8]	TPENBL		out 8	DO[108]		
out 9	DO[109]			out 9	UO[9]	BATALM		out 9	DO[109]		
out 10	DO[110]			out 10	UO[10]	BUSY		out 10	DO[110]		
out 11	DO[111]			out 11	UO[11]	ACK1/SNO1		out 11	DO[111]		
out 12	DO[112]			out 12	UO[12]	ACK2/SNO2		out 12	DO[112]		
out 13	DO[113]			out 13	UO[13]	ACK3/SNO3		out 13	DO[113]		
out 14	DO[114]			out 14	UO[14]	ACK4/SNO4		out 14	DO[114]		
out 15	DO[115]			out 15	UO[15]	ACK5/SNO5		out 15	DO[115]		
out 16	DO[116]			out 16	UO[16]	ACK6/SNO6		out 16	DO[116]		
out 17	DO[117]			out 17	UO[17]	ACK7/SNO7		out 17	DO[117]		
out 18	DO[118]			out 18	UO[18]	ACK8/SNO8		out 18	DO[118]		
out 19	DO[119]			out 19	UO[19]	SNACK		out 19	DO[119]		
out 20	DO[120]			out 20	UO[20]	Reserve		out 20	DO[120]		
out 21	UO[1]	CMDENBL		out 21	DO[81]			out 21	DO[81]		
out 22	UO[6]	FAULT		out 22	DO[82]			out 22	DO[82]		
out 23	UO[9]	BATALM		out 23	DO[83]			out 23	DO[83]		
out 24	UO[10]	BUSY		out 24	DO[84]			out 24	DO[84]		

*1: The physical number "in 22" is also allocated to UI[4](CSTOP)
 *2: The physical number "in 23" is also allocated to UI[17](PNSTROBE)

Fig. 3.1 (b) Allocation of peripheral device control interfaces (2/2)



I/O Link slave interface
Input signal address (72 point)

	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
0	in 1	in 2	in 3	in 4	in 5	in 6	in 7	in 8
1	in 9	in 10	in 11	in 12	in 13	in 14	in 15	in 16
2	in 17	in 18	in 19	in 20	in 21	in 22	in 23	in 24
3	in 25	in 26	in 27	in 28	in 29	in 30	in 31	in 32
4	in 33	in 34	in 35	in 36	in 37	in 38	in 39	in 40
5	in 41	in 42	in 43	in 44	in 45	in 46	in 47	in 48
6	in 49	in 50	in 51	in 52	in 53	in 54	in 55	in 56
7	in 57	in 58	in 59	in 60	in 61	in 62	in 63	in 64
8	in 65	in 66	in 67	in 68	in 69	in 70	in 71	in 72

Output signal address (68 point)

	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
0	out 1	out 2	out 3	out 4	out 5	out 6	out 7	out 8
1	out 9	out 10	out 11	out 12	out 13	out 14	out 15	out 16
2	out 17	out 18	out 19	out 20	out 21	out 22	out 23	out 24
3	out 25	out 26	out 27	out 28	out 29	out 30	out 31	out 32
4	out 33	out 34	out 35	out 36	out 37	out 38	out 39	out 40
5	out 41	out 42	out 43	out 44	out 45	out 46	out 47	out 48
6	out 49	out 50	out 51	out 52	out 53	out 54	out 55	out 56
7	out 57	out 58	out 59	out 60	out 61	out 62	out 63	out 64
8	out 65	out 66	out 67	out 68				

Fig. 3.1 (c) Allocation of I/O link slave interface (1/2)

Standard allocation of I/O link slave interfaces

Automatic UOP allocation :
Simple (Slave)Automatic UOP allocation :
All (Slave)Automatic UOP allocation: None *3
Automatic UOP allocation: All *3
Automatic UOP allocation: Simple *3
Automatic UOP allocation: All(CRMA16) *3
Automatic UOP allocation: Simple(CRMA16) *3

in 1	UI[2]	*HOLD
in 2	UI[5]	RESET *1
in 3	UI[6]	START *2
in 4	UI[8]	ENBL
in 5	UI[9]	PNS1
in 6	UI[10]	PNS2
in 7	UI[11]	PNS3
in 8	UI[12]	PNS4
in 9	DI[1]	
in 10	DI[2]	
in 11	DI[3]	
in 12	DI[4]	
in 13	DI[5]	
in 14	DI[6]	
in 15	DI[7]	
in 16	DI[8]	
in 17	DI[9]	
in 18	DI[10]	
in 19	DI[11]	
in 20	DI[12]	
in 21	DI[13]	
in 22	DI[14]	
:	:	
in N	DI[N-8]	
:	:	
in 72	DI[64]	

in 1	UI[1]	*IMSTP
in 2	UI[2]	*HOLD
in 3	UI[3]	*SFSPD
in 4	UI[4]	CSTOPI
in 5	UI[5]	RESET
in 6	UI[6]	START
in 7	UI[7]	HOME
in 8	UI[8]	ENBL
in 9	UI[9]	RSR1/PNS1
in 10	UI[10]	RSR2/PNS2
in 11	UI[11]	RSR3/PNS3
in 12	UI[12]	RSR4/PNS4
in 13	UI[13]	RSR5/PNS5
in 14	UI[14]	RSR6/PNS6
in 15	UI[15]	RSR7/PNS7
in 16	UI[16]	RSR8/PNS8
in 17	UI[17]	PNSTROBE
in 18	UI[18]	PROD START
in 19	DI[1]	
in 20	DI[2]	
in 21	DI[3]	
in 22	DI[4]	
:	:	
in N	DI[N-18]	
:	:	
in 72	DI[54]	

in 1	DI[1]	
in 2	DI[2]	
in 3	DI[3]	
in 4	DI[4]	
in 5	DI[5]	
in 6	DI[6]	
in 7	DI[7]	
in 8	DI[8]	
in 9	DI[9]	
in 10	DI[10]	
in 11	DI[11]	
in 12	DI[12]	
in 13	DI[13]	
in 14	DI[14]	
in 15	DI[15]	
in 16	DI[16]	
in 17	DI[17]	
in 18	DI[18]	
in 19	DI[19]	
in 20	DI[20]	
in 21	DI[21]	
in 22	DI[22]	
:	:	
in N	DI[N]	
:	:	
in 72	DI[72]	

out 1	UO[1]	CMDENBL
out 2	UO[6]	FAULT
out 3	UO[9]	BATALM
out 4	UO[10]	BUSY
out 5	DO[1]	
out 6	DO[2]	
out 7	DO[3]	
out 8	DO[4]	
out 9	DO[5]	
out 10	DO[6]	
out 11	DO[7]	
out 12	DO[8]	
out 13	DO[9]	
out 14	DO[10]	
out 15	DO[11]	
out 16	DO[12]	
out 17	DO[13]	
out 18	DO[14]	
out 19	DO[15]	
out 20	DO[16]	
out 21	DO[17]	
out 22	DO[18]	
:	:	
out N	DO[N-4]	
:	:	
out 68	DO[64]	

out 1	UO[1]	CMDENBL
out 2	UO[2]	SYSRDY
out 3	UO[3]	PROGRUN
out 4	UO[4]	PAUSED
out 5	UO[5]	HELD
out 6	UO[6]	FAULT
out 7	UO[7]	ATPERCH
out 8	UO[8]	TPENBL
out 9	UO[9]	BATALM
out 10	UO[10]	BUSY
out 11	UO[11]	ACK1/SNO1
out 12	UO[12]	ACK2/SNO2
out 13	UO[13]	ACK3/SNO3
out 14	UO[14]	ACK4/SNO4
out 15	UO[15]	ACK5/SNO5
out 16	UO[16]	ACK6/SNO6
out 17	UO[17]	ACK7/SNO7
out 18	UO[18]	ACK8/SNO8
out 19	UO[19]	SNACK
out 20	UO[20]	Reserve
out 21	DO[1]	
out 22	DO[2]	
:	:	
out N	DO[N-20]	
:	:	
out 68	DO[48]	

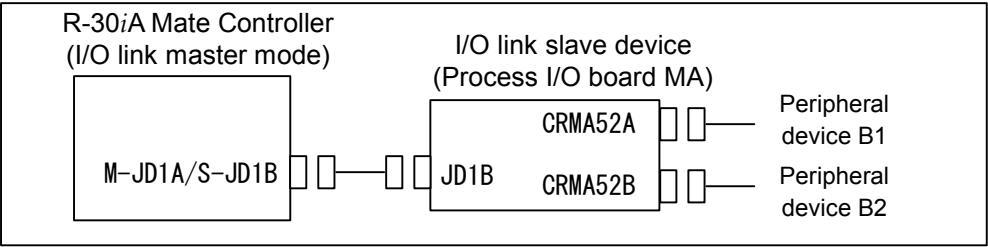
out 1	DO[1]	
out 2	DO[2]	
out 3	DO[3]	
out 4	DO[4]	
out 5	DO[5]	
out 6	DO[6]	
out 7	DO[7]	
out 8	DO[8]	
out 9	DO[9]	
out 10	DO[10]	
out 11	DO[11]	
out 12	DO[12]	
out 13	DO[13]	
out 14	DO[14]	
out 15	DO[15]	
out 16	DO[16]	
out 17	DO[17]	
out 18	DO[18]	
out 19	DO[19]	
out 20	DO[20]	
out 21	DO[21]	
out 22	DO[22]	
:	:	
out N	DO[N]	
:	:	
out 68	DO[68]	

*1 : The physical number "in 2" is also allocated to UI[4](CSTOPI)

*2 : The physical number "in 3" is also allocated to UI[17](PNSTROBE)

*3 : When automatic UOP allocation is neither "Simple (Slave)" nor "All (Slave)", I/O link mode is master mode by default, and I/O link slave is not available. The allocation described here is the standard allocation in the case that I/O link mode is changed to slave mode.

Fig. 3.1 (c) Allocation of I/O link slave interface (2/2)



Peripheral device control interface of Process I/O board MA

Peripheral device interface B1 CRMA52A			Peripheral device interface B2 CRMA52B		
	A	B		A	B
01	+24F	+24F	01	+24F	+24F
02	in 1	in 2	02	in 11	in 12
03	in 3	in 4	03	in 13	in 14
04	in 5	in 6	04	in 15	in 16
05	in 7	in 8	05	in 17	in 18
06	in 9	in 10	06	in 19	in 20
07			07		
08	out 1	out 2	08	out 9	out 10
09	out 3	out 4	09	out 11	out 12
10	out 5	out 6	10	out 13	out 14
11	out 7	out 8	11	out 15	out 16
12			12		
13	0V	0V	13	0V	0V
14	0V	0V	14	0V	0V
15	DOSRC3	DOSRC3	15	DOSRC3	DOSRC3

Fig. 3.1 (d) Allocation of I/O link master interface (1/2)

Standard allocation of I/O link master interface (in 1-20, out 1-167 can be used for Process I/O board MA)

Automatic UOP allocation : Simple

Automatic UOP allocation : All *3

Automatic UOP allocation: None

Automatic UOP allocation: All(CRMA16)

Automatic UOP allocation: Simple(CRMA16)

Automatic UOP allocation: All(Slave) *4

Automatic UOP allocation: Simple(Slave) *4

in 1	UI[2]	*HOLD
in 2	UI[5]	RESET *1
in 3	UI[6]	START *2
in 4	UI[8]	ENBL
in 5	UI[9]	PNS1
in 6	UI[10]	PNS2
in 7	UI[11]	PNS3
in 8	UI[12]	PNS4
in 9	DI[121]	
in 10	DI[122]	
in 11	DI[123]	
in 12	DI[124]	
in 13	DI[125]	
in 14	DI[126]	
in 15	DI[127]	
in 16	DI[128]	
in 17	DI[129]	
in 18	DI[130]	
in 19	DI[131]	
in 20	DI[132]	
in 21	DI[133]	
:	:	:
in N	DI[N+112]	
:	:	:

in 1	UI[1]	*IMSTP
in 2	UI[2]	*HOLD
in 3	UI[3]	*SFSPD
in 4	UI[4]	CSTOPI
in 5	UI[5]	RESET
in 6	UI[6]	START
in 7	UI[7]	HOME
in 8	UI[8]	ENBL
in 9	UI[9]	RSR1/PNS1
in 10	UI[10]	RSR2/PNS2
in 11	UI[11]	RSR3/PNS3
in 12	UI[12]	RSR4/PNS4
in 13	UI[13]	RSR5/PNS5
in 14	UI[14]	RSR6/PNS6
in 15	UI[15]	RSR7/PNS7
in 16	UI[16]	RSR8/PNS8
in 17	UI[17]	PNSTROBE
in 18	UI[18]	PROD START
in 19	DI[121]	
in 20	DI[122]	
in 21	DI[123]	
:	:	:
in N	DI[N+102]	
:	:	:

in 1	DI[121]	
in 2	DI[122]	
in 3	DI[123]	
in 4	DI[124]	
in 5	DI[125]	
in 6	DI[126]	
in 7	DI[127]	
in 8	DI[128]	
in 9	DI[129]	
in 10	DI[130]	
in 11	DI[131]	
in 12	DI[132]	
in 13	DI[133]	
in 14	DI[134]	
in 15	DI[135]	
in 16	DI[136]	
in 17	DI[137]	
in 18	DI[138]	
in 19	DI[139]	
in 20	DI[140]	
in 21	DI[141]	
:	:	:
in N	DI[N+120]	
:	:	:

out 1	UO[1]	CMDENBL
out 2	UO[6]	FAULT
out 3	UO[9]	BATALM
out 4	UO[10]	BUSY
out 5	DO[121]	
out 6	DO[122]	
out 7	DO[123]	
out 8	DO[124]	
out 9	DO[125]	
out 10	DO[126]	
out 11	DO[127]	
out 12	DO[128]	
out 13	DO[129]	
out 14	DO[130]	
out 15	DO[131]	
out 16	DO[132]	
out 17	DO[133]	
out 18	DO[134]	
out 19	DO[135]	
out 20	DO[136]	
out 21	DO[137]	
:	:	:
out N	DO[N+116]	
:	:	:

out 1	UO[1]	CMDENBL
out 2	UO[2]	SYSRDY
out 3	UO[3]	PROGRUN
out 4	UO[4]	PAUSED
out 5	UO[5]	HELD
out 6	UO[6]	FAULT
out 7	UO[7]	ATPERCH
out 8	UO[8]	TPENBL
out 9	UO[9]	BATALM
out 10	UO[10]	BUSY
out 11	UO[11]	ACK1/SNO1
out 12	UO[12]	ACK2/SNO2
out 13	UO[13]	ACK3/SNO3
out 14	UO[14]	ACK4/SNO4
out 15	UO[15]	ACK5/SNO5
out 16	UO[16]	ACK6/SNO6
out 17	UO[17]	ACK7/SNO7
out 18	UO[18]	ACK8/SNO8
out 19	UO[19]	SNACK
out 20	UO[20]	Reserve
out 21	DO[121]	
:	:	:
out N	DO[N+100]	
:	:	:

out 1	DO[121]	
out 2	DO[122]	
out 3	DO[123]	
out 4	DO[124]	
out 5	DO[125]	
out 6	DO[126]	
out 7	DO[127]	
out 8	DO[128]	
out 9	DO[129]	
out 10	DO[130]	
out 11	DO[131]	
out 12	DO[132]	
out 13	DO[133]	
out 14	DO[134]	
out 15	DO[135]	
out 16	DO[136]	
out 17	DO[137]	
out 18	DO[138]	
out 19	DO[139]	
out 20	DO[140]	
out 21	DO[141]	
:	:	:
out N	DO[N+120]	
:	:	:

*1 : The physical number "in 2" is also allocated to UI[4](CSTOPI)

*2 : The physical number "in 3" is also allocated to UI[17](PNSTROBE)

*3 : The number output of Process I/O board MA is 16, it is not enough for automatic UOP allocation "All", and the allocation will be the same as "None". The allocation described here is the case that the I/O link slave device with 18 or more input and 20 or more output is used.

*4 : When automatic UOP allocation is "Simple(Slave)" or "All(Slave)", I/O link mode is slave mode by default, and I/O link master is not available. The allocation described here is the standard allocation in the case that I/O link mode is changed to master mode.

Fig. 3.1 (d) Allocation of I/O link master interface (2/2)

3.1.1 Digital I/O

Digital I/O (DI/DO) is a group of general-purpose signals that send or receive the data of the peripheral equipment via the process I/O printed circuit board (or I/O unit). Moreover, this can send or receive the data of master (CNC) of I/O link. The digital signal is set on or off.

Configuration of Input/Output

In digital I/O, the configuration of the signal lines can be redefined. Set the following items.

**CAUTION**

Before the physical numbers are assigned, the use of the signals should be carefully checked. Otherwise, injury or property damage would occur.

- **RACK**

The rack indicates the kind of hardware which composes I/O module.

- 0 = Process I/O printed circuit board
- 1 to 16 = I/O Unit-MODEL A and MODEL B

Racks 1, 2, and so on are assigned to the base units of I/O unit model A and the interface units of I/O unit model B in the order in which they are connected.

- **SLOT**

The slot indicates the number of I/O module parts which composes RACK.

- When the process I/O printed circuit board is used, the first connected board is SLOT 1, the second is SLOT 2 and others are numbered sequentially as this.
- When the I/O unit model A is used, the number of the backplane slot in which the module is placed is the slot value of the module.
- When the I/O unit-MODEL B is used, the slot number of the basic unit is specified by the DIP switch in the basic unit.

- **START PT**

START PT allocates the logical number to the physical number to map the signal lines. The first physical number in the class of eight signals should be specified.

NOTE

- 1 A physical number specifies the pin of Input/Output lines on the I/O module. Logical number is allocated to this physical number. I/O configuration can be changed as every one signal.
- 2 Any physical number can be specified as the start point. Not allocated signal is automatically allocated to other logical number.

I/O configuration can be done with I/O configuration screen and I/O detail screen. When the allocation or settings of I/O is changed, turn the power off and on to use new information. When the kind of I/O printed circuit board are changed to the different one, I/O configuration may be done again.

Attribute of I/O

- **Polarity**

The polarity selects whether the current is switched on or off when the signal is set on.

- **NORMAL** = The current is turned on when the signal is set on.
- **INVERSE** = The current is turned on when the signal is set off.

- **Complementary**

Complementary is the function to set on or off two successive digital output signals: When a signal having an odd number goes on (off), complementary sets the next signal having an even number off (on).

- **Skipping simulated signals**

If a wait using a wait command is performed on an input signal set as a simulated one, the wait can be automatically canceled by detecting a timeout.

Output

The value of a digital output signal can be specified by executing a program or performing manual operation. (See Section 4.6, "I/O INSTRUCTION," and Section 6.4, "MANUAL I/O CONTROL".)

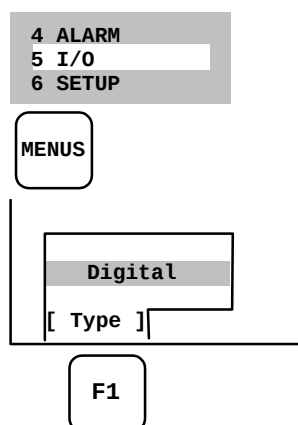
Simulated input/output

When simulated input/output is selected, a program can be tested without sending or receiving signals to or from the external equipment. (See Subsection 6.3.1, "Specifying Test Execution")

Procedure 3-1 Configuring Digital I/O

Step

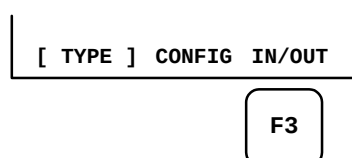
- 1 Press the MENUS key. The screen menu is displayed.
- 2 Select 5 [I/O].
- 3 Press F1 [TYPE]. The screen change menu is displayed.
- 4 Select "Digital."



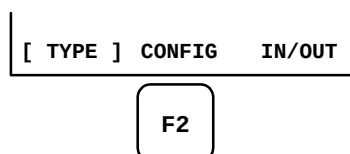
Digital I/O Selection Screen

I/O Digital Out				JOINT 30%
#	SIM	STATUS		
DO[1]	U	OFF	[	]
DO[2]	U	OFF	[	]
DO[3]	U	OFF	[	]
DO[4]	U	OFF	[	]
DO[5]	U	OFF	[	]
DO[6]	U	OFF	[	]
DO[7]	U	OFF	[	]
DO[8]	U	OFF	[	]
DO[9]	U	OFF	[	]
[TYPE] CONFIG IN/OUT ON OFF				

- 5 To switch the input screen to the output screen, or vice versa, press the F3 key, IN/OUT.



- 6 To allocate I/O, press F2, CONFIG. To return to the selection screen, press F2, MONITOR.



Digital I/O Configuration Screen

I/O Digital Out					JOINT 10%
#	RANGE	RACK	SLOT	START	STAT.
1	DO[1- 20]	0	1	21	ACTIV
2	DO[21-512]	0	1	0	UNASG
[TYPE] MONITOR IN/OUT DELETE HELP					

- 7 Manipulating the I/O assignment screen
 - a) Place the cursor on "Range," and specify the range of signals to be assigned.
 - b) Line division is performed automatically according to the specified range.
 - c) Enter appropriate values for "Rack," "Slot," and "Start point."
 - d) When the entered values are valid, abbreviation "PEND" is displayed in "Status."

If any entered value is invalid, abbreviation "INVAL" is displayed in "Status."

Unnecessary lines can be deleted by pressing F4 (Delete).

The abbreviations that will appear in “Status” mean the following:

ACTIV : This assignment is now in use.

PEND : Assignment is normal. Turning the power off and on again causes the ACTIV status to be entered.

INVAL : A specified value is invalid.

UNASG : No assignment has been made.

- 8 To return to the list screen, press F2, MONITOR.

I/O Digital Out				JOINT 30%
#	SIM	STATUS		
DO[1]	U	OFF	[DT SIGNAL 1]	
DO[2]	U	OFF	[DT SIGNAL 2]	
DO[3]	U	OFF	[DT SIGNAL 3]	
DO[4]	U	OFF	[DT SIGNAL 4]	
[TYPE] CONFIG IN/OUT DETAIL HELP >				

- 9 To set the attribute of I/O, press NEXT key and press F4, DETAIL of the next page.

DETAIL HELP >

F4

Digital I/O detail Screen

I/O Digital Out	JOINT 10%
Port Detail	1/3
Digital Output: [1]	
1 Comment: []	
2 Polarity: Normal	
3 Complementary: FALSE [1 - 2]	
[TYPE] PRV-PT NXT-PT	

To return to the list screen, press PREV key.

- 10 To add a comment:

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

JOINT 30%
1 []

ENTER

- b Select the method of naming the comment.

- c Press the appropriate function keys to add the comment.

- d When you are finished, press the ENTER key.

- 11 To set the item, move the cursor to the setting column, and select the function key menu.

- 12 To set the next digital I/O group, presses F3, NEXT.

[TYPE]	PREV	NEXT

F3

- 13 When you are finished, press the PREV key to return to the list screen.

I/O Digital Out				JOINT 30%
#	SIM	STATUS		
DO[1]	U	OFF	[DT SIGNAL 1]	
DO[2]	U	OFF	[DT SIGNAL 2]	
DO[3]	U	OFF	[DT SIGNAL 3]	
DO[4]	U	OFF	[DT SIGNAL 4]	
[TYPE] CONFIG IN/OUT DETAIL HELP >				

- 14 Turn off the controller. Turn on the controller so it can use the new information.

**WARNING**

Power should be turned on again to make a new setting valid. Otherwise, injury or property damage would occur.

**CAUTION**

- 1 In the first power-up after I/O re-allocation, power recovery would not be executed even if it is enabled.
- 2 After all I/O signals are set, the setting information should be saved in external storage in case the information needs to be re-loaded. Otherwise, the current setting information would be lost when it is changed.

- 15 To perform forced output or simulated input/output of a signal, place the cursor on ON or OFF and press the corresponding function key.

I/O Digital In			JOINT 30%
DI[1]	S	OFF	
IN/OUT ON OFF			
F4			

I/O Digital In			JOINT 30%
DI[1]	S	ON	[DIGITAL1]
[TYPE] CONFIG IN/OUT ON OFF			

For the forced output and simulated input/output of a signal, see Chapter 6, Section 6.4.

**WARNING**

The controller uses signals to control the peripheral equipment. The forced output or simulated input/output may adversely affect the security of the system. Check the use of signals in the system before attempting the forced output or simulated input/output.

3.1.2 Group I/O

Group I/O (GI/GO) is a group of general-purpose signals that send or receive the data by using two or more signal lines as the same group.

The value of the group I/O is represented in decimal or hexadecimal. When the data is sent, the value is transformed to the binary number.

Assignment of I/O signal

In the group I/O, the signal number can be defined to one group. Signal lines from 2 to 16 can be defined as one group. The defined group can overlap with the digital I/O.

NOTE

However, the defined group should not overlap with the digital output which is included in the complementary pair.

- RACK

The rack indicates the kind of hardware which composes I/O modules.

- 0 = process I/O printed circuit board
- 1 to 16 = I/O Unit-MODEL A / B

The base unit of the I/O unit-MODEL A and the interface unit of the I/O unit-MODEL B are defined as racks 1, 2, ... according to the sequence of connection.

- SLOT

The slot indicates the number of I/O module parts which composes the rack.

- When the process I/O printed circuit board is used, the first connected board is SLOT 1, the second is SLOT 2 and others are numbered sequentially as this.
- When the I/O unit model A is used, the number of the backplane slot in which the module is placed is the slot value of the module.
- When the I/O unit-MODEL B is used, the slot number of the basic unit is specified by the DIP switch in the basic unit.

- START PT

START PT allocates the logical number to the physical number to map the signal lines. The first physical number in the class of eight signals should be specified. The first physical number of the signal line is specified with this rack.

NOTE

- 1 A physical number specifies the Input/Output pin on the I/O module. Logical number is allocated to this physical number.
- 2 The physical number can begin with any number.
- 3 When two or more I/O boards are connected, the signal lines on the different boards can not be allocated to one group.

- NUM PTS

NUM PTS specifies the number of the digital signals which is assigned to one group.

NOTE

The number of the signal allocated to 1 group is from 2 to 16 points.

I/O configuration can be done with I/O configuration screen and I/O detail screen. When I/O configuration is changed, turn off the controller, and turn on the controller to use the new information.



CAUTION

At the first power-on after the I/O assignment is modified, the output signals are all off regardless of whether processing for power failures is enabled.

Execution of output

The value of the group output can be set by executing the program or manual I/O control.(See Section 4.6, "I/O INSTRUCTION", and Section 6.4,"MANUAL I/O CONTROL")

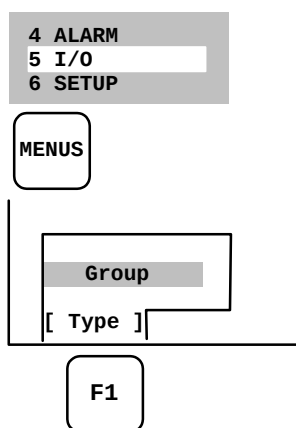
Execution of simulated I/O

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.(See Subsection 6.3.1 "Specifying Test Execution")

Procedure 3-2 Configuring group I/O

Step

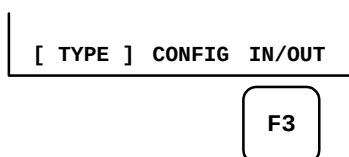
- 1 Press the MENUS key. The screen menu is displayed.
- 2 Select 5 [I/O].
- 3 Press F1 [TYPE]. The screen change menu is displayed.
- 4 Select Group. Group I/O list screen is displayed.



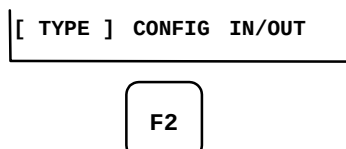
Group I/O list screen

I/O Group Out				JOINT 30%	
	#	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	[	]
[TYPE] CONFIG IN/OUT SIMULATE UNSIM					

- 5 To switch the input screen to the output screen, or vice versa, press the F3 key, IN/OUT.



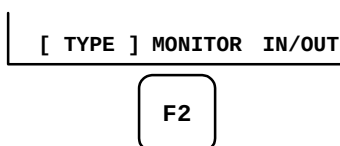
- 6 To allocate I/O, press F2,CONFIG.



Group I/O configuration screen

I/O Group Out				JOINT 30%	
GO #	RACK	SLOT	START PT	NUM	PTS
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0
9	0	0	0	0	0
[TYPE] MONITOR IN/OUT HELP >					

To return to the list screen, press F2,MONITOR.

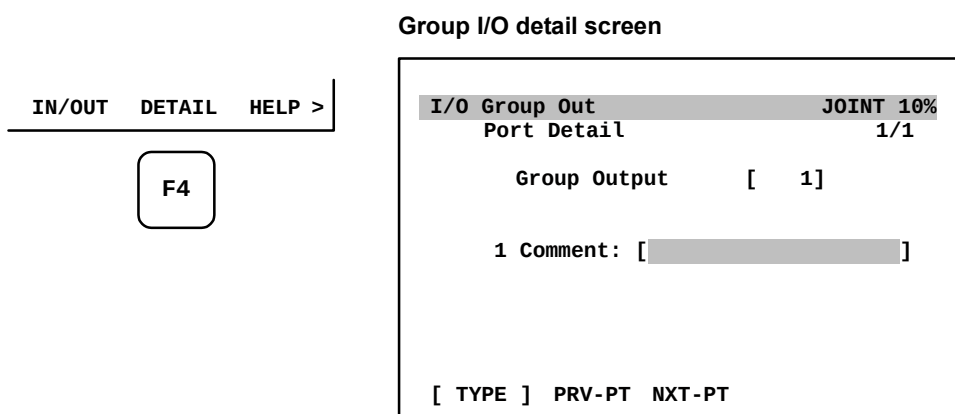


- 7 To configure the I/O, move the cursor to each item and type the value.

NOTE

The physical number to which the logical number of group I/O is assigned can be the same to which the digital I/O is assigned.

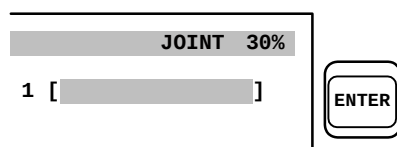
- 8 To set the attribute of I/O, press NEXT key of the selection screen and press F4,DETAIL of the next page.



To return to the list screen, press PREV key.



- 9 To add a comment:
- Move the cursor to the comment line and press the ENTER key.



- Select the method of naming the comment.
 - Press the appropriate function keys to add the comment.
 - When you are finished, press the ENTER key.
- 10 To set the item, move the cursor to the setting column, and select the function key menu.
- 11 When you are finished, press the PREV key to return to the list screen.
- 12 Turn off the controller. Turn on the controller so it can use the new information.

**WARNING**

Power should be turned on again to make a new setting valid. Otherwise, injury or property damage would occur.

**CAUTION**

- 1 In the first power-up after I/O re-allocation, power recovery would not be executed even if it is enabled.
- 2 After all I/O signals are set, the setting information should be saved in external storage in case the information needs to be re-loaded. Otherwise, the current setting information would be lost when it is changed.

3.1.3 Analog I/O

Analog I/O (AI/AO) signals are sent to and from the peripheral equipment via the input/output signal lines. The analog input/output voltages are converted to digital form when they are read or written. Therefore, they do not directly correspond to the input/output voltages.

Configuration of input/output

The physical numbers for the analog signal lines can be redefined.

NOTE

The standard configuration is factory-set up. To use a different configuration from the standard setting, make a reconfiguration.

**CAUTION**

Before the physical numbers are re-defined, the use of the signals should be carefully checked. Otherwise, injury or property damage would occur.

- **RACK**

Indicates the type of hardware composing the I/O modules.

- 0 = process I/O printed circuit board
- 1 to 16 = I/O unit-MODEL A / B

The base unit of the I/O unit-MODEL A and the interface unit of the I/O unit-MODEL B are defined as racks 1, 2, ..., according to the sequence of connection.

- **SLOT**

Indicates the number for the I/O module parts which compose RACK. The slot number for the backplane in the I/O unit-MODEL A serves as the slot number for the module.

- **CHANNEL**

Allocates the physical number to the logical number for mapping the signal lines.

NOTE

A physical number specifies the pin of an input/output line on the I/O module. The logical number is allocated to this physical number. This allocation can be altered.

I/O configuration can be done on the I/O configuration screen and I/O detail screen. When I/O configuration is changed, turn the controller off and on again to use the new information.

**CAUTION**

At the first power-on after the I/O assignment is modified, the output signals are all off regardless of whether processing for power failures is enabled.

Execution of output

The value of the analog output can be set by executing the program or manual I/O control (Sections 4.6 and 6.4).

Execution of simulated I/O

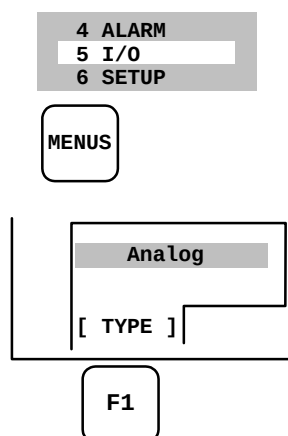
Simulating I/O allows you to test a program that uses I/O. Simulating I/O does not actually send output signals or receive input signals (Subsection 6.3.1).

Procedure 3-3 Setting analog I/O**NOTE**

The standard configuration is factory-set up. To use a different configuration from the standard setting, reconfigure the I/O.

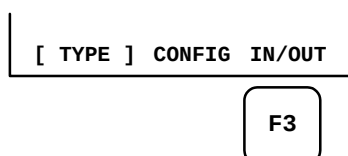
Step

- 1 Press the MENUS key. The screen menu is displayed.
- 2 Select 5, [I/O].
- 3 Press F1, [TYPE]. The screen change menu is displayed.
- 4 Select Analog. The analog I/O list screen is displayed.

**Analog I/O list screen**

I/O Analog In				JOINT 30%	
#	SIM	VALUE		1/25	
AI[1]	U	0	[	]	
AI[2]	U	0	[	]	
AI[3]	*	*	[	]	
AI[4]	*	*	[	]	
AI[5]	*	*	[	]	
AI[6]	*	*	[	]	
AI[7]	*	*	[	]	
AI[8]	*	*	[	]	
AI[9]	*	*	[	]	
AI[10]	*	*	[	]	
[TYPE] CONFIG IN/OUT SIMULATE UNSIM					

- 5 To switch the input screen to the output screen, press F3, [IN/OUT].



- 6 To allocate I/O, press F2, [CONFIG].

Analog I/O configuration screen

[TYPE] CONFIG IN/OUT

F2

I/O Analog In				JOINT 30%
				1/25
AI #	RACK	SLOT	CHANNEL	
1	0	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	

[TYPE] MONITOR IN/OUT
HELP >

To return to the list screen, press F2, [MONITOR].

[TYPE] MONITOR IN/OUT

F2

- 7 To configure the signals, move the cursor to each item and enter the value.
- 8 To return to the list screen, press F2, [MONITOR].

[TYPE] MONITOR IN/OUT

F2

I/O Analog In				JOINT 30%
				1/25
#	SIM	VALUE		
AI[1]	U	0	[analog signal 1]	
AI[2]	U	0	[analog signal 2]	
AI[3]	*	*	[]	
AI[4]	*	*	[]	

[TYPE] CONFIG IN/OUT
SIMULATE UNSIM

- 9 Press NEXT key of the list screen and press F4, [DETAIL] of the next page. The analog I/O detail screen is displayed.

Analog I/O detail screen

IN/OUT DETAIL HELP >

F4

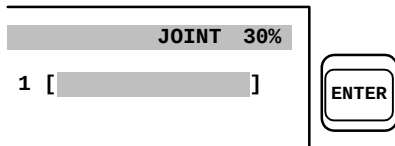
I/O Analog Out		JOINT 10%
		1/1
Port Detail		
Analog Output	[1]	
1 Comment: []		

[TYPE]
PRV-PT
NXT-PT

To return to the list screen, press the PREV key.

PREV

- 10 To add a comment:
- a Move the cursor to the comment line and press the ENTER key.



- b Select the method of naming the comment.
- c Press the appropriate function keys to add the comment.
- d When you are finished, press the ENTER key.
- 11 To specify the signal attribute, move the cursor to the corresponding field, and select the function key.
- 12 When you are finished, press the PREV key to return to the list screen.
- 13 Turn the controller off and on again so that it can use the new information.

**WARNING**

Power should be turned on again to make a new setting valid. Otherwise, injury or property damage would occur.

**CAUTION**

- 1 In the first power-up after I/O re-allocation, power recovery would not be executed even if it is enabled.
- 2 After all I/O signals are set, the setting information should be saved in external storage in case the information needs to be re-loaded. Otherwise, the current setting information would be lost when it is changed.

3.2 ROBOT I/O

Robot I/O are signals digital signals

Robot to operate the following executions.

- Other signals are used as the end effector I/O via the robot. The end effector I/O is connected to the connector at the end of the robot arm to enable its use.

The end effector I/O consists of eight input and eight output general-purpose signals. No signal numbers can be redefined for these signals.

NOTE

The number of general-purpose input/output signals of the end effector I/O depends on the model of the robot. Refer to the mechanical unit maintenance manual.

- Hand breakage input signal, *HBK

The *HBK signal is connected to the robot hand and detects a breakage in the tool. In the normal state, the *HBK signal is set on. When the *HBK signal goes off, an alarm occurs and the robot is immediately stopped.

NOTE

Hand breakage detection can be disabled on the system setting screen. See the item of enabling and disabling hand breakage detection in Section 3.15, "SYSTEM CONFIG MENU."

- **Abnormal air pressure input signal, *PPABN input**

The *PPABN signal detects a drop in the air pressure. In the normal state, the *PPABN signal is set on. When a drop in air pressure occurs, the *PPABN signal goes off, an alarm is issued, and the robot is immediately stopped.

- ***ROT input**

The overtravel (robot overtravel) signal indicates an overtravel along each axis of the mechanical unit of the robot. In the normal status, the *ROT signal is on. When this signal is turned off, an alarm is generated and the robot is stopped immediately.

The *ROT input does not appear on the cable terminal of the end effector because it is processed within the mechanical unit of the robot.

While the *HBK or *ROT signal is off, the alarm state can temporarily be released by holding down the shift key and pressing the alarm release key. While holding down the shift key, move the tool to the appropriate position by jog feed.

RI [1 to 8] INPUT**RO [1 to 8] OUTPUT**

The end effector signals, (RI [1 to 8] and RO [1 to 8]), are general-purpose input and output signals.

Attribute of I/O

- **Polarity**

The polarity selects whether the current is switched on or off when the signal is set on.

- **NORMAL** = The current is turned on when the signal is set on.
- **INVERSE** = The current is turned on when the signal is set off.

- **Complementary**

Complementary is the function to set on or off two successive digital output signals: When a signal having an odd number goes on (off), complementary sets the next signal having an even number off (on).

- **Skipping simulated signals**

If a wait using a wait command is performed on an input signal set as a simulated one, the wait can be automatically canceled by detecting a timeout.

Output

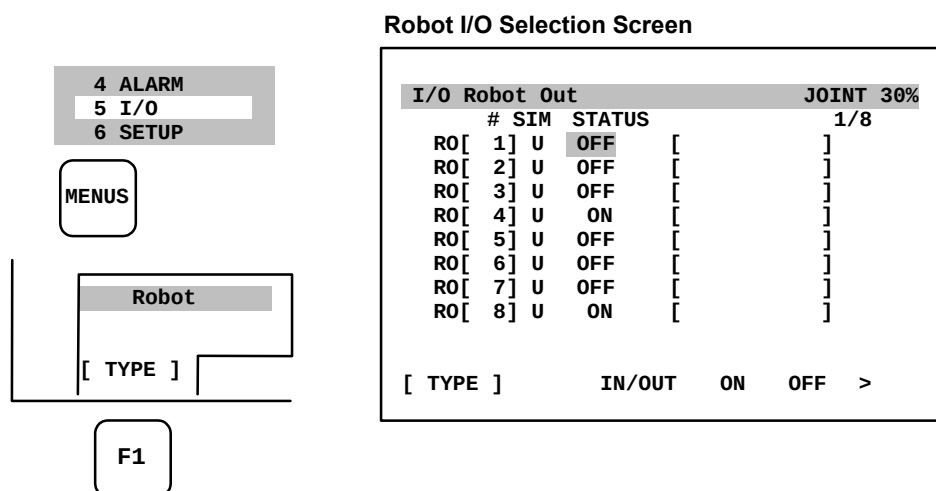
The value of a digital output signal can be specified by executing a program or performing manual operation. (See Section 4.6, "I/O INSTRUCTION," and Section 6.4, "MANUAL I/O CONTROL".)

Simulated input/output

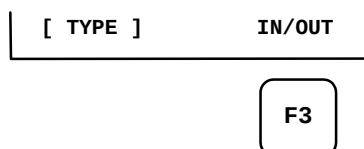
When simulated input/output is selected, a program can be tested without sending or receiving signals to or from the external equipment. (See Subsection 6.3.1, "Specifying Test Execution")

Procedure 3-4 Setting Robot I/O**Step**

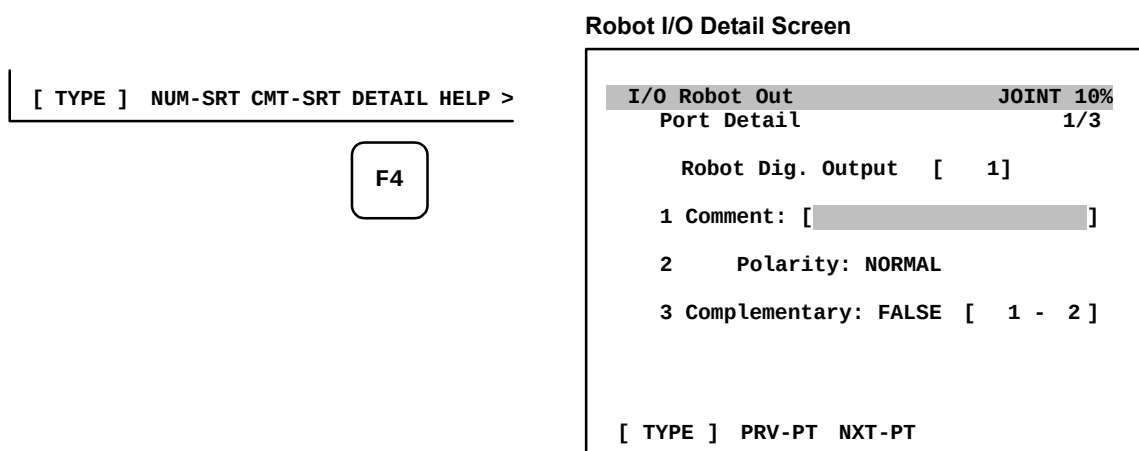
- 1 Press the MENUS key. The screen menu is displayed.
- 2 Select 5 (I/O).
- 3 Press the F1 key, [TYPE]. The screen change menu is displayed.
- 4 Select "Robot."



- 5 To switch the input screen to the output screen, press the F3 key, IN/OUT.



- 6 To set the attribute of I/O, press NEXT key and press F4, DETAIL of the next page.



To return to the list screen, press the PREV key.

- 7 To add a comment:
- Move the cursor to the comment line and press the ENTER key.
 - Select the method of naming the comment.
 - Press the appropriate function keys to add the comment.
 - When you are finished, press ENTER key.
- 8 To set the polarity and the complementary pair, move the cursor to the setting column, and select the function key menu.
- 9 When you are finished, press PREV to return to the list screen.

I/O Robot Out				JOINT 30%	
#	SIM	STATUS		1/8	
RO[1]	U	OFF	[	]	
RO[2]	U	OFF	[	]	
RO[3]	U	OFF	[	]	
RO[4]	U	ON	[	]	
[TYPE]			IN/OUT	ON	OFF

- 10 Turn off the controller. Turn on the controller so it can use the new information.

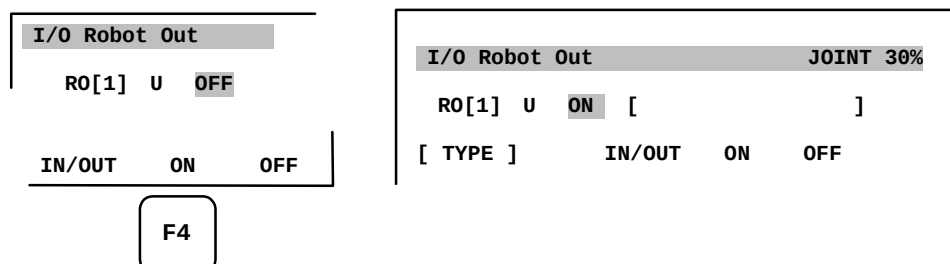
**WARNING**

Power should be turned on again to make a new setting valid. Otherwise, injury or property damage would occur.

**CAUTION**

After all I/O signals are set, the setting information should be saved in external storage in case the information needs to be re-loaded. Otherwise, the current setting information would be lost when it is changed.

- 11 To perform forced output of a signal, place the cursor on ON or OFF and press the corresponding function key.



For the forced output of a signal, see Chapter 6, Section 6.4.

**WARNING**

The controller uses signals to control the peripheral equipment. The forced output may adversely affect the security of the system. Check the use of signals in the system before attempting the forced output.

3.3 PERIPHERAL I/O

Peripheral I/O signals (UI/UO) are a group of specialized signals whose usage is decided by the system. These signals are connected with a remote controller and the peripheral devices and they are used to control the robot from the outside.

Configuration of I/O

There are two types of peripheral I/O signal allocation: all allocation and simple allocation. For details, see Section 3.1, "I/O".

Remote condition

When the robot is in the remote state, the program can be started by using the peripheral I/O. Signals(*HOLD,ENBL) which has relation to safety is always effective whether the remote condition is satisfied or not.

When the following remote conditions are satisfied, the robot is in the remote state.

- The teach pendant enable switch is set off.
- The remote signal (SI[2]) is on. (For how to turn the remote signal on and off, see the description of Remote/Local setup in Section 3.15, "SYSTEM CONFIG MENU.")
- The *SFSPD input of the peripheral device I/O is on.
- The ENBL input of the peripheral device I/O is on.
- A value of 0 (peripheral device) is set for system variable \$RMT_MASTER.

NOTE

\$RMT_MASTER Specifies the kind of remote device.

0 : Peripheral device

1 : CRT/KB

2 : Host computer

3 : No remote device

A program including a motion (group) can be started only when the remote conditions and the following operation conditions are satisfied:

- The ENBL signal of the peripheral I/O is set on.
- The servo power is on (not in the alarm state).

The CMDENBL signal indicates whether the above conditions are satisfied. The signal is output when the following conditions are satisfied:

- The remote conditions are satisfied.
- Not alarm status.
- The continuous operation mode is selected (the single step mode is disabled).

*IMSTP input UI [1] (Always enabled.)

The immediate stop signal turns servo power off by the software.

The *IMSTP input is on in the normal status. When this signal is turned off, the following processing is performed:

- An alarm is generated and the servo power is turned off.
- The robot operation is stopped immediately. Execution of the program is also stopped.



WARNING

The *IMSTP signal is controlled by software. The use of this signal for safety-critical processing is not recommended. Please use this signal together with the external emergency stop for safety-critical processing. For details of external emergency stop signal, refer to the "Maintenance Manual."

*HOLD input UI [2] (Always enabled.)

The temporary stop signal specifies a temporary stop from an external device.

The *HOLD input is on in the normal status. When this signal is turned off, the following processing is performed:

- The robot is decelerated until it stops, then the program execution is halted.
- If ENABLED is specified at "Break on hold" on the general item setting screen, the robot is stopped, an alarm is generated, and the servo power is turned off.

*SFSPD input UI [3] (Always enabled.)

The safety speed signal temporarily stops the robot when the safety fence door is opened. This signal is normally connected to the safety plug of the safety fence door.

The *SFSPD input is on in the normal status. When this signal is turned off, the following processing is performed:

- The operation being executed is decelerated and stopped, and execution of the program is also stopped. At this time, the feedrate override is reduced to the value specified for \$SCR.\$FENCEOVRD.
- When the *SFSPD input is off and a program is started from the teach pendant, the feedrate override is reduced to the value specified for \$SCR.\$SFRUNOVLIM. When jog feed is executed, the feedrate override is reduced to the value specified for \$SCR.\$SFJOGOVLIM. When *SFSPD is off, the feedrate override cannot exceed these values.

**WARNING**

The *SFSPD signal controls deceleration and stop by software. To stop the robot immediately for safety purposes, use this signal together with the FENCE signal on the emergency stop unit. For details of these signals, refer to the "Maintenance Manual."

NOTE

When the *IMSTP, *HOLD, and *SFSPD signals are not used, jumper these signal lines.

CSTOPI input UI [4] (Always enabled.)

The cycle stop signal terminates the program currently being executed. It also releases programs from the wait state by RSR.

- When FALSE is selected for CSTOPI for ABORT on the Config system setting screen, this signal terminates the program currently being executed as soon as execution of the program completes. It also releases (Clear) programs from the wait state by RSR. (Default)
- When TRUE is selected for CSTOPI for ABORT on the Config system setting screen, this signal immediately terminates the program currently being executed. It also releases (Clear) programs from the wait state by RSR.

**WARNING**

When FALSE is selected for CSTOPI for ABORT on the Config system setting screen, CSTOPI does not stop the program being executed until the execution is complete.

Fault reset input signal, RESET, UI [5]

The RESET signal cancels an alarm. If the servo power is off, the RESET signal turns on the servo power. The alarm output is not canceled until the servo power is turned on. The alarm is canceled at the instant this signal falls in default setting.

Enable input signal, ENBL, UI [8]

The ENBL signal allows the robot to be moved and places the robot in the ready state. When the ENBL signal is off, the system inhibits a jog feed of the robot and activation of a program including a motion (group). A program which is being executed is halted when the ENBL signal is set off.

NOTE

When the ENBL signal is not monitored, strap the signal with the ground.

RSR1 to RSR8 inputs UI [9-16] (Enabled in the remote state.)

These are robot service request signals. When one of these signals is received, the RSR program corresponding to the signal is selected and started to perform automatic operation. When another program is being executed or is stopped temporarily, the selected program is added to the queue and is started once the program being executed terminates. (→ Subsection 3.8.1, "Robot service request")

PNS1 to PNS8 UI [9-16] PNSTROBE UI [17] (Enabled in the remote state.)

These are program number select signals and a PN strobe signal. When the PNSTROBE input is received, the PNS1 to PNS8 inputs are read to select a program to be executed. When another program is being executed or temporarily stopped, these signals are ignored. (→ Subsection 3.8.2, "Program number select")

When the remote conditions are satisfied, program selection using the teach pendant is disabled while PNSTROBE is on.

PROD_START input UI [18] (Enabled in the remote state.)

The automatic operation start (production start) signal starts the currently selected program from line 1. This signal functions at its falling edge when turned off after being turned on.

When this signal is used together with a PNS signal, it executes the program selected by the PNS signal starting from line 1. When this signal is used together with no PNS signal, it executes the program selected using the teach pendant starting from line 1.

When another program is being executed or temporarily stopped, this signal is ignored. (Program number select Subsection 3.8.2)

START input UI [6] (Enabled in the remote state.)

This is an external start signal. This signal functions at its falling edge when turned off after being turned on. When this signal is received, the following processing is performed:

- When FALSE is selected for START for CONTINUE only on the Config system setting screen, the program selected using the teach pendant is executed from the line to which the cursor is positioned. A temporarily stopped program is also continued. (Default)
- When TRUE is selected for START for CONTINUE only on the Config system setting screen, a temporarily stopped program is continued. When the program is not temporarily stopped, it cannot be started.

NOTE

To start a program from a peripheral device, the RSR or PROD_START input is used. To start a temporarily stopped program, the START input is used.

CMDENBL input UO [1]

The input accept enable (command enable) signal is output when the following conditions are satisfied. This signal indicates that a program including an operation (group) can be started from the remote controllers.

- The remote conditions are satisfied.
- The operation enable conditions are satisfied.
- The mode is continuous operation (single step disable).

SYSRDY output UO [2]

SYSRDY is output while the servo power is on. This signal places the robot in the operation enable state. In the operation enable state, jog feed can be executed and a program involving an operation (group) can be started. The robot enters the operation enable state when the following operation enable conditions are satisfied:

- The ENBL input of the peripheral device I/O is on.
- The servo power is on (not in the alarm state).

PROGRUN output UO [3]

PROGRUN is output while a program is being executed. It is not output while a program is temporarily stopped.

PAUSED output UO [4]

PAUSED is output when a program is temporarily stopped and waits for restart.

HELD output UO [5]

HELD is output when the hold button is pressed or the HOLD signal is input. It is not output when the hold button is released.

FAULT output UO [6]

FAULT is output when an alarm occurs in the system. The alarm state is released by the FAULT_RESET input. FAULT is not output when a warning (WARN alarm) occurs.

ATPERCH output UO [7]

ATPERCH is output when the robot is in a previously defined reference position.

Up to three reference positions can be defined. This signal is output only when the robot is in the first reference position. For any other reference positions, general-purpose signals are assigned.

TPENBL output UO [8]

TPENBL is output when the enable switch of the teach pendant is set to on.

BATALM output UO [9]

BATALM indicates a low-voltage alarm for the backup battery of the controller or robot pulsecoder. Turn the power to the controller on and replace the battery.

BUSY output UO [10]

BUSY is output while a program is being executed or while processing using the teach pendant is being performed. It is not output while a program is temporarily stopped.

ACK1 to ACK8 outputs UO [11-18]

When the RSR function is enabled, ACK1 to ACK4 are used together with the function. When an RSR input is accepted, a pulse of the corresponding signal is output as an acknowledgment. The pulse width can be specified. (→ Subsection 3.8.1, "Robot service request")

SNO1 to SNO8 outputs UO [11-18]

When the PNS function is enabled, SNO1 to SNO8 are used together with the function. The currently selected program number (signal corresponding to the PNS1 to PNS8 inputs) is always output, in binary code, as confirmation. The selection of another program changes SNO1 to SNO8. (→ Subsection 3.8.2, "Program number select")

SNACK output UO [19]

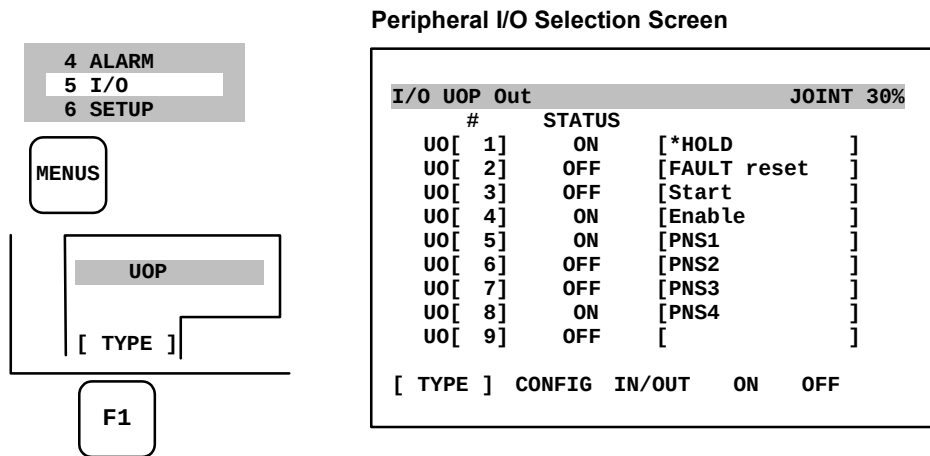
When the PNS function is enabled, SNACK is used together with the function. When the PNS inputs are accepted, a pulse of this signal is output as an acknowledgment. The pulse width can be specified. (→ Subsection 3.8.2, "Program number selection")

Procedure 3-5 Assigning Peripheral I/O

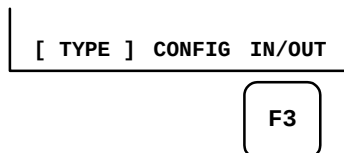
Step

- 1 Press the MENUS key. The screen menu is displayed.
- 2 Select 5, [I/O].
- 3 Press the F1 key, [TYPE]. The screen change menu is displayed.

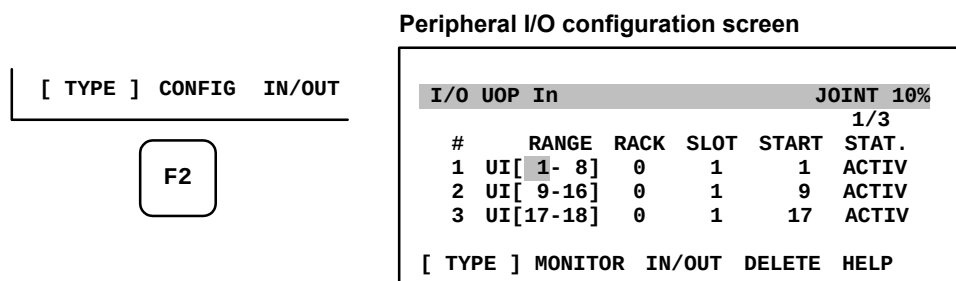
- 4 Select UOP.



- 5 To switch the input screen to the output screen, or vice versa, press the F3 key, IN/OUT.



- 6 To allocate I/O, press F2,CONFIG.



To return to the list screen, press F2,MONITOR

- 7 Manipulating the I/O assignment screen

- Place the cursor on "Range," and specify the range of signals to be assigned.
- Line division is performed automatically according to the specified range.
- Enter appropriate values for "Rack," "Slot," and "Start point."
- When the entered values are valid, abbreviation "PEND" is displayed in "Status."

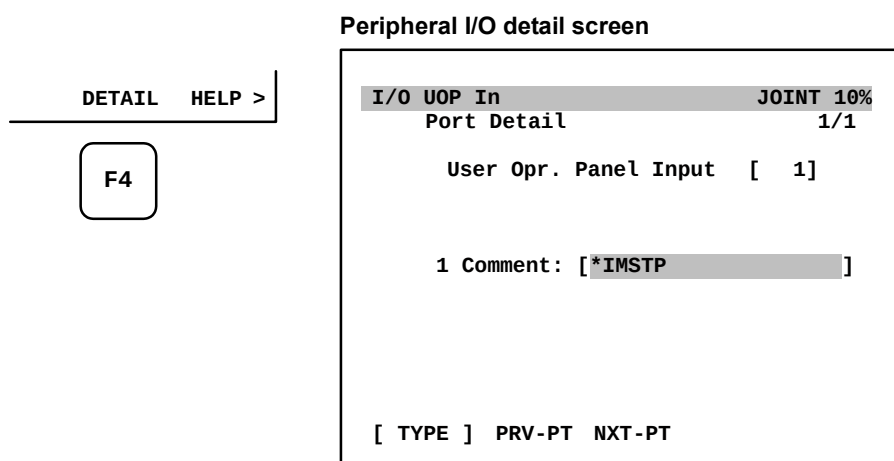
If any entered value is invalid, abbreviation "INVAL" is displayed in "Status."

Unnecessary lines can be deleted by pressing F4 (Delete).

The abbreviations that will appear in "Status" mean the following:

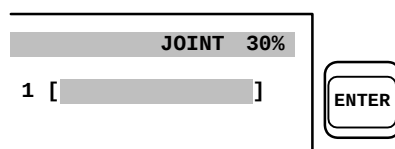
- ACTIV : This assignment is now in use.
 PEND : Assignment is normal. Turning the power off and on again causes the ACTIV status to be entered.
 INVAL : A specified value is invalid.
 UNASG : No assignment has been made.

- 8 To set the attribute of I/O, press NEXT key of the list screen and press F4, DETAIL of the next page.



To return to the list screen, press the PREV key.

- 9 To add a comment:
 - a Move the cursor to the comment line and press the ENTER key.



- b Select the method of naming the comment.
 - c Press the appropriate function keys to add the comment.
 - d When you are finished, press the ENTER key.

NOTE

The comment of peripheral equipment I/O is written by the tool software and can be changed. Even if the comment is rewritten, the function is not changed.

- 10 To set the item, move the cursor to the setting column, and select the function key menu.
- 11 When you are finished, press the PREV key to return to the list screen.
- 12 Turn off the controller. Turn on the controller so it can use the new information.



WARNING

Power should be turned on again to make a new setting valid. Otherwise, injury or property damage would occur.



CAUTION

- 1 In the first power-up after I/O re-allocation, power failure recovery would not be executed even if it is enabled.
- 2 After all I/O signals are set, the setting information should be saved in external storage in case the information needs to be re-loaded. Otherwise, the current setting information would be lost when it is changed.

**CAUTION**

When peripheral I/O signals are allocated to peripheral device control interfaces:

DI[101] to DI[118]	}	"Full (CRMA16)"
DO[101] to DO[120]		
DI[81 to 88]	}	"Simple (CRMA16)"
DO[81 to 84]		

Do not use the above items because they are allocated to the same physical numbers for peripheral I/O signals.

3.4 OPERATOR'S PANEL I/O

The operator's panel I/O means dedicated digital signals for passing data indicating the status of the buttons on the operator's panel/box

For the operator's panel I/O, the signal numbers cannot be mapped (redefined). Sixteen input and sixteen output signals are defined as standard. For the definition of the signals of the operator's panel I/O, see Fig. 3.4.

When the operator's panel is enabled, the operator's panel I/O can be used to start a program. However, any signals which have a significant effect on safety are always enabled.

The operator's panel is enabled when the following operator's panel enable conditions are satisfied:

- The enable switch on the teach pendant is set to off.
- The remote signal (SI[2]) is off. (For how to turn the remote signal on and off, see the description of #139-3-1 in Section 3.15, "SYSTEM CONFIG MENU.")
- The *SFSPD input of the peripheral device I/O is on.

To start a program involving operation (group), the following conditions must be satisfied:

- The ENBL input of the peripheral device I/O is on.
- The servo power is on (not in the alarm state).

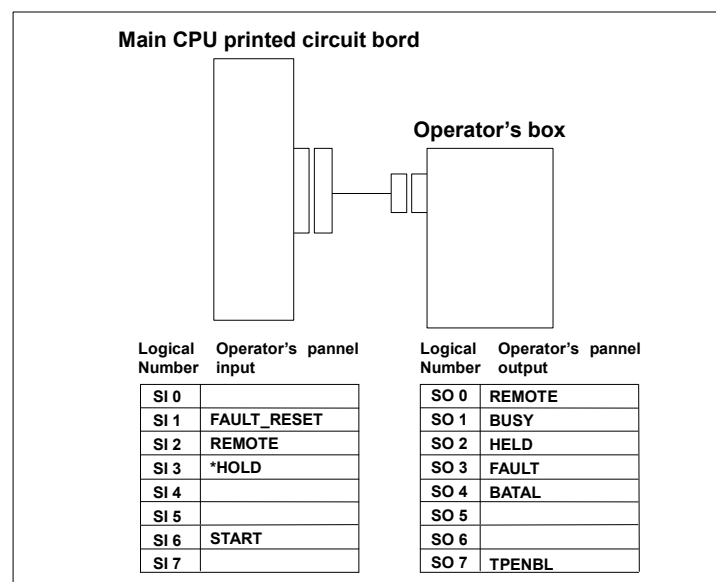


Fig. 3.4 Operator's Panel I/O

Table 3.4 (a) Operator's Panel Input Signals

Input signal	Description
*HOLD SI [3] Always enabled. Not provided for the operator box.	The temporary stop (hold) signal specifies temporary stop of the program. The *HOLD signal is on in the normal status. When this signal is turned off: - The robot operation being executed is decelerated, then stopped. - The program being executed is temporarily stopped.

Input signal	Description
FAULT_RESET SI [2] Always enabled. Not provided for the operator box.	The alarm release (fault reset) signal releases the alarm state. When the servo power is off, this signal turns on the servo power. In this case, the alarm state is not released until the servo power is turned on.
REMOTE SI [2] Always enabled. Not provided for the operator box.	The remote signal (remote) switches between remote mode and local mode of the system. In remote mode (SI[2] = on), when the remote conditions are satisfied, a program can be started using the peripheral device I/O. In local mode (SI[2] = off), when the operator's panel enable conditions are satisfied, a program can be started from the operator's panel. To turn the remote signal (SI[2]) on and off, set Remote/Local setup on the system config menu. For details, see Section 3.15, "SYSTEM CONFIG MENU."
START SI [6] Enabled in the operator's panel enable state.	The start signal starts the currently selected program using the teach pendant from the line to which the cursor is positioned or restarts a temporarily stopped program. This signal functions at its falling edge when turned off after being turned on.

Table 3.4 (b) Operator's Panel Output Signals

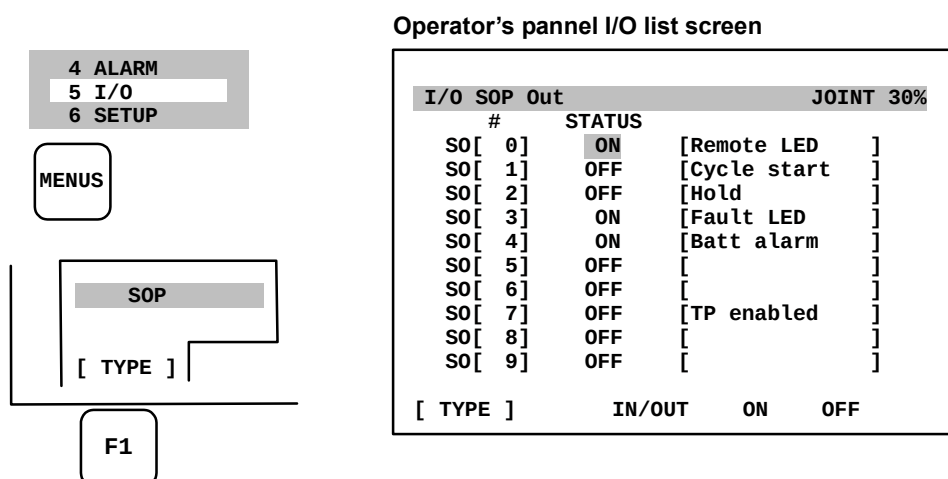
Output signal	Description
REMOTE SO [0] Not provided for the operator box.	The remote signal is output when the remote conditions are satisfied (remote conditions Section 3.3, "Peripheral I/O").
BUSY SO [1] Not provided for the operator box.	The busy signal is output while processing such as program execution or file transfer is being performed. It is not output when a program is temporarily stopped.
HELD SO [2] Not provided for the operator box.	The hold signal is output when the hold button is pressed or the HOLD signal is input.
FAULT SO [3] Not provided for the operator box.	The alarm (fault) signal is output when an alarm occurs in the system. The alarm state is released by the FAULT_RESET input. This signal is not output when a warning (WARN alarm) occurs.
BATAL output SO [4] Not provided for the operator box.	The abnormal battery (battery alarm) signal indicates a low-voltage alarm for the battery in the controller. While keeping the power to the controller on, replace the battery.
TPENBL output SO [7] Not provided for the operator box.	The teach pendant enable (TP enable) signal is output when the enable switch on the teach pendant is on.

Procedure 3-6 Displaying the operator's panel I/O**NOTE**

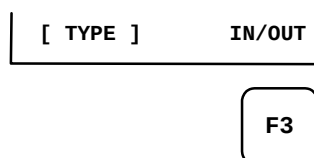
For the operator's panel I/O, the signal numbers cannot be redefined.

Step

- 1 Press MENUS to display the screen menu.
- 2 Select "5 I/O."
- 3 Press F1 [TYPE] to display the screen switching menu.
- 4 Select "SOP."



- 5 Press F3 (IN/OUT) to switch the display between the input and output screens.



NOTE

The input signal status can only be checked. Values cannot be changed forcibly.

3.5 I/O Link SCREEN

The I/O link screen can be used to make settings related to FANUC I/O unit model B and display the configuration of the I/O link units.

The I/O link screen consists of the following screens:

- I/O link list screen
- Model B unit list screen
- Number of ports setting screen

3.5.1 I/O Link List Screen

The I/O link list screen displays a list of I/O units in slave mode that are connected to the I/O link. It also displays the rack and slot numbers of each unit.

For I/O unit model A/B, only the interface units are displayed. In this case, a value of 0 is displayed for the rack number.

The following figure is an example of the I/O link list screen when process I/O board CA, one unit of I/O unit model B, and two units of I/O unit model A are connected to the robot controller. The names of the I/O units are displayed in the order in which the units are connected to the robot controller.

I/O Link Device			JOINT 100%	
			1/4	
Device name	Comment		Rack	Slot
1 PrcI/O CA	[	]	0	1
2 Model B	[	]	1	0
3 Model A	[	]	2	0
4 Model A	[	]	3	0
[TYPE]	DETAIL		CLR_ASG	

To display this screen, first press MENU to display the screen menu, then select "5 I/O." Then, press F1, [TYPE] to display the screen switching menu, then select Link Device.

The following table lists the device names displayed on the screen and the corresponding actual device names.

Word on TP	Device
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 FA	Process I/O Board FA
PrcI/O GA	Process I/O Board GA
PrcI/O HA	Process I/O Board HA
PrcI/O HB	Process I/O Board HB
PrcI/O HC	Process I/O Board HC
PrcI/O JA	Process I/O Board JA
PrcI/O JB	Process I/O Board JB
R-J2 Mate	R-J2 Mate. Slave Mode
Weld I/F	Weld Interface Board
Others	Other I/O devices except above devices

When F3 (DETAIL) is pressed, Model B screen or Number of Ports Setting Screen is displayed according to the type of the unit. When F3 (DETAIL) is pressed for the following units, the detail screen is displayed. When F3 (DETAIL) is pressed for other units, no screen change occurs. Each detail screen is described later.

Word on TP	Detail Screen
Model B	Model B
90-30 PLC	Number of Ports
I/O adptr	Number of Ports
R-J2 Mate	Number of Ports
Unknown	Number of Ports

On this screen, a comment can be specified for each I/O unit. Move the cursor to Comment and press the enter key. The screen enters comment input mode.

F5 (CLR_ASG) is described later.

3.5.2 Model B Unit List Screen

The model B unit list screen displays a list of units of FANUC I/O unit model B. FANUC I/O unit model B does not automatically recognize the connected DI/DO units. On this screen, set the types of the DI/DO units. The address set using the DIP switch of each DI/DO unit is used as the line number on this screen. One additional unit can be connected to each DI/DO unit. This screen can also be used to specify whether to connect an additional unit and the type of additional unit.

When the cursor is positioned to a "Model B" item on the I/O link list screen, press F3 (DETAIL) to display Model B screen as shown below:

I/O Link Device				JOINT 100%	
Model B				Rack 1	1/30
Slot	Base	Exp.	Comment		
1	*****	*****	[		]
2	*****	*****	[		]
3	*****	*****	[		]
:	:	:	:	:	:
30	*****	*****	[		]
[TYPE]		LIST [CHOICE]		CLR_ASG	

At first, nothing is set, as shown above. To use model B, set the types of the units on this screen.

When DI/DO unit BOD16A1 is connected to the interface unit and the address is set to 1, set the unit as shown below.

Position the cursor to the position shown above (Base column on line 1), then press F4, [CHOICE]. The options are displayed as shown below:

1	*****	5	BOA12A1
2	BID16A1	6	BIA16P1
3	BOD16A1	7	BMD88Q1
4	BMD88A1	8	

Slot	Base	Exp.	Comment
1	*****	*****	[]
2	*****	*****	[]
3	*****	*****	[]
:	:	:	:
30	*****	*****	[]

[TYPE]	LIST	[CHOICE]	CLR_ASG

Select BOD16A1 on this screen. The unit is set as shown below:

I/O Link Device				JOINT 100%	
Model B				Rack 1	1/30
Slot	Base	Exp.	Comment		
1	BOD16A1	*****	[		]
2	*****	*****	[		]
3	*****	*****	[		]
:	:	:	:	:	:
30	*****	*****	[		]
[TYPE]		LIST [CHOICE]		CLR_ASG	

When the cursor is positioned to column Base and F4 [CHOICE] is pressed, a menu appears. This menu contains the following items. When no unit is set, "*****" is displayed. "*****" indicates that no unit is connected.

- BMD88A1
- BID16A1
- BOD16A1
- BOA12A1

When the cursor is positioned to column Exp. and F4 [CHOICE] is pressed, a menu appears. This menu contains the following items. When no unit is set, "*****" is displayed. "*****" indicates that no unit is connected.

- BMD88P1
- BID16P1
- BOD16P1
- BIA16A1
- BMD88Q1

After a unit is set on this screen, the unit I/O can be used by turning the power off, then on again.

When the setting of a unit is changed, processing for I/O power failures is not performed at the next power-on, even when processing for power failures is enabled.

To enter a comment, press the enter key with the cursor positioned to column Comment. The comment is displayed following PRIO-100 Model B comm fault, displayed when the DI/DO unit is disconnected from the interface unit.

When SAVE is selected on this screen after FCTN key is pressed, a file named DIOCFGSV.IO is saved. This file contains the contents set on the I/O link screen. It also contains the I/O assignment, comments, and other information. Such information can be saved in this file from other I/O and file screens. F5 (CLR_ASG) is described later.

3.5.3 Number of ports Setting Screen

For I/O units such as the I/O link connection unit and 90-30PLC that cannot be used without setting the number of signals, set the number of signals on this screen.

When the cursor is positioned to "90-30PLC" on the I/O link list screen, press the F3 (DETAIL) key. Then, Number of ports setting screen appears as shown below.

I/O Link Device			JOINT 100%
			1/2
90-30 PLC	Rack 1	Slot 1	
Port name			Points
1 Digital Input:			0
2 Digital Output:			0
[TYPE]	LIST	CLR_ASG	

Move the cursor to the number indicating the number of signals and enter a numeric value to set the number of signals.

The target I/O unit can be used by turning the power off, then on again after the number of signals is set on this screen.

When the number of signals is changed, processing for I/O power failures is not performed at the next power-on, even when processing for power failures is enabled.

When SAVE is selected on this screen after FCTN key is pressed, a file named DIOCFGSV.IO is saved. This file contains the contents set on the I/O link screen. It also contains the I/O assignment, comment, and other information. Such information can be saved in this file from other I/O and file screens in the same way as normal.

Explanation of F5 (CLR_ASG)

When the number of signals is set for a model-B unit or I/O unit on the I/O link screen, the I/O assignment may differ from the standard assignment according to the setting procedure. The following

operation can set all I/O assignment to the standard settings. When setting the number of signals for a model-B unit or I/O unit for the first time, perform the following operation.

* When the unit is used with non-standard settings, this operation deletes the assignment information. Press F5 (CLR_ASG). The following message appears.

Clear all assignments?	
YES	NO
F4	F5

Press F4 (YES) to delete all assignment information. When the power to the controller is turned off, then on again, the assignment is set to the standard settings.

3.6 I/O CONNECTION FUNCTION

The I/O connection function enables the RI/DI/SI or ES status to be output to DO/RO to report the signal input status to external devices.

The standard input/output ranges are shown below:

- RI[mmm] → DO[nnn].
(1<=mmm<=8, 0<=nnn<=512)
- DI[iii] → RO[jjj].
(0<=iii<=512, 1<=jjj<=8)
- DI[kkk] → DO[lll].
(0<=kkk<=512, 0<=lll<=512)
- SI[qqq] → DO[rrr].
(0<=qqq<=15, 0<=rrr<=512)
- ES → DO[ttt].
(0<=ttt<=512)

Explanation of the function/settings

Assign signals and enable or disable each assignment on Interconnect.

The following five types of screens are available:

- DI DO connection setting screen (RI → DO)
- DI DO connection setting screen (DI → RO)
- DI DO connection setting screen (DI → DO)
- DI DO connection setting screen (SI → DO)
- DI DO connection setting screen (ES → DO)

DI DO connection setting screen (RI → DO)

Assign DO signal numbers to RI1 to RI8. Whether to enable or disable each assignment can also be set.

DI DO connection setting screen (DI → RO)

Assign RO1 to RO8 to DI signal numbers. Whether to enable or disable each assignment can also be set.

DI DO connection setting screen (DI → DO)

Assign an DO signal number to each DI number. Whether to enable or disable each assignment can also be set.

DI DO connection setting screen (SI → DO)

Assign DO signal numbers to SI0 to SI15. Whether to enable or disable each assignment can also be set.

DI DO connection setting screen (ES → DO)

Assign DO signal numbers to sixteen kinds of ES (Emergency Stop) signal. Whether to enable or disable each assignment can also be set.

List of ES singals

1	[EMGOP]	SOP Emergency Stop
2	[EMGTP]	TP Emergency Stop
3	[DEADMAN]	TP Deadman release
4	[FENCE]	Fence Open
5	[ROT]	Robot Overtravel
6	[HBK]	Hand Broken
7	[EMGEX]	External Emergency Stop
8	[PPABN]	Pneumatic Pressure Abnormal (Low Air Alarm)
9	[BELTBREAK]	Belt Broken
10	[FALM]	Fan Alarm [-- FALM is not used nowadays --]
11	[SVOFF]	SVOFF Input
12	[IMSTP]	UOP IMSTP
13	[BRKHLD]	Brake on hold
14	[USRALM]	USER Alarm
15	[SRVDSCNCT]	Servo Disconnect
16	[NTED]	Non Teacher Enabling Device

NOTE

Refer to the Controller Maintenance Manual for more information on emergency stop signals.

Example)

When "ENABLE DI[2] → RO[3]" is set, the status of DI[2] is output to RO[3].

NOTE

- 1 When DI[i] → DO[j] is set and this assignment is enabled, the status of DI[i] is output to DO[j] at regular intervals. Therefore, if the contents of DO[j] are changed using the TP or a program, the contents of DO[j] is turned back because the status of DI[i] is output to DO[j] again.
- 2 Whether to enable or disable each assignment can be changed only on the setting screen, described above.
- 3 When different multiple input signals are assigned to the same output signal, the status of each input signal is output. For example, assume that the following settings are made:
 ENABLE RI[1] → DO[1]
 ENABLE RI[2] → DO[1]
 In this case, when the status of RI[1] is ON and the status of RI[2] is OFF, the DO[1] output will be unpredictable. (DO[1] alternately indicates ON and OFF in practice.)

Procedure 3-7 Setting the I/O connection function**Step**

- 1 Press MENUS to display the screen menu.
- 2 Select "5 I/O."
- 3 Press F1, [TYPE] to display the screen switching menu.

- 4 Select Interconnect. The DI DO connection setting screen appears.

4 ALARM
5 I/O
6 SETUP

MENUS

Interconnect

[TYPE]

F1

DI DO connection setting screen(RI DO)

INTERCONNECT		JOINT 100%	
		1/8	
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]
[TYPE]		[SELECT] ENABLE DISABLE	

- 5 Press SELECT.
- 6 Position the cursor to the screen to be displayed and press the ENTER key or specify the item number of the screen to be displayed using a numeric key.

1 RI-> DO
2 DI-> RO
3 DI-> DO
4 SI-> DO
5 ES-> DO

[TYPE]

SELECT

F3

DI RO connection setting screen(DI RO)

INTERCONNECT		JOINT 100%	
		1/8	
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]
[TYPE]		[SELECT] ENABLE DISABLE	

3.7 SIMULATED INPUT SKIP FUNCTION

Overview

The robot controller provides a function whereby if a wait is performed with a wait instruction on an input signal set to a simulated status, the wait is automatically canceled when a timeout is detected.

The simulated input skip function can be used with digital input signals and robot input signals. It is possible to specify whether to enable the simulated input skip function for each signal.

On the input signal list screen, a signal for which the simulated input skip function is enabled is displayed with a U/S enclosed in parentheses (), which indicates whether the signal is in a simulated status and whether the simulated input skip function is enabled for that signal. On the screen shown in the example below, DI[1] is set up as a simulated signal and the simulated input skip function is enabled for it whereas DI[7] is not set to a simulated status but the simulated input skip function is enabled.

I/O Digital In				
	#	SIM	STATUS	
DI[1]	(S)	OFF	[	]
DI[2]	U	ON	[	]
DI[3]	U	ON	[	]
DI[4]	U	OFF	[	]
DI[5]	U	OFF	[	]
DI[6]	U	OFF	[	]
DI[7]	(U)	OFF	[	]
DI[8]	U	OFF	[	]
DI[9]	U	OFF	[	]

If the simulated input skip function is enabled, the prompt below appears before the program starts. Pressing the ENTER key causes program operation to start. This prompt appears if there is at least one input signal for which the simulated input skip function is enabled.

**The Simulated Input Skip
feature is enabled !
WAIT instructions may time
out automatically.**

[OK]

If a timeout occurs after a wait instruction, and the wait is automatically canceled, the warning message below appears on the alarm line.

PRI0-189 (Program, Line) WAIT will time out

The time after which a timeout is detected after a wait instruction can be set with the "Sim. Input Wait Delay" item on the system configuration menu. If this setting is changed, the change will be applied immediately.

It is possible to monitor to see if there are any input signals for which the simulated input skip function is enabled and output them as output signals. For the "Set if Sim. Skip Enabled" item on the system configuration menu, set the number of the output signal that will turn on if the simulated input skip function is enabled. To make the setting effective, turn off the power and then back on.

WARNING

Setting an input signal to a simulated status and using the simulated input skip function should only be temporary during test operation. Never do so during production line operation.

By selecting the "UNSIM ALL I/O" item on the function menu, it is possible to release all signals from a simulated status.

By setting up "Set if INPUT SIMULATED" on the system configuration menu, it is possible to monitor to see if there are any input signals that are set to a simulated status and output them as output signals. For this item, set the number of the output signal that will turn on if one of digital, group, robot, and analog signals is set to a simulated status. To make the setting effective, turn off the power and then back on.

Procedure for setting up the simulated input skip function

For an input signal to be skipped if in a simulated status, enable the simulated input skip function.

- Step

1. Press the "MENU" key.
2. Select the "I/O" item.
3. Press the F1 [TYPE] key.
4. Select the "Digital" or "Robot" item.
5. If output signals are displayed, press F3 "IN/OUT" to switch to the input signal list screen.
6. Position the cursor on the signal for which the simulated input skip function is to be enabled.
7. Press the "NEXT" key and then the F3 "DETAIL" key.
8. On the input signal detail screen, position the cursor on "Skip when simulated".
9. Press the F4 "TRUE" key.

3.8 SETTING AUTOMATIC OPERATION

Automatic operation is the function with which the remote controller starts a program, using the peripheral I/O. The automatic operation includes the following functions:

- The robot service request (RSR) function selects and starts a program according to the robot service request signals (RSR1 to RSR8 inputs). When another program is being executed or is temporarily stopped, the selected program enters the wait state and is started once the program currently being executed terminates.
- The program number selection (PNS) function selects or examines a program, using the program number selection signals (PNS1 to PNS8 PNSTROBF) and the START signal.
While a program is temporarily stopped or being executed, these signals are ignored.
- The automatic operation start signal (PROD_START input) starts the currently selected program from line 1. When another program is temporarily stopped or is being executed, this signal is ignored.
- The cycle stop signal (CSTOPI input) is used to terminate the program currently being executed.
 - When FALSE is selected for CSTOPI for ABORT on the system configuration menu, this signal terminates the program currently being executed once the execution is complete. It also releases programs from the wait state by RSR. (Default)
 - When TRUE is selected for CSTOPI for ABORT on the system configuration menu, this signal forcibly terminates the program currently being executed immediately. It also releases (Clear) programs from the wait state by RSR.
- The external start signal (START input) is used to start a program that is temporarily stopped.
 - When FALSE is selected for START for CONTINUE only on the system configuration menu, this signal starts the currently selected program from the current line. This signal also starts a temporarily stopped program. (Default)
 - When TRUE is selected for START for CONTINUE only on the system configuration menu, this signal starts only a temporarily stopped program. When no program is temporarily stopped, this signal is ignored.

A program can be started by entering the peripheral I/O only when the robot is in the remote state. The remote state is established when the following remote conditions are satisfied:

- The teach pendant enable switch is off.
- The remote signal (SI[2]) is on. (For how to turn the remote signal (SI[2]) on and off, see the description of Remote/Local setup in Section 3.15, "SYSTEM CONFIG MENU.")
- The *SFSPD signal of the peripheral I/O is set on.
- The ENBL signal of the peripheral I/O is set on.
- System variable \$RMT_MASTER is set to 0 (peripheral equipment).

NOTE

The value of \$RMT_MASTER can be set to 0 (peripheral equipment), 1 (CRT/KB), 2 (host computer), or 3 (no remote equipment).

A program including a motion (group) can be started when the following ready conditions are satisfied:

- The ENBL input signal of the peripheral I/O is set on.
- The servo power is turned on (not in the alarm state).

The CMDENBL signal indicates whether the above conditions are satisfied. The CMDENBL signal is output when the following conditions are satisfied:

- The remote conditions are satisfied.
- The ready conditions are satisfied.
- The continuous operation mode is selected (the single step mode is disabled).

NOTE

If TRUE is specified at "START for CONTINUE only" on the system configuration screen, the START signal is effective for only a program on hold.

3.8.1 Robot Service Request (RSR)

The robot service request (RSR) starts a program from an external device. The eight robot service request signals (RSR1 to RSR8) are used for this function.

- 1 The controller uses the RSR1 to RSR8 inputs to determine whether the input RSR signal is enabled. When the signal is disabled, it is ignored.
Whether to enable or disable RSR1 to RSR8 is set in system variables \$RSR1 to \$RSR8 and can be changed on the RSR setting screen or by using the program RSR instruction.

NOTE

If the peripheral device input signal (UI) is disabled, select TRUE for Enable UI signals on the system configuration screen.

- 2 Eight RSR registration numbers can be registered for RSR. The value obtained by adding a base number to an RSR registration number is used as the program number (four digits). For example, when RSR2 is input, the following value is used as the program number:

$$(\text{Program number}) = (\text{RSR2 registration number}) + (\text{base number})$$

The selected program is named as follows:

$$\text{RSR} + (\text{program number})$$

NOTE

Specify the name of a program for automatic operation in "RSR" + (program number) format. Enter a 4-digit number such as RSR0121m, not RSR121. If not, the robot will not operate.

The base number is set in \$SHELL_CFG.\$JOB_BASE and can be changed using Base number on the RSR setting screen or a program parameter instruction.

- 3 A pulse of the RSR acknowledgment output (ACK1 to ACK8) corresponding to the RSR1 to RSR8 input is output. When the ACK1 to ACK8 signal is output, the controller accepts another RSR input.
- 4 When a program is in the terminated state, the selected program is started. When another program is being executed or is temporarily stopped, the request (job) is entered into the queue and the selected program is started when the program being executed terminates.
Jobs (RSR programs) are executed in the order in which they are entered into the queue.
- 5 Waiting programs are canceled (cleared) by the cycle stop signal (CSTOPI input) or upon forced program termination.

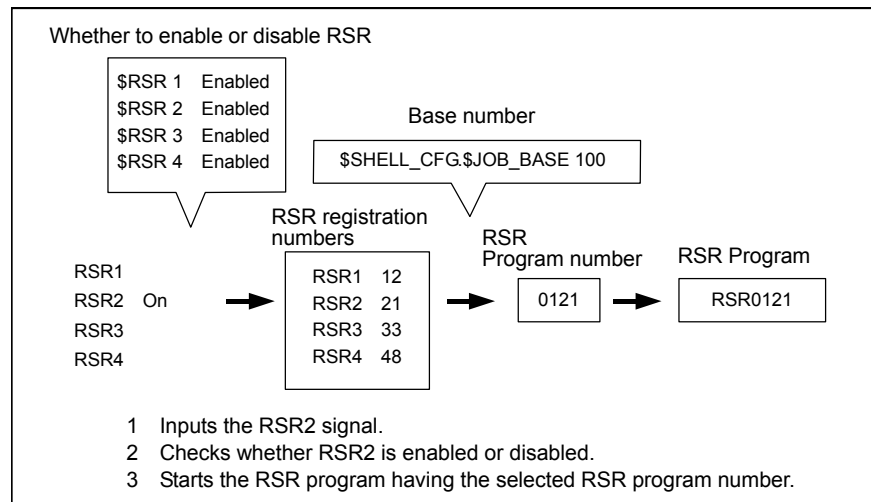


Fig. 3.8.1 (a) Robot Service Request

Starting a program by RSR is enabled in the remote state.

Starting a program involving operation (group) by RSR is enabled when the operation enable conditions as well as the remote conditions are satisfied.

The CMDENBL output is provided to indicate whether the above conditions are satisfied.

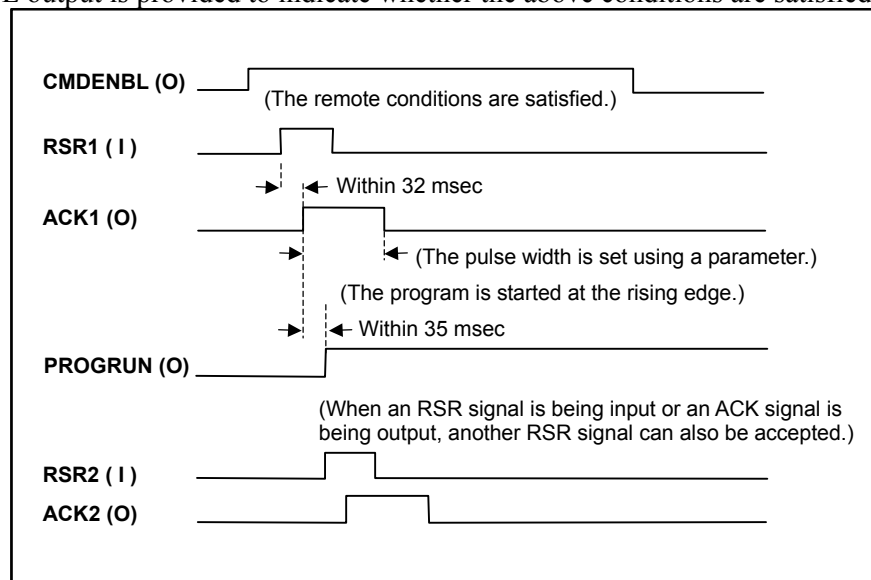


Fig. 3.8.1 (b) Sequence of Automatic Operation by RSR

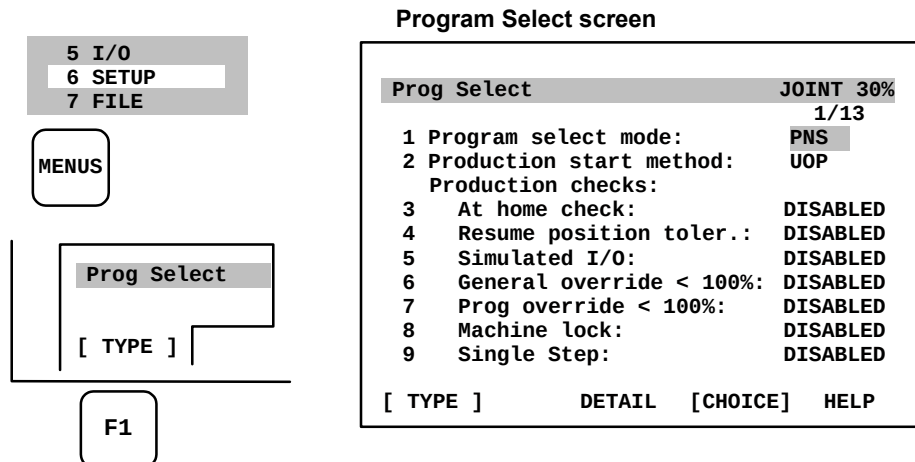
Set RSR for SETUP Prog Select on the RSR setting screen.

Table 3.8.1 RSR Setting Items

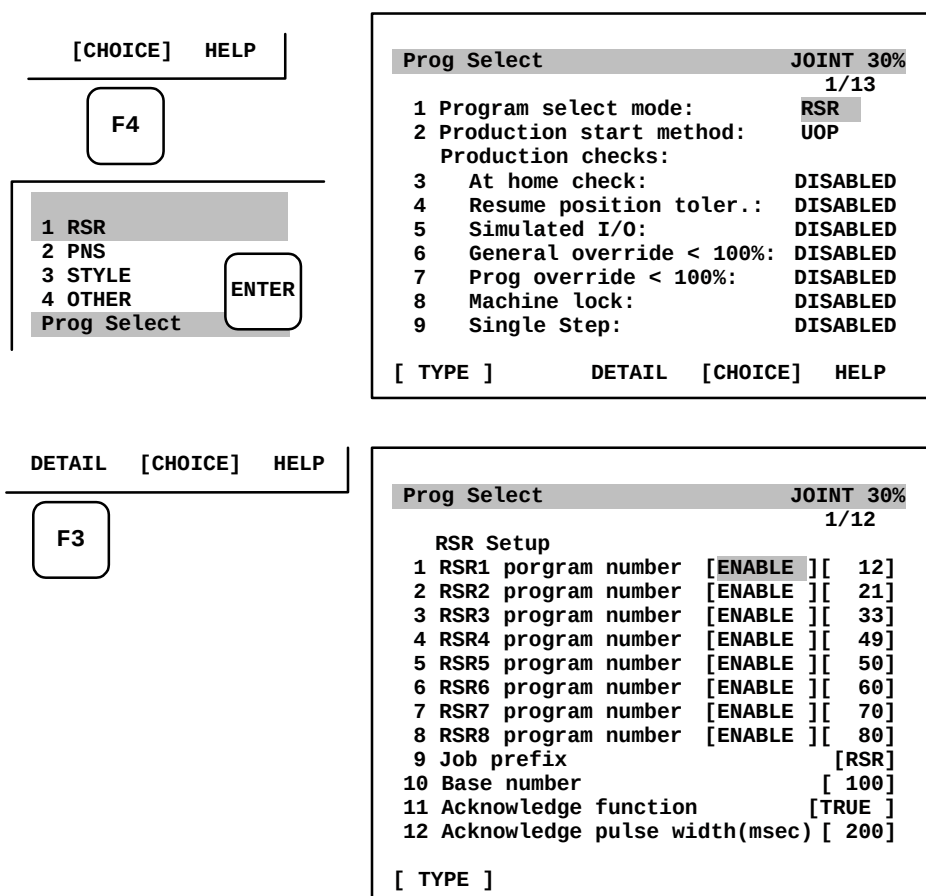
Item	Description
RSR1 to 8 program number	Specifies whether to enable or disable RSR1 to RSR8 and the RSR registration numbers. When an RSR signal is disabled and the specified signal is input, the program is not started. Setting whether to enable or disable each RSR is stored in system variable \$RSR1 to \$RSR8.
Job prefix	Top character string of the name of the program to be started. By default, it is set to "RSR".
Base number	Added to the RSR registration number to obtain the RSR program number.
Acknowledge function	Sets whether to output RSR acknowledgment signals (ACK1 to ACK8).
Acknowledge pulse width	Sets the pulse output period (unit: msec) when the output of each RSR acknowledgment signal (ACK1 to ACK8) is enabled.

Procedure 3-8 Setting RSR**Step**

- 1 Press MENUS to display the screen menu.
- 2 Select SETUP.
- 3 Press F1, [TYPE] to display the screen switching menu.
- 4 Select "Prog Select." The Prog Select screen appears.



- 5 Position the cursor to "Program select mode". Press F4 [CHOICE] and select RSR, then press F3 DETAIL.(Prog Select screen →3.8.4)



- 6 Position the cursor to the target item and enter a value.

- 7 After changing Program select mode, to enable the change, turn the power off, then on again.

**WARNING**

After the type of automatic operation function is changed, the power to the controller must be turned off, then on again to enable the change. If not, the setting is not accepted.

3.8.2 Program Number Selection (PNS)

The remote controller uses the program number selection (PNS) function to select or collate a program. Specify a desired PNS program number with the input signals, PNS1 to PNS8.

Step

- 1 The controller reads the PNS1 to PNS8 input signals as a binary number by the PNSTROBE pulse input. When a program is being executed or is temporarily stopped, these signals are ignored. When the PNSTROBE pulse input is on, the selection of a program from the teach pendant is disabled.

NOTE

If the peripheral device input signal (UI) is disabled, select TRUE for Enable UI signals on the system configuration screen.

- 2 The data of signals PNS1 to PNS8 is converted into a decimal PNS number. The sum of the PNS number and the reference number is a PNS program number (four digits).

(Program number)=(PNS number)+(Base number)

The specified PNS+(Program number) program number is named as follows.

NOTE

Specify the name of a program for automatic operation in "PNS" + (program number) format. Enter a 4-digit number such as PNS0138, not PNS138. If not, the robot will not operate.

When zero is input by the PNS1 to PNS8 inputs, or the program designated by the PNS1 to PNS8 inputs does not exist, even if the PNSTROBE signal is input, nothing is done. When the START signal is input after that, the program selected on the teach pendant is started. If no program is selected at that time, nothing is done. (In case that the program designated by PNS inputs does not exist, if the PNSTROBE signal is input, the warning is displayed. And, in case that no program is selected on the teach pendant, when the START signal is input, the warning is displayed.)

NOTE

In case that zero is input by the PNS1 to PNS8 inputs and the PNSTROBE signal is input, the behavior of the controller is decided according to the UOP allocation type. The above description is for the "Simple allocation". In case that the UOP allocation type is "All allocation", when zero is input by the PNS1 to PNS8 inputs and the PNSTROBE signal is input, the system enters the status in which no program is selected on the teach pendant.

The base number is set in \$SHELL_CFG.\$JOB_BASE and can be changed using Base number on the PNS setting screen or a program parameter instruction.

- 3 SNO1 to SNO8 are output to indicate a PNS number as a binary code as confirmation. An SNACK pulse is output simultaneously. If the PNS number cannot be represented as an 8-bit numeric value, SNO1 to SNO8 output a zero.

4 The remote controller checks that the SNO1 to SNO8 output value is the same as the PNS1 to PNS8 input value when SNACK is output, and sends the automatic operation start input (PROD_START).

5 The controller receives the PROD_START input and starts the program.

Starting a program by PNS is enabled in the remote state.

Starting a program involving an operation (group) is enabled when the operation enable conditions as well as the remote conditions are satisfied.

The CMDENBL output is provided to indicate whether the above conditions are satisfied.

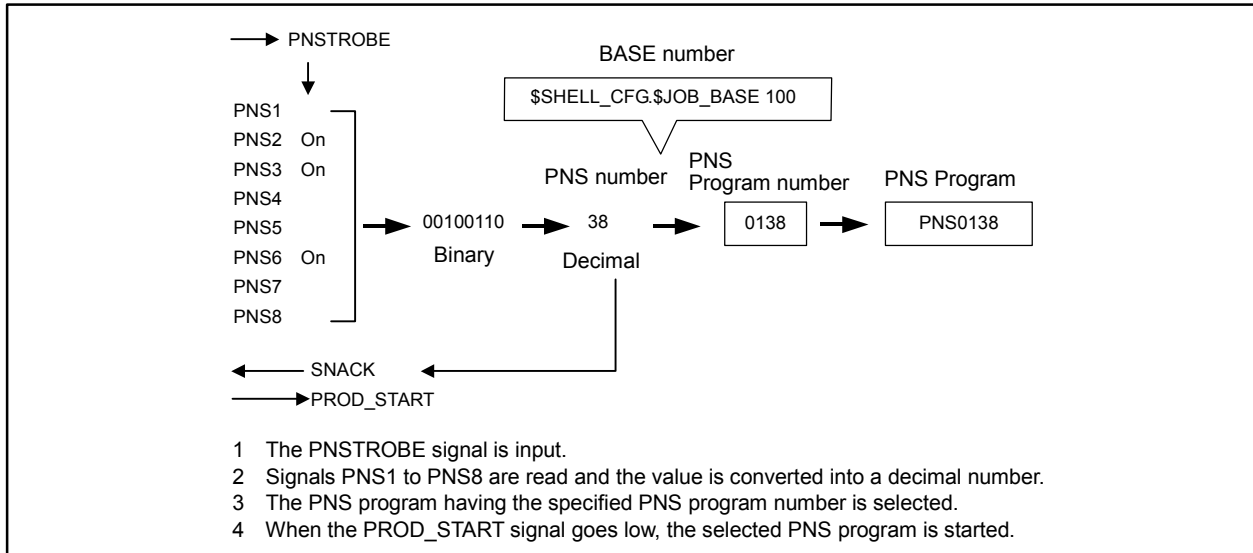


Fig. 3.8.2 (a) Program Number Selection

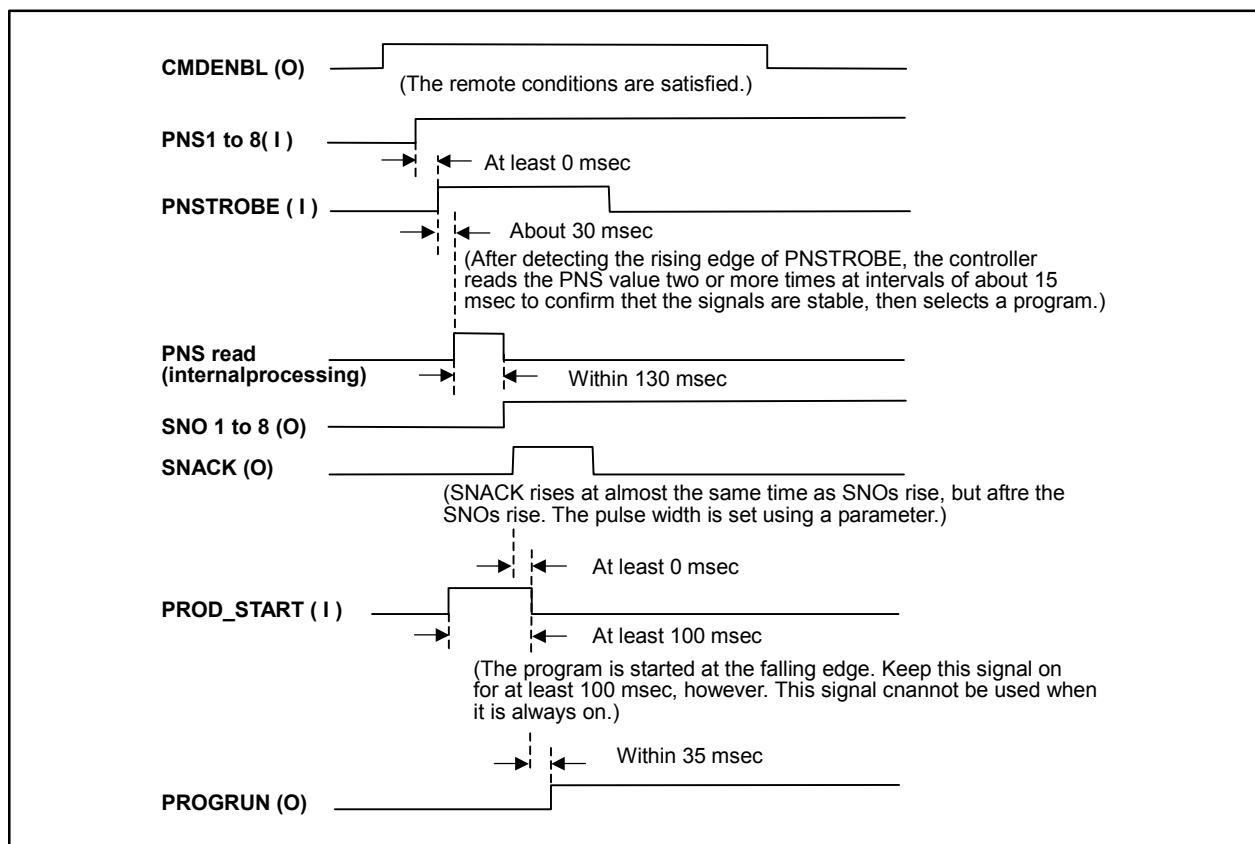


Fig. 3.8.2 (b) Sequence of Automatic Operation by PNS

Set the PNS function on the PNS setting screen [6 (SETUP). Prog Select]. Refer to Table 3.8.2.

Table 3.8.2 Setting the PNS function

Items	Descriptions
Job prefix	Top character string of the name of the program selected. By default, it is set to "PNS".
Base number	The reference number is added to the PNS number to obtain a PNS program number.
Acknowledge pulse width (msec)	Sets the pulse output period (unit: msec) of the PNS acknowledgment signal (SNACK).

Procedure 3-9 Setting the PNS function

Step

- 1 Press the MENUS key. The screen menu is displayed.
- 2 Select "6 (SETUP)."
- 3 Press the F1 key, TYPE. The screen change menu is displayed.
- 4 Select "Prog Select". Prog Select screen is displayed.

5 I/O
6 SETUP
7 FILE

MENUS

Prog Select

[TYPE]

F1

Program Select screen

Prog Select		JOINT 30%
		1/13
1 Program select mode:	RSR	
2 Production start method:	UOP	
Production checks:		
3 At home check:	DISABLED	
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:	DISABLED	
[TYPE]		DETAIL [CHOICE] HELP

- 5 Position the cursor to "Program select mode". Press F4 [CHOICE] and select PNS, then press F3 DETAIL.(Prog Select screen →3.8.4)

[CHOICE] HELP

F4

1 RSR
2 PNS
3 STYLE
4 OTHER
Prog Select

ENTER

Prog Select

Prog Select		JOINT 30%
		1/13
1 Program select mode:	PNS	
2 Production start method:	UOP	
Production checks:		
3 At home check:	DISABLED	
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:	DISABLED	
[TYPE]		DETAIL [CHOICE] HELP

DETAIL	[CHOICE]	HELP
F3		
Prog Select		JOINT 30% 1/3
PNS Setup		
1 Job prefix	[PNS]	
2 Base number	[100]	
3 Acknowledge pulse width(msec)	[200]	
[TYPE]		

- 6 Place the cursor on a desired field and enter a value.
- 7 After changing RSR to PNS, to enable the change, turn the power off, then on again.

**WARNING**

After the type of automatic operation function is changed, the power to the controller must be turned off, then on again to enable the change. If not, the setting is not accepted.

3.8.3 STYLE

The remote controller uses STYLE function to select or collate a program. Specify a desired STYLE program number with the input signals, STYLE1 to STYLE8.

Step

- 1 STYLE function needs programs to be set to each STYLE number in advance. Then program name is not restricted as PNS and RSR.
- 2 The controller reads the STYLE1 to STYLE8 input signals as a binary number. And the data of signals STYLE1 to STYLE8 is converted into a decimal STYLE number.
- 3 From the remote controller, the start input (START or PROD_START) is sent. Then program is selected by STYLE number, and selected program is started.
- 4 SNO1 to SNO8 are output to indicate a STYLE number as a binary code as confirmation. An SNACK pulse is output simultaneously.
- 5 If the program is pausing, and the start input (START or PROD_START) is sent, no program selection is made and execution is resumed.

NOTE

If the peripheral device input signal (UI) is disabled, select TRUE for Enable UI signals on the system configuration screen.

Starting a program by STYLE is enabled in the remote state.

Starting a program involving an operation (group) is enabled when the operation enable conditions as well as the remote conditions are satisfied.

The CMDENBL output is provided to indicate whether the above conditions are satisfied.

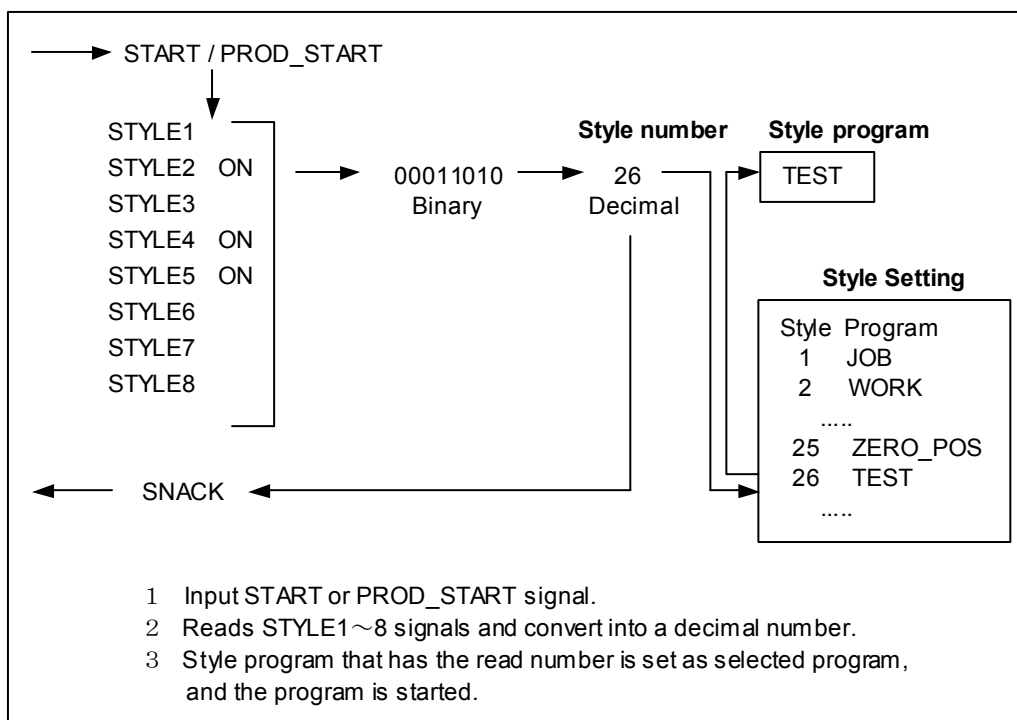


Fig. 3.8.3 (a) STYLE

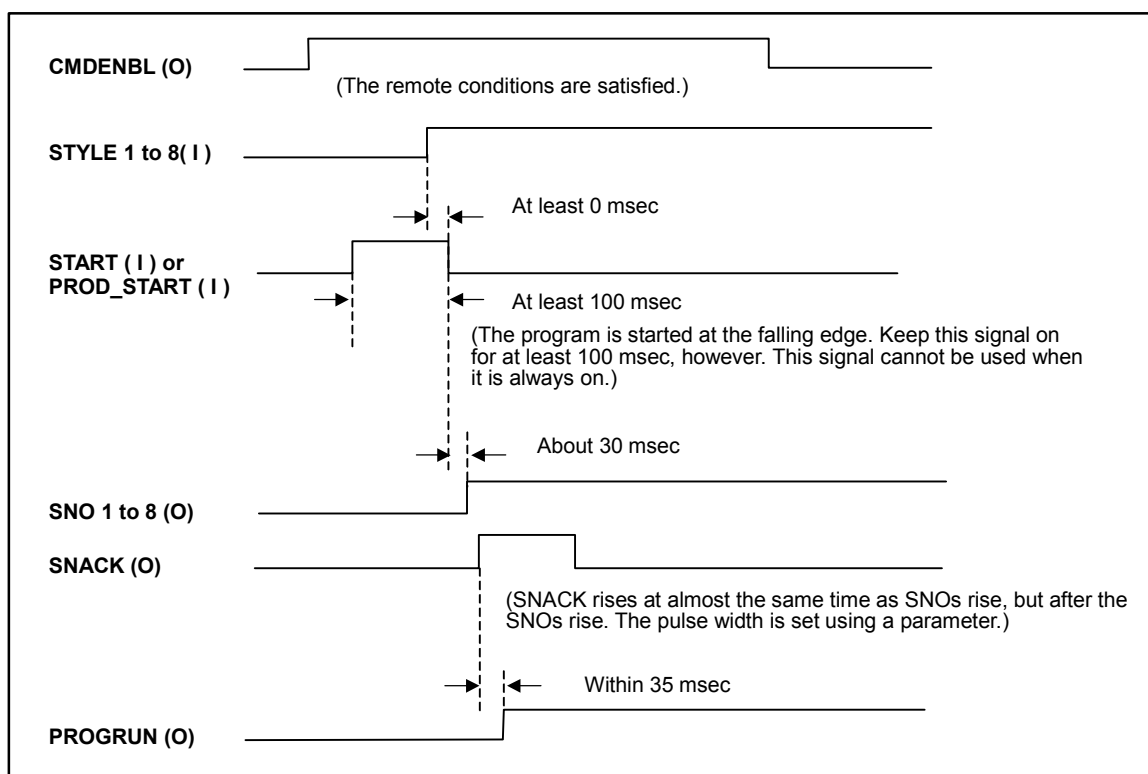


Fig. 3.8.3 (b) Sequence of Automatic Operation by STYLE

Setting the STYLE function

Set the STYLE function on the Prog Select screen [6 (SETUP). Prog Select]. Refer to Table 3.8.3.

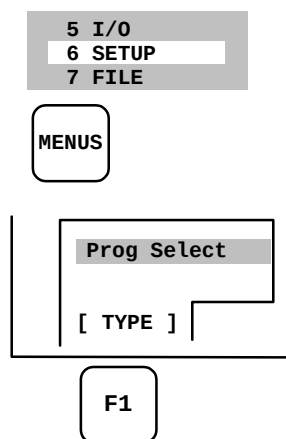
Table 3.8.3 Setting the STYLE function

Items	Descriptions
Acknowledge function	Sets whether to output STYLE acknowledgment signals (SNO1 to SNO8 and SNACK). This is disabled as initial setting.
Acknowledge pulse width (msec)	Sets the pulse output period (unit: msec) of the PNS acknowledgment signal (SNACK).
Max number in style table	Sets the max number which is used for STYLE.

Procedure 3-10 Setting the STYLE function

Step

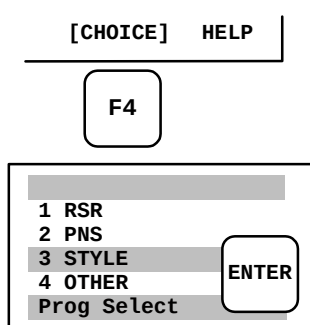
- 1 Press the MENUS key. The screen menu is displayed.
- 2 Select "6 (SETUP)."
- 3 Press the F1 key, TYPE. The screen change menu is displayed.
- 4 Select "Prog Select". Prog Select screen is displayed.



Program Select screen

Prog Select		JOINT 30%
		1/13
1 Program select mode:		RSR
2 Production start method:		UOP
Production checks:		
3 At home check:		DISABLED
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:		DISABLED
[TYPE]		DETAIL [CHOICE] HELP

- 5 Position the cursor to "Program select mode". Press F4 [CHOICE] and select STYLE, then press F3 DETAIL. (Prog Select screen → 3.8.4)



Prog Select		JOINT 30%
		1/13
1 Program select mode:		STYLE
2 Production start method:		UOP
Production checks:		
3 At home check:		DISABLED
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:		DISABLED
[TYPE]		DETAIL [CHOICE] HELP

DETAIL	[CHOICE]	HELP
F3		

Prog Select				JOINT 30%
Style Table Setup				1/32
Style	ProgName	Valid	Comment	
1	JOB	YES		
2	WORK	YES		
3	*****>	YES		
4	*****>	YES		
5	*****>	YES		
6	*****>	YES		
7	*****>	YES		
8	*****>	YES		
9	*****>	YES		
[TYPE] CLEAR CONFIG [CHOICE]				

- 6 By placing the cursor on a desired field, pressing F4 [CHOICE] and selecting program, set STYLE program.
- 7 By pressing F3 CONFIG additionally, acknowledge signal can be set. (This is disabled as initial setting.)

CLEAR	CONFIG	[CHOICE]
F3		

Prog Select		JOINT 30%
Style Table Setup		1/3
1 Acknowledge function: [FALSE]		
2 Acknowledge pulse width(msec): [400]		
3 Max number in style table: [32]		
[TYPE]		

- 8 After changing setting from RSR/PNS/OTHER to STYLE, to enable the change, turn the power off, then on again.

**WARNING**

After the type of automatic operation function is changed, the power to the controller must be turned off, then on again to enable the change. If not, the setting is not accepted.

3.8.4 Prog Select Screen

Overview

On the "Prog Select" screen, the following can be performed:

- As program selection methods, it is now possible to select , PNS, RSR, STYLE, and OTHER .
- As program start methods, it is now possible to select UOP and OTHER.
- Various checks are performed at a program start or resumption.

Prog Select screen

Prog Select	
	1/14
1 Program select mode:	PNS
2 Production start method:	UOP
Production checks:	
3 At home check:	ENABLED
4 Resume position toler.:	ENABLED
5 Simulated I/O:	DISABLED
6 General override < 100%:	DISABLED
7 Prog override < 100%:	DISABLED
8 Machine lock:	DISABLED
9 Single step:	DISABLED
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
[TYPE]	HELP

Program selection methods

It is now possible to select PNS, RSR, STYLE, and OTHER.

(If the UOP allocation type is "Simple allocation", RSR cannot be selected.)

- RSR
See Subsection 3.8.1, "Robot Service Request (RSR)."
- PNS
See Subsection 3.8.2, "Program Number Selection (PNS)."
- STYLE
See Subsection 3.8.3, "STYLE."
- OTHER
The program specified for the system variable \$SHELL_WRK. \$CUST_NAME is selected.
A program selection is made when a program is started with the method specified by following "Automatic operation start methods", described later.
If the program is pausing, no program selection is made and execution is restarted.
Usually, do not use this function because it is for special purposes.

Automatic operation start methods

If the program selection method is "STYLE" or "OTHER", it is possible to select a program start method from UOP, and OTHER.

- UOP
A program is started with UI[6:START] or UI[18:PROD_START].
- OTHER
A program is started by changing the system variable \$SHELL_WRK. \$CUST_START from FALSE to TRUE.
This method cannot be selected if the program selection method is either RSR or PNS.
Usually, do not use this function because it is for special purposes.

- * If the program selection method is either STYLE or OTHER and the automatic operation start method is UOP, it is possible to select and start a program with either UI[6:START] or UI[18:PROD_START], provided that a program is terminated. In this case, the program is started at the first line.
If a program is pausing, the program can be resumed with either UI[6:START] or UI[18:PROD_START]. In this case, no program selection is made.

Note that if the program selection method is either RSR or PNS, the effects of UI[6:START] and UI[18:PROD_START] are the same as those in the past.

List of program selection methods and automatic operation start methods

		Automatic operation start method	
		UOP	Other
Program selection method	RSR	Program selection/start : RSR 1 to 8	Not available.
	PNS	Program selection : PNS1 to 8, PNSTROBE Program start : UI[18:PROD_START] or UI[6:START]	Not available.
	Style	Program selection:STYLE1 to 8 Program start : UI[18:PROD_START] or UI[6:START]	Program selection:STYLE1 to 8 Program start :\$SHELL_WRK.\$CUST_START
	Other	Program selection: \$SHELL_WRK.\$CUST_NAME Program start : UI[18:PROD_START] or UI[6:START]	Program selection: \$SHELL_WRK.\$CUST_NAME Program start :\$SHELL_WRK.\$CUST_START

Automatic operation check

It is possible to specify whether to enable or disable each of the automatic operation check items on the Prog Select screen.

Check item	Explanation
At home check	Checks to see if the robot is in its home position. Home position refers to the reference position for which "Is a valid HOME" is set to ENABLE on the Reference Position Set up screen (MENUS → "SETUP" → F1 [TYPE] → "Ref. Position"). If At home check is enabled, "Is a valid HOME" must be set to ENABLE for at least one reference position of group 1. REF POSN Reference Position1/13 Ref.Position Number:1 1 Comment:[*****] 2 Enable/Disable:DISABLE 3 Is a valid HOME:ENABLE 4 Signal definition:DO[0] 5 J1:0.000 +/-0.000 : If "Is a valid HOME" is set to ENABLE on the Reference Position Set up screen, the "HOME_IO" program will be started when the robot reaches that position. If not using "HOME_IO", delete all contents of the "HOME_IO" program. NOTE The "HOME_IO" program is configured not to accept a forced termination request so that it can always be executed to the end.
Resume position toler.	Checks to see if the robot is near the position at which the program paused.
Simulated I/O	Suppresses a program start/resumption if I/O is simulated.
General override < 100%	Suppresses a program start/resumption if the general override is less than 100%.
Prog override < 100%	Suppresses a program start/resumption if \$MCR_GRP[.].\$PRGOVERRIDE is less than 100.
Machine lock	Suppresses a program start/resumption if robot operation is disabled.
Single step	Suppresses a program start/resumption in case of a single step.
Process ready	Enables the user to check whether to make a program start/resumption depending on the status of the cooling machine, cooling water, welding transformer, etc. (The process ready conditions differ from one application to another). Selections are displayed allowing the user to continue the program, recheck the I/O status, and stop the program.

By positioning the cursor on each item and pressing F3 (DETAIL), it is possible to make detailed settings for that check item.

For some check items, it is not possible to make some of the settings on the Detailed Setup screen.

- * It is not possible to specify whether to enable or disable the Resume position toler. check item on the Prog Select screen. Specify this on the Resume tolerance check screen (MENUS → "SETUP" → F1 [TYPE] → "Resume Tol.").

Automatic operation check detailed setting screen (At home check as an example)

Prog Select DETAIL	
	1/3
Check: At home check	
1 Check when run:	ENABLED
Check when resume:	DISABLED
2 Prompt if failure:	DISABLED
Post error if failure:	ENABLED
3 Post warning if forced:	DISABLED
Force condition:	DISABLED
[TYPE]	ENABLED DISABLED

Detailed setting	Explanation
Check when run	Checks specified items at a program start. For the Resume position toler. check, this item cannot be enabled.
Check when resume	Checks specified items at a program resumption. For the At home check, this item cannot be enabled.
Prompt if failure	Displays a prompt message on the screen if Check when run or Check when resume is set to ENABLED and the check causes a program start or resumption to be interrupted. At this time, it is possible to select whether to continue or stop the program. The text of the prompt message differs depending on the check item.
Post error if failure	Generates the alarms below if Check when run or Check when resume is set to ENABLED and the check causes a program start or restart to be interrupted. "SYST-011 Failed to run task" "SYST-079 Startup check failed"
Post warning if forced	Effective only if Force condition is enabled. If enabled, this item displays a warning in the event of Force condition. The text of the warning differs depending on the check item.
Force condition	If Check when run or Check when resume is set to ENABLED, this item causes that check item to be forcibly satisfied. This item takes precedence over all other detailed settings. For the check items of At home check, Resume position toler., and Machine lock, it is not possible to set Force condition to ENABLED. The action performed with Force condition differs depending on the check item.

The details of Prompt if failure, Force condition, and Post warning if forced for each check item are as follows:

- At home check

Condition	Action in the event of an error
Prompt if failure	The prompt below appears. Robot arm is not at home. Jog or move robot to a home position. [OK]
Force condition	
Post warning if forced	

- Resume position toler.

Condition	Action in the event of an error
Prompt if failure	The prompt below appears. Robot arm moved too far. ANSWERING CONTINUE WILL INITIATE MOTION. CONTINUE [STOP] When "CONTINUE" is selected, the program is executed with the warning "SYST-104 Resume tolerance ignored".
Force condition	
Post warning if forced	

- Simulated I/O

Condition	Action in the event of an error
Prompt if failure	The prompt below appears. Simulated I/O ports exist (set from the I/O screens) CONTINUE FORCE [CANCEL] • When "CONTINUE" is selected, the program continues to run. • If "FORCE" is selected, simulated I/O is forcibly canceled and the program is executed. • When "CANCEL" is selected, the program does not start but stops.
Force condition	Simulated I/O is automatically forcibly canceled and the program runs.
Post warning if forced	Simulated I/O is automatically forcibly canceled and the program runs, and the message below appears. "SYST-084 I/O forced unsimulated"

- General override < 100%

Condition	Action in the event of an error
Prompt if failure	The prompt below appears. General override (set from teach pendant hardkeys) is less than 100% CONTINUE FORCE [STOP] • When "CONTINUE" is selected, the program continues to run. • When "FORCE" is selected, the program runs at an override of 100%. • When "STOP" is selected, the program does not start but stops.
Force condition	The program automatically runs at an override of 100%.

- Prog override < 100%

Condition	Action in the event of an error
Prompt if failure	The prompt below appears. Program override is less than 100% CONTINUE FORC [STOP] • When "CONTINUE" is selected, the program continues to run. • When "FORCE" is selected, the program runs by setting \$MCR_GRP[].\$PRGOVERRIDE to 100. • When "STOP" is selected, the program does not start but stops.

Condition	Action in the event of an error
Force condition	The program automatically runs by setting \$MCR_GRP[].\$PRGOVERRIDE to 100.
Post warning if forced	The program automatically runs by setting \$MCR_GRP[].\$PRGOVERRIDE to 100, and the message below appears. "SYST-088 Prog override forced to 100%"

- Machine lock

Condition	Action in the event of an error
Prompt if failure	The prompt below appears. Motion is disabled CONTINUE [STOP] - When "CONTINUE" is selected, the program starts with the warning below. "SYST-108 Machine lock ignored" - When "STOP" is selected, , the program does not start but stops.
Force condition	
Post warning if forced	

- Single step

Condition	Action in the event of an error
Prompt if failure	The prompt below appears. Single step is enabled (set from STEP TP key) CONTINUE RECHECK [STOP] - When "CONTINUE" is selected, the program starts with the warning "SYST-109 Single step ignored". - When "RECHECK" is selected, the single step check is performed again, and in case of other than a single step, the program is started. - When "STOP" is selected, the program does not start but stops.
Force condition	A program start is made in the same way as when "CONTINUE" is selected and, at the same time, the single step is automatically canceled.
Post warning if forced	Single step is automatically canceled with the warning below, and the program start. "SYST-092 Single step forced off"

- Process ready

Condition	Action in the event of an error
Prompt if failure	The prompt below appears. Application process error. Please check process peripheral equipment. CONTINUE RECHECK [STOP] - When "CONTINUE" is selected, the program starts with the warning "SYST-110 Process ready ignored". - When "RECHECK" is selected, the process ready check is performed again, and if the conditions are met, the program is started. - When "STOP" is selected, the program does not start but stops.
Force condition	
Post warning if forced	

General settings

It is possible to make general settings for program selections and starts.

- **Heartbeat timing :**
Enables the user to specify the output cycle of the heartbeat signal for cell output.
The heartbeat signal is an output signal that switches between ON and OFF at "n" millisecond intervals. PLC uses this signal to check that the robot is operating normally. This item specifies a heartbeat signal ON/OFF switching interval. An output signal can be allocated using the I/O cell output menu. If the timing is zero or if no input is allocated, the heartbeat signal is disabled.
- **Low TEMP DRAM memory :**
Enables the user to check the minimum temporary DRAM memory. If the memory is less than the minimum memory, a warning appears.
- **Low PERM CMOS memory :**
Enables the user to check the minimum permanent CMOS memory. If the memory is less than the minimum memory, a warning appears.

3.8.5 Cell Interface I/O

Cell interface I/O

Cell interface I/O signals are used for communication between the robot and the cell controller (PLC).

Cell interface input signals

Cell interface input signals are explained in Table 3.8.5 (a).

To configure cell interface I/O, use Procedure 3-11.

Table 3.8.5 (a) Cell interface input signals

input signal	Explanation
Tryout Mode (Effective to the material handling/gripper option only)	This signal is to be allocated if tryout mode is used. The robot enters tryout mode under the conditions below. Signal = ON TP disabled \$shell_wrk.\$isol_mode = FALSE SI[REMOTE]=ON If the signal is allocated and TP is disabled, it is possible to change tryout mode from the soft panel.

Cell interface output signals

By using the cell interface output screen, it is possible to perform the following:

- Display the status of an output signal.
- Set an output signal to a simulated status.
- Forcibly send an output signal.
- Allocate an output signal.

Cell interface output signals are explained in Table 3.8.5 (b).

To configure cell interface I/O, use Procedure 3-11.

Table 3.8.5 (b) Cell interface output signals

Output signal	Explanation
Input Simulated	This output is used to notify PLC that there is a simulated input signal.
Output Simulated	This output is used to notify PLC that there is a simulated output signal.
VERRIDE=100	This output is used to notify PLC that the override for the teach pendant is at 100%.

Output signal	Explanation
Robot ready	This output is used to notify 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 to a production start. The conditions contained in the signal must be previously set with "Status screen/robot ready". It is possible to check the status of each condition on this status screen.
Tryout Status	Used to notify PLC of the tryout mode status. ON = Tryout mode enabled
Heartbeat	This signal switches between ON and OFF after each heartbeat signal cycle. This signal is used to check the status of communication between the robot and PLC.
MH Fault (*)	If the controller is not in tryout mode, this signal turns ON if an alarm of the material handling function is generated. FAULT RESET causes this output to turn OFF.
MH Alert (*)	This output turns ON if a disabled alarm is generated within 20 cycles. FAULT RESET causes this output to turn OFF.
Refpos1[n]	If REF POS is enabled, this output turns ON if the robot is at reference position n of operation group 1.

NOTE

The item marked with an asterisk (*) is effective to the material handling/gripper option only.

Procedure 3-11 Configuring cell interface I/O**Step**

- 1 Press the MENUS key and select [I/O].
- 2 Press F1 [TYPE].
- 3 Select [Cell Interface]. The cell input screen or cell output screen appears. The cell input screen is shown below as an example. The display contents differ depending on the program start method.

I/O Cell Inputs		JOINT 50%	
		1/7	
INPUT SIGNAL	TYPE #	SIM	STATUS
1 Tryout Mode	DI[0]	U	***
[TYPE] CONFIG IN/OUT SIM UNSIM >			

To switch between the input screen and the output screen, press F3 [IN/OUT]. The cell output screen is shown below. The display contents differ depending on the program start method.

I/O Cell Outputs		JOINT 50%	
		1/21	
OUTPUT SIGNAL	TYPE #	SIM	STATUS
1 Input Simulated	DO[0]	U	***
2 OVERRIDE = 100	DO[0]	U	***
3 Robot ready	DO[0]	U	***
4 Tryout Status	DO[0]	U	***
5 Heartbeat	DO[0]	U	***
6 MH Fault	DO[0]	U	***
7 MH Alert	DO[0]	U	***
[TYPE] CONFIG IN/OUT SIM UNSIM >			

To specify whether to place an I/O signal in a simulated status, position the cursor on the SIM field for that I/O signal.

- 4 To specify whether to place an I/O signal in a simulated status, position the cursor on the SIM field for that I/O signal.
 - To place the signal in a simulated status, press F4 [SIM]. The signal is placed in a simulated status.
 - To release a signal from a simulated status, press F5 [UNSIM]. The signal is released from a simulated status.
- 5 To forcibly turn an I/O signal ON or OFF, position the cursor on the STATUS field for that I/O signal.
 - To turn the I/O signal ON, press F4 [ON].
 - To turn the I/O signal OFF, press F5 [OFF].
- 6 To allocate the signal, press F2 [CONFIG]. The screen below appears.

I/O Cell Outputs		JOINT 50%
Output Signal Details		1/2
Signal name:	Style Ack	
1 Output type/no:	D0[0]	
[TYPE] PREV_IO NEXT-IO [CHOICE] VERIFY		

**CAUTION**

If \$SHELL_CFG.\$SET_IOCMT = TRUE, and a signal number is input on this input screen or output screen, the comments on the corresponding signal on the I/O digital screen or I/O group screen is updated with the signal name displayed here.

- If the type of a signal can be changed and is required to be changed, move the cursor to the TYPE field for that signal, press F4 [CHOICE], select an I/O type, and press [ENTER].
- To change the number of a non-UOP I/O signal, place the cursor on [Number], input a signal number, and press the [ENTER] key.
- To check whether allocation is enabled, press F5 [VERIFY].
 - If the signal exists and allocation is enabled, the message "Port assignment is valid" appears.
 - If allocation is not enabled, the message "Port assignment is invalid" appears. Re-input is necessary.
 - VERIFY does not perform a double allocation check.
- To display the detailed information for the previous I/O signal, press F2 [PREV_IO].
- To display the detailed information for the next I/O signal, press F3 [NEXT-IO].

About the cell output signal, Robot ready (This is optional function.)

The cell output signal, **Robot ready**, notifies PLC whether the robot is a production start ready status.

The Robot ready output signal does not turn ON unless the various check items listed in Table 4-3 result in 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.

The display and change procedure is explained in Procedure 3-12.

Table 3.8.5(c) 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.
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 3-12 Displaying the status check screen

Step

- 1 Press the MENUS key and select [STATUS].
- 2 Press F1 [TYPE] and select [Robot ready].

STATUS Robot ready		JOINT 50%
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:]	--
[TYPE] CONFIG		>

- 3 If any settings have been changed, press the NEXT key and then F1 [REDO] button to check the current status.

Changing monitor items (CMDENBL and SYSRDY conditions not changeable)

- Step

- 1 Position the cursor to a desired item and press F2 CONFIG.
- 2 To monitor the item, select YES. Otherwise, select NO.
- 3 To return to the list, press F2 LIST.

Adding monitor signals

- Step

- 1 Position the cursor on a DO[] field and press F2 CONFIG.
- 2 Change the SIGNAL type, number, OK condition (ON, OFF), as desired.
- 3 To monitor it, select YES. Otherwise, select NO.
- 4 To return to the list, press F2 LIST.

3.9 SETTING COORDINATE SYSTEMS

A coordinate system defines the position and attitude of the robot. The system is defined for the robot or in a work space. A joint coordinate system and a Cartesian coordinate system are used.

Joint coordinate system

The joint coordinate system is defined for robot joints. The position and attitude of the robot are defined by angular displacements with regard to the joint coordinate system of the joint base.

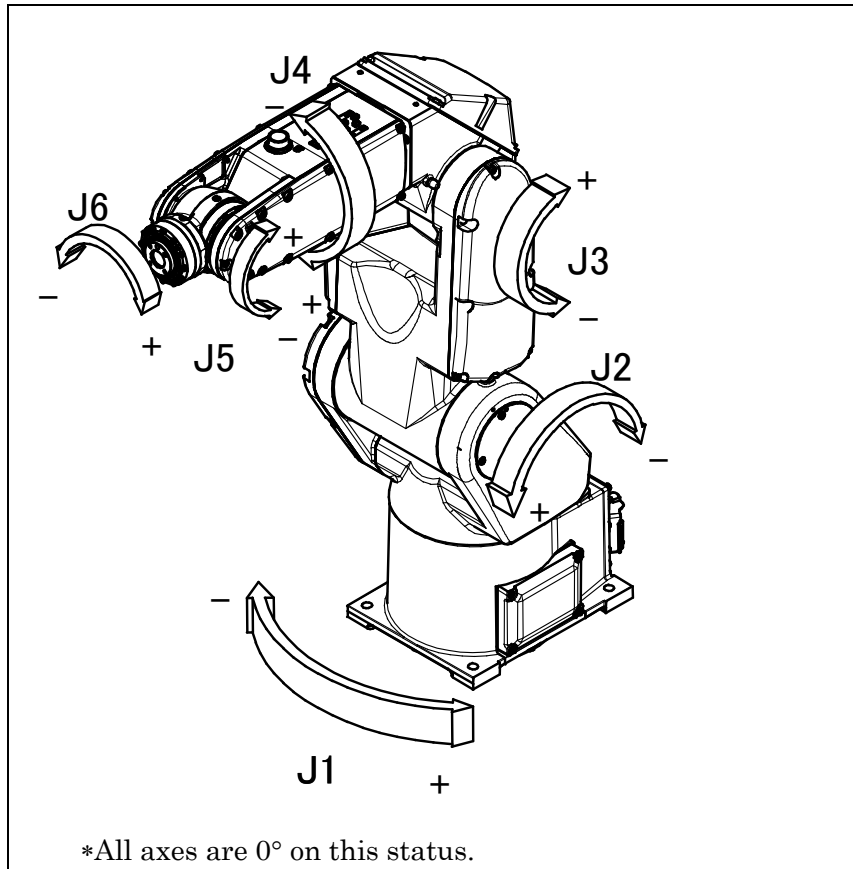


Fig. 3.9 (a) Joint Coordinate System

Cartesian coordinate system

The position and attitude of the robot in the Cartesian coordinate system are defined by coordinates x , y , and z from the origin of the space Cartesian coordinate system to the origin (tool tip point) of the tool Cartesian coordinate system and angular displacements w , p , and r of the tool Cartesian coordinate system against the X -, Y -, and Z -axis rotations of the space Cartesian coordinate system. The meaning of (w, p, r) is shown below.

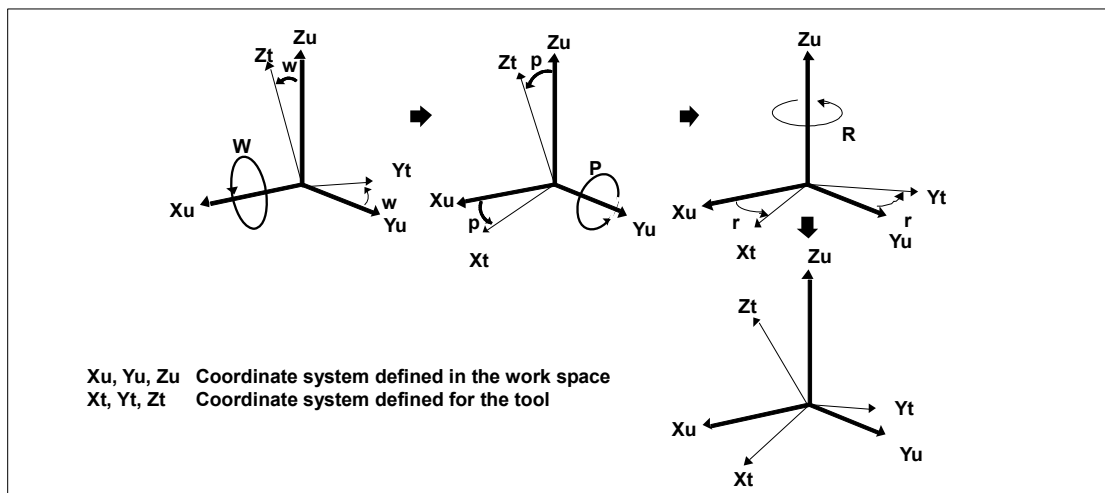


Fig. 3.9 (b) Meaning of (w, p, r)

To operate the robot in a user-specified environment, use a corresponding Cartesian coordinate system. The following five coordinate systems are available:

Mechanical interface coordinate system (Coordinate system fixed to the tool)

A standard Cartesian coordinate system defined for the mechanical interface of the robot (the surface of wrist flange). The coordinate system is fixed at a position determined by the robot. On the basis of the coordinate system, a tool coordinate system is specified.

Tool coordinate system

A coordinate system that defines the position of the tool center point (TCP) and the attitude of the tool. The tool coordinate system must be specified. If the coordinate system is not defined, the mechanical interface coordinate system substitutes for it.

World coordinate system (Coordinate system fixed in the work space)

A standard Cartesian coordinate system fixed in a work space. The coordinate system is fixed at a position determined by the robot. On the basis of the coordinate system, a user coordinate system and a jog coordinate system are specified. The world coordinate system is used for specifying position data and executing the corresponding instruction. Refer to the Appendix B.6 "World Frame Origin" for the origin of the world frame.

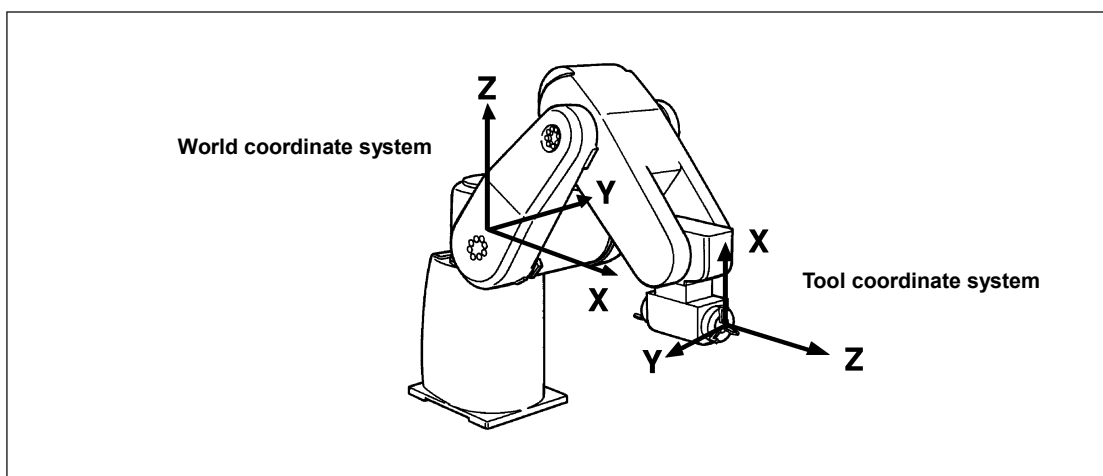


Fig. 3.9 (c) World and Tool Coordinate Systems

User coordinate system

A Cartesian coordinate system defined by the user in each work space. It is used to specify a position register, execute the corresponding position register instruction and position compensation instruction, etc. If the coordinate system is not defined, the world coordinate system substitutes for it.

WARNING

If the tool or user coordinate system is changed after program teaching, the programmed points and ranges should be reset. Otherwise, the equipment would be damaged.

Jog coordinate system

A coordinate system defined by the user. The jog coordinate system is used to efficiently move the robot by jog feed. You need not take care of the jog frame origin, since it is used only when the jog frame is selected as the manual-feed coordinate systems. If the coordinate system is not defined, the world coordinate system substitutes for it.

3.9.1 Setting a Tool Coordinate System

A tool coordinate system is a Cartesian coordinate system that defines the position of the tool center point (TCP) and the attitude of the tool. On the tool coordinate system, the zero point usually represents the TCP and the Z-axis usually represents the tool axis. When the tool coordinate system is not defined, the mechanical interface coordinate system substitutes for it.

Tool coordinates include (x, y, z) indicating the position of the tool center point (TCP), and (w, p, r) indicating the attitude of the tool. Coordinates x, y, and z indicate the position of TCP on the mechanical interface coordinate system. Coordinates w, p, and r indicate the attitude of the tool and the angular displacement around the X-, Y-, and Z-axes of the mechanical interface coordinate system.

The tool center point is used to specify the position data. The attitude of the tool is required to perform tool attitude control.

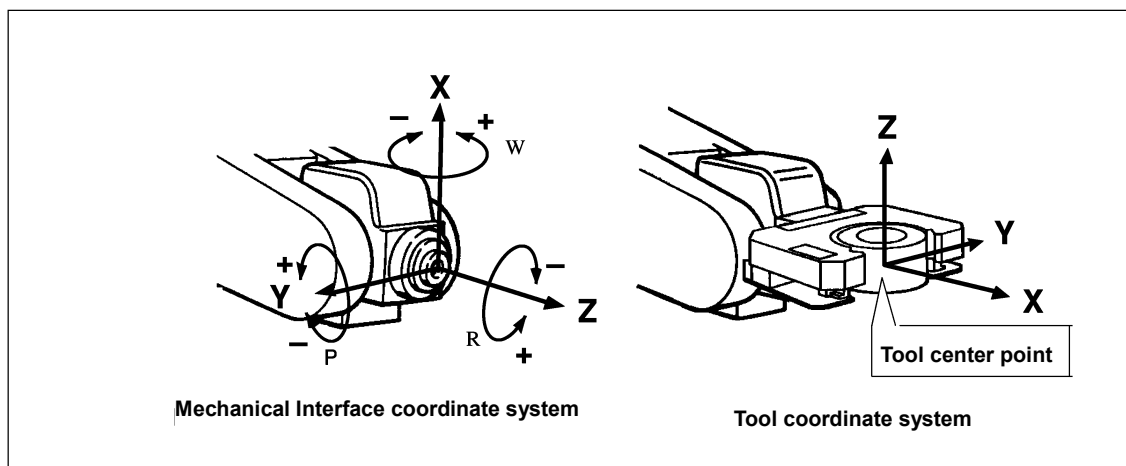


Fig. 3.9.1 (a) Tool Coordinate System

The tool coordinate system is defined by using the frame setup screen or changing the following system variables. Ten tool coordinate systems can be defined. The desired one can be selected.

- \$MNUTOL [1, i] (Frame number i = 1 to 10) is set the value.
- \$MNUTOLNUM [group] is set the used tool frame number.

NOTE

In software 7DA4 or later, the number of the tool frame can be increased to up to 29 by the following procedure.

1. Perform a controlled start.
2. Press MENUS key.
3. Select "4. Variables".
4. Change the value of the system variable \$SCR.\$MAXNUMUTOOL. (Up to 29)
5. Perform a cold start.

In case that the tool frame number used in the program is equal to or greater than 11, that program cannot be loaded into the robot controller that software series is prior to 7DA4.

The tool frame can be set by following three methods.

Three Point Method (TCP auto set)

Use the three point method to define the tool center point(TCP).The three approach points must be taught with the tool touching a common point from three different approach statuses.

As a result, the location of TCP is automatically calculated.

To set the TCP accurately, three approach directions had better differ from others as much as possible.

In the three point method, only the tool center point (x,y,z) can be set. The setting value of the tool orientation (w, p, r) is the standard value(0,0,0). The tool orientation should be defined by the six point method or direct list method after the location is set.

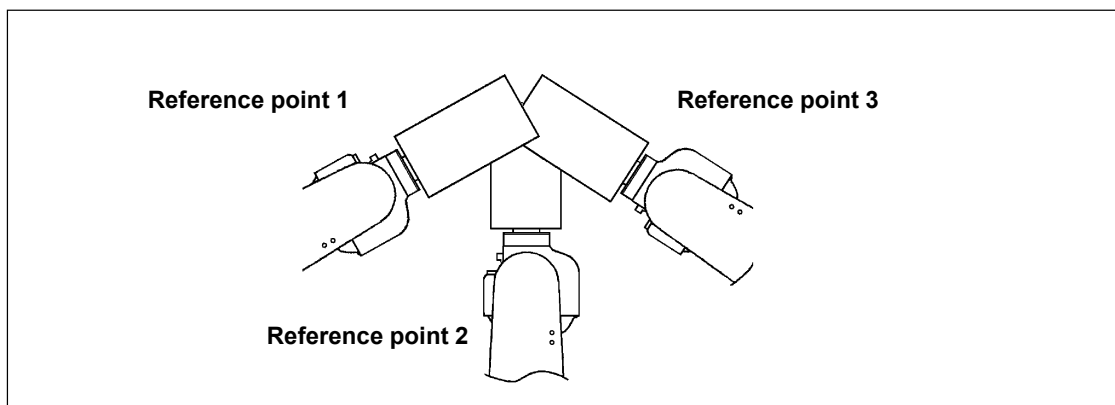


Fig. 3.9.1 (b) TCP auto set by the three point method

Six Point Method

The tool center point can be set in the same method as the three point method.

Then, set the tool attitude (w, p, r).

Teach the robot so that w, p, and r indicate a given point in space, a point in the positive direction of the X-axis parallel to the tool coordinate system, and a point on the XZ plane. Also, teach the robot using Cartesian or tool jog so that the tilt of the tool does not change.

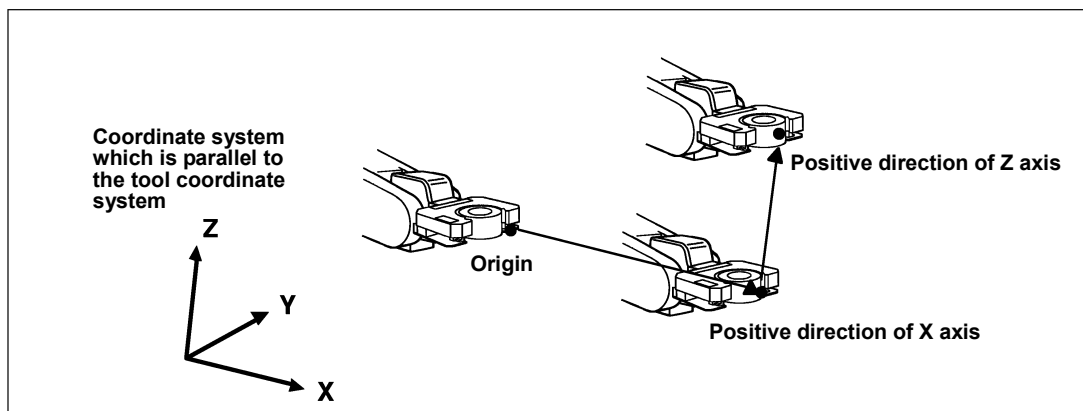


Fig. 3.9.1 (c) Six point method

Direct entry method

The following values can be entered directly. One is the value (x,y,z) of the TCP position. The other is the rotating angle (w,p,r), which specifies the tool frame orientation, around the x-,y-,and z-axis of the mechanical interface frame.

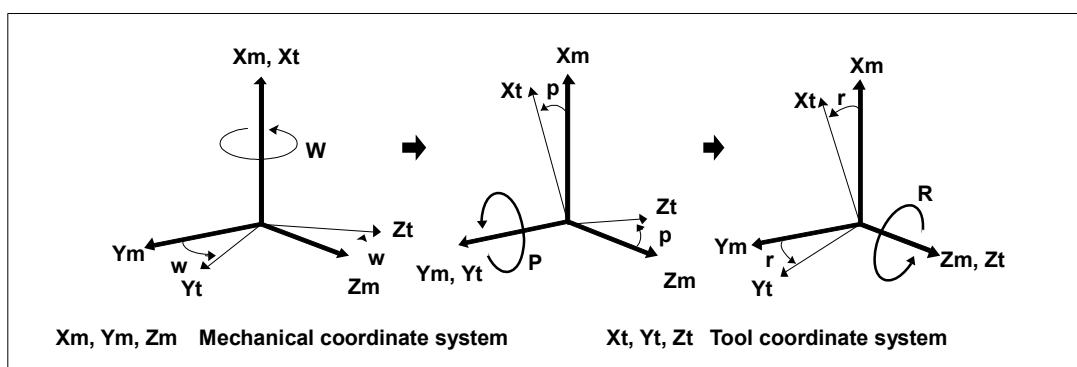
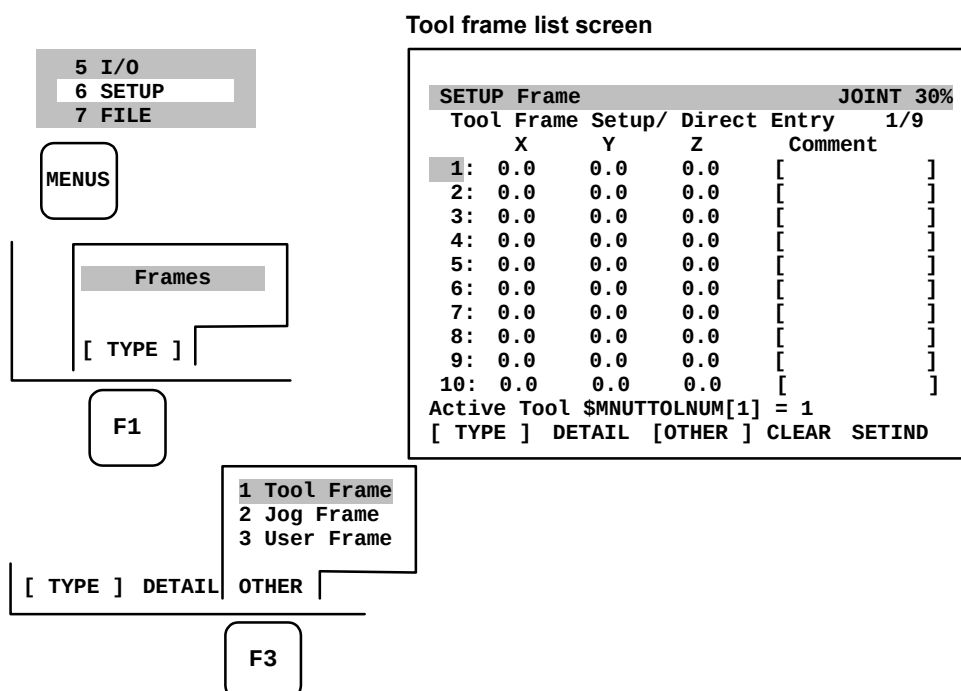


Fig. 3.9.1 (d) Meaning of (w, p, r) used in direct teaching method

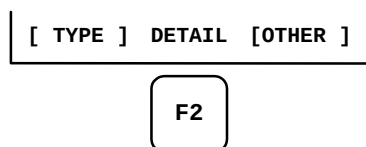
Procedure 3-13 TCP auto set (Three Point Method)

Step

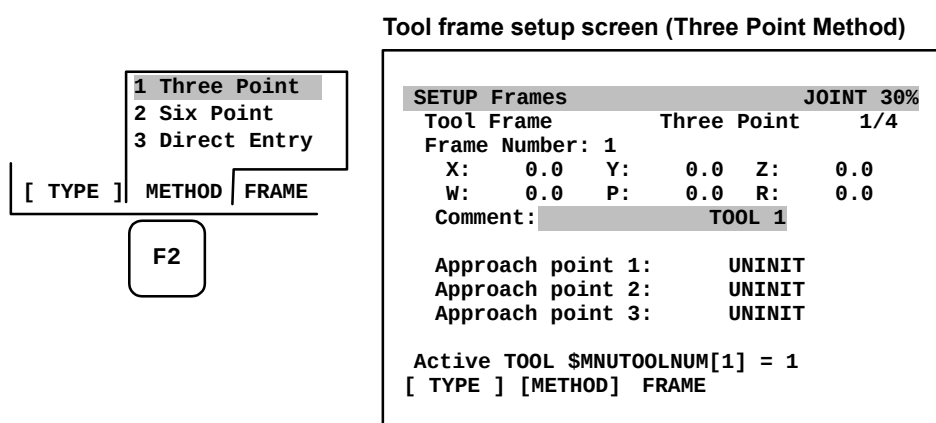
- 1 Press the MENUS key. The screen menu is displayed.
- 2 Select "6 (SETUP)."
- 3 Press the F1 key, TYPE. The screen change menu is displayed.
- 4 Select Frames.
- 5 Press F3, OTHER and then select Tool Frame. Tool frame list screen is displayed.



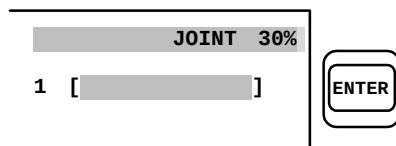
- 6 Move the cursor to the line of the tool frame number you want to set.
- 7 Press F2,DETAIL. The tool frame setup screen of the selected frame number is displayed.



- 8 Press F2,METHOD and then select Three Point.



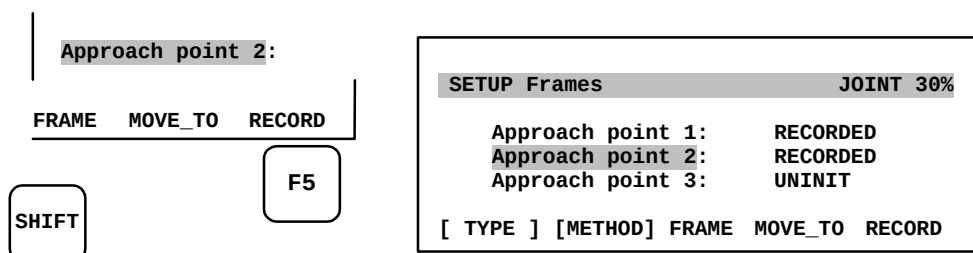
- 9 To add a comment:
 - a Move the cursor to the comment line and press the ENTER key.



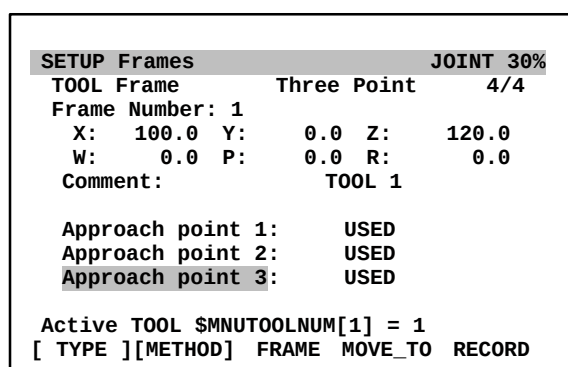
- b Select the method of naming the comment.
 - c Press the appropriate function keys to add the comment.
 - d When you are finished, press ENTER key.
- 10 Record each approach point:
- a Move the cursor to each Approach point.
 - b Jog the robot to the position you want to record.
 - c Press and hold the SHIFT key and press F5,RECORD to record the data of current position as the reference position. As for the taught reference point, RECORDED is displayed.

NOTE

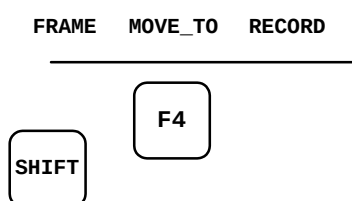
Move the tool in three different directions to bring the tool tip to an identical point. Then, record the three reference points.



- d When all the reference points are taught, USED is displayed. The tool frame has been set.



- 11 To move the robot to a recorded position, press and hold the SHIFT key and press F4,MOVE_TO.



- 12 To see each recorded position data, move the cursor to each reference position item and press the ENTER key. The position detail screen of each position data is displayed. To return to the previous screen, press the PREV key.
- 13 To display the tool frame list screen, press the PREV key. You can see the settings (x, y, z, and comment) for all tool frames.

PREV

SETUP Frames				JOINT 30%	
Tool Frame	X	Y	Z	Comment	1/10
1:	100.0	0.0	120.0	[TOOL1	]
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	[	]
10:	0.0	0.0	0.0	[	]

Active TOOL \$MNUTOOLNUM[1] = 1
 [TYPE] DETAIL [OTHER] CLEAR SETIND

- 14 To use the set tool frame as an effective tool frame now, press F5,SETIND.

[OTHER] CLEAR SETIND

F5

CAUTION

- 1 If you do not press F5, SETIND, the tool frame will not be effective.
- 2 After all coordinate systems are set, the setting information should be saved in external storage in case the information needs to be re-loaded. Otherwise, the current setting information would be lost when it is changed.

NOTE

To select the number of a coordinate system to be used, the jog menu can also be used. See Subsection 5.2.3 "Moving the Robot Jog Feed."

- 15 To delete the data of the set frame, move the cursor to the desired frame and press F4,CLEAR.

[OTHER] CLEAR SETIND

F4

Procedure 3-14 Setting Up Tool Frame Using the Six Point Method

Step

- 1 Display the tool frame list screen (Refer to the three point method).

SETUP Frames				JOINT 30%
Tool Frame	/ Direct Entry			2/10
	X	Y	Z	Comment
1:	100.0	0.0	120.0	[TOOL1]
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	[]
10:	0.0	0.0	0.0	[]
Active TOOL \$MNUTOOLNUM[1] = 1				
[TYPE] DETAIL [OTHER] CLEAR SETIND				

- 2 Move the cursor to the tool frame number line you want to set.
- 3 Press F2,DETAIL. The tool frame setup screen of the selected frame number is displayed.
- 4 Press F2,METHOD.
- 5 Select Six Point. The tool frame setup / six point screen is displayed.

Tool frame setup screen (Six Point Method)

[TYPE] DETAIL [OTHER]

F2

1 Three Point

2 Six Point

3 Direct Entry

[TYPE] METHOD FRAME

F2

SETUP Frames				JOINT 30%			
Tool Frame	Six Point			1/7			
Frame Number: 2							
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							
[TYPE] [METHOD] FRAME							

- 6 Add a comment and teach the reference point. For details, refer to TCP auto set (Three Point Method).
 - a Press and hold the SHIFT key and press F5,RECORD to record the data of current position as the reference position. As for the taught reference point, RECORDED is displayed.

FRAME MOVE_TO RECORD

SHIFT

F5

SETUP Frames		JOINT 30%
Approach point 1:		RECORDED
Approach point 2:		RECORDED
Approach point 3:		RECORDED
Orient Origin Point:		RECORDED
X Direction Point:		UNINIT
Z Direction Point:		UNINIT
Active TOOL \$MNUTOOLNUM[1] = 1		
[TYPE] [METHOD] FRAME MOVE_TO RECORD		

- b When all the reference points are taught, USED is displayed. The tool frame has been set.

SETUP Frames		JOINT 30%
Tool Frame	Six Point	7/7
Frame Number: 2		
X: 200.0	Y: 0.0	Z: 255.5
W: -90.0	P: 0.0	R: 180.0
Comment: TOOL2		
Approach point 1: USED		
Approach point 2: USED		
Approach point 3: USED		
Orient Origin Point: USED		
X Direction Point: USED		
Z Direction Point: USED		
Active TOOL \$MNUToolNUM[1] = 1		
[TYPE] [METHOD] FRAME		

- 7 Press the PREV key. The tool frame list screen is displayed. You can see all the tool frame settings.

PREV

SETUP Frames		JOINT 30%
Tool Frame	/ Direct Entry	2/10
X	Y	Z
1: 100.0	30.0	120.0
2: 200.0	0.0	255.0
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
10: 0.0	0.0	0.0
Active TOOL \$MNUToolNUM[1] = 1		
[TYPE] DETAIL [OTHER] CLEAR SETIND		

- 8 To make the set tool frame effective, press F5 (SETIND), then enter the frame number.

[OTHER] CLEAR SETIND

F5

⚠ CAUTION

- 1 If you do not press F5, SETIND, the tool frame will not be effective.
- 2 After all coordinate systems are set, the setting information should be saved in external storage in case the information needs to be re-loaded. Otherwise, the current setting information would be lost when it is changed.

NOTE

To select the number of a coordinate system to be used, the jog menu can also be used. See Subsection 5.2.3 "Moving the Robot Jog Feed."

To delete the data of the set frame, move the cursor to the desired frame and press F4,CLEAR.

[OTHER] CLEAR SETIND

F4

Procedure 3-15 Setting Up Tool Frame Using the Direct Entry Method

Step

- 1 Display the tool frame list screen (Refer to the three point method).

SETUP Frames				JOINT 30%
Tool Frame	/ Direct Entry			3/10
	X	Y	Z	Comment
1:	100.0	30.0	120.0	[TOOL1]
2:	200.0	0.0	255.0	[TOOL2]
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	[]
10:	0.0	0.0	0.0	[]
Active TOOL \$MNUTOOLNUM[1] = 1				
[TYPE] DETAIL [OTHER] CLEAR SETIND				

- 2 Move the cursor to the tool frame number line you want to set.
- 3 Press F2,DETAIL or press the ENTER key. The tool frame setup screen of the selected frame number is displayed.
- 4 Press F2,METHOD.
- 5 Select Direct Entry. Tool Frame Setup / Direct Entry screen is displayed.

Tool frame setup screen (Direct Entry Method)

[TYPE] DETAIL [OTHER]

F2

1 Three Point
2 Six Point
3 Direct Entry

[TYPE] METHOD FRAME

F2

SETUP Frames		JOINT 30%
Tool Frame	Direct Entry	1/7
Frame Number: 3		
1: Comment:	*****	
2: X:		0.0
3: Y:		0.0
4: Z:		0.0
5: W:		0.0
6: P:		0.0
7: R:		0.0
Configuration:		N D B,0,0,0
Active TOOL \$MNUTOOLNUM[1] = 1		
[TYPE] [METHOD] FRAME		

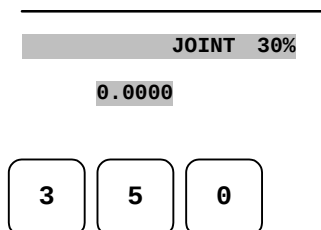
- 6 Add a comment. Refer to TCP auto set (Three Point Method) for details.

JOINT 30%

1 []

ENTER

- 7 Enter the coordinate values of the tool frame.
 - a Move the cursor to each component.
 - b Enter a new numerical value by using numerical keys.
 - c Press the ENTER key. A new numerical value is set.



SETUP Frames		JOINT 30%
Tool Frame	Direct Entry	4/7
Frame Number: 3		
1: Comment:*****		
2: X:		0.000
3: Y:		0.000
4: Z:		350.000
5: W:		180.000
6: P:		0.000
7: R:		0.000
Configuration:		N D B,0,0,0
Active TOOL \$MNUTOLNUM[1] = 1		
[TYPE] [METHOD] FRAME		

- 8 To display the tool frame list screen, press the PREV key. You can see the settings of all the tool frame.



SETUP Frames		JOINT 30%
Tool Frame	/ Direct Entry	3/10
	X Y Z	Comment
1:	100.0 30.0 120.0	[TOOL1]
2:	200.0 0.0 255.0	[TOOL2]
3:	0.0 0.0 350.0	[TOOL3]
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	[]
10:	0.0 0.0 0.0	[]
Active TOOL \$MNUTOLNUM[1] = 1		
[TYPE] DETAIL [OTHER] CLEAR SETIND		

- 9 To make the set tool frame effective, press F5 (SETIND), then enter the frame number.

[OTHER] CLEAR SETIND

F5

⚠ CAUTION

- 1 If you do not press F5, SETIND, the tool frame will not be effective.
- 2 After all coordinate systems are set, the setting information should be saved in external storage in case the information needs to be re-loaded. Otherwise, the current setting information would be lost when it is changed.

NOTE

To select the number of a coordinate system to be used, the jog menu can also be used. See Subsection 5.2.3 "Moving the Robot Jog Feed."

To delete the data of the set frame, move the cursor to the desired frame and press F4,CLEAR.

[OTHER] CLEAR SETIND

F4

3.9.2 Setting a User Coordinate System

A user coordinate system is a Cartesian coordinate system defined for each work space by the user. If the coordinate system is not defined, the world coordinate system substitutes for it.

Define the user coordinate system by (x, y, z) indicating the position of the zero point and (w, p, r) indicating the angular displacement around the X-, Y-, and Z-axes on the world coordinate system.

The user coordinate system is used to specify a position register and execute the corresponding position register instruction and position compensation instruction. For the specification of the position register, see Section 7.4, "Position Register." For the execution of the position register instruction, see Subsection 4.3.2, "Position Data." For the execution of the position compensation instruction, see Subsection 4.3.5, "Additional Motion Instruction."



CAUTION

If teaching is made by joint coordinates, changing the user coordinate system does not affect the position variables and position registers. However, note that both position variables and registers are affected by the user coordinate systems when the robot is taught in the Cartesian format.

In the right figure, the coordinates of user coordinate system 1 are

x : 70.0 mm
y : -2500.0 mm
z : 200.0 mm
w : 0.0 deg
p : 0.0 deg
r : -30.0 deg

The coordinates of user coordinate system 2 are

x : 500.0 mm
y : 0.0 mm
z : 200.0 mm
w : 0.0 deg
p : 0.0 deg
r : 0.0 deg

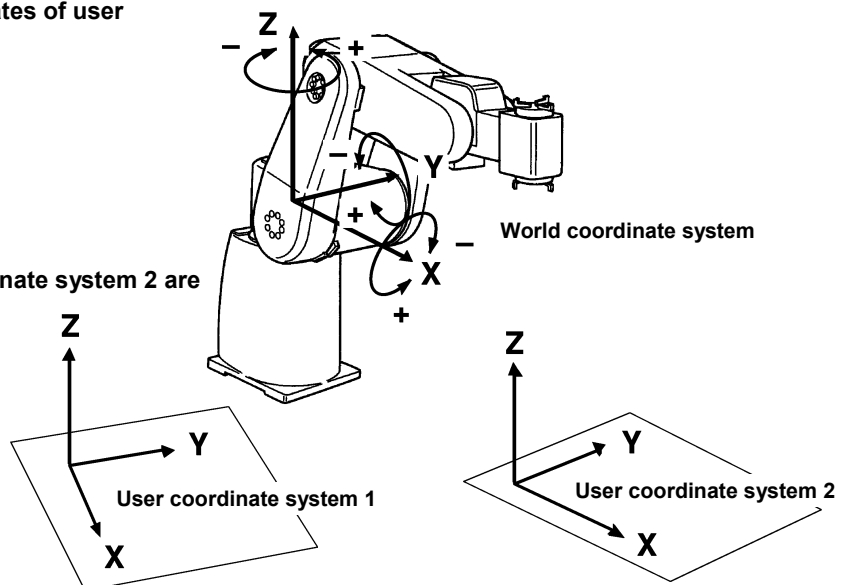


Fig. 3.9.2 (a) World and User Coordinate Systems

The following system variables are changed by defining the user frame with the frame setup screen. Nine user coordinate systems can be defined. The desired one can be selected.

- \$MNUFRAME [1, i] (Frame number i = 1 to 9) is set the value.
- \$MNUFRAMENUM [1] is set the user frame number you want to use.

NOTE

In software 7DA4 or later, the number of the user frame can be increased to up to 61 by the following procedure.

1. Perform a controlled start.
2. Press MENUS key.
3. Select "4. Variables".
4. Change the value of the system variable \$SCR.\$MAXNUMUFRAM. (Up to 61)
5. Perform a cold start.

In case that the user frame number used in the program is equal to or greater than 11, that program cannot be loaded into the robot controller that software series is prior to 7DA4.

The user frame can be defined by the following three methods.

Three Point Method

Teach the following three points: the origin of the x-axis, the point which specifies the positive direction of the x-axis, and the point on the x-y plane.

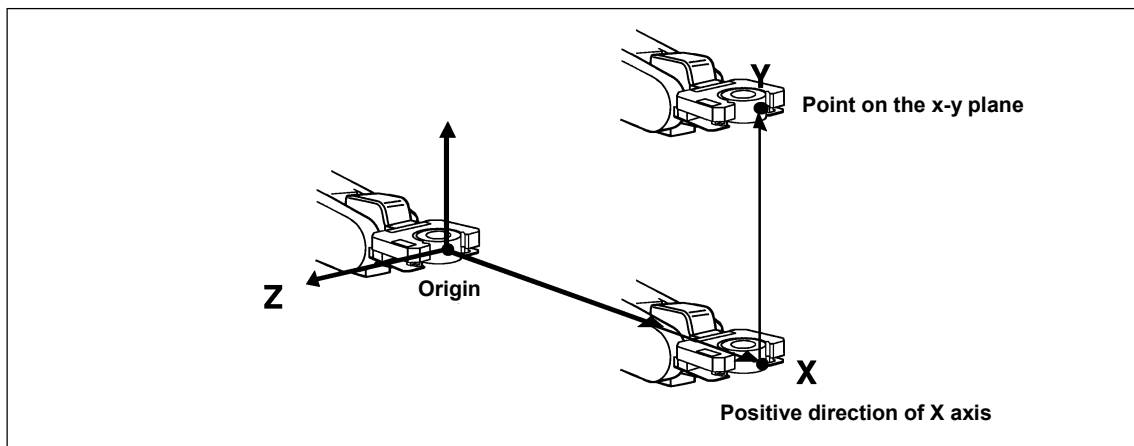


Fig. 3.9.2 (b) Three Point Method

Four Point Method

Teach the following four points: the origin of the x-axis parallel to the frame, the point which specifies the positive direction of the x-axis, a point on the x-y plane, and the origin of the frame.

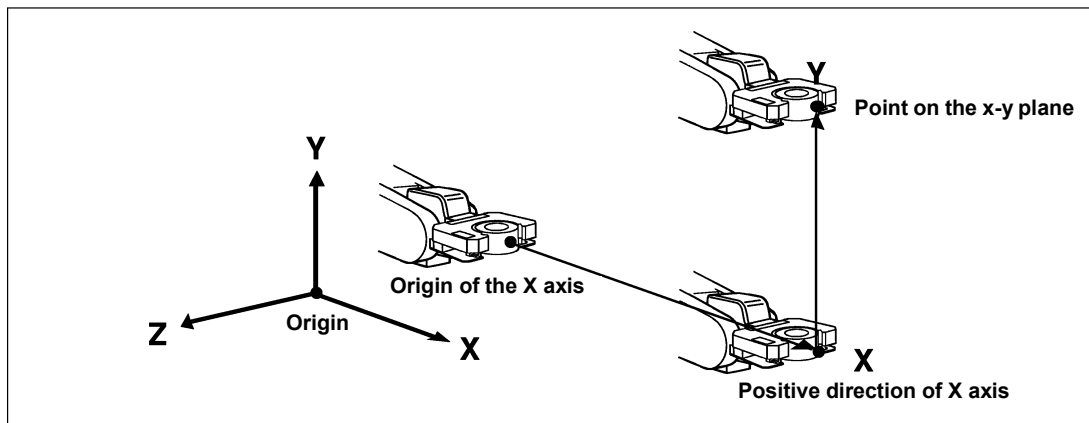


Fig. 3.9.2 (c) Four Point Method

Direct Entry Method

Enter the following values directly: the value (x,y,z) which specifies the origin of the user frame and is the coordinate values of the world frame and the rotating angle (w,p,r) around the x-,y-,and z-axis of the world frame.

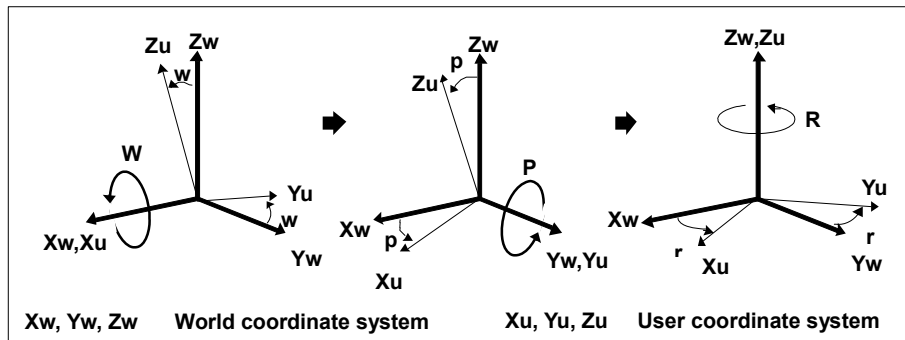


Fig. 3.9.2 (d) Meaning of (w,p,r) used in direct list method

Procedure 3-16 Setting Up User Frame Using Three Point Method

Step

- 1 Press the **MENUS** key. The screen menu is displayed.
- 2 Select "6 (SETUP)."
- 3 Press the **F1** key, **TYPE**. The screen change menu is displayed.
- 4 Select **Frames**.
- 5 Press **F3**, **OTHER** and then select User Frame. The user frame list screen is displayed.

User frame list screen

5 I/O
6 **SETUP**
7 FILE

MENUS

Frames

[TYPE]

F1

1 Tool Frame
2 Jog Frame
3 **User Frame**

[TYPE] **DETAIL** **OTHER**

F3

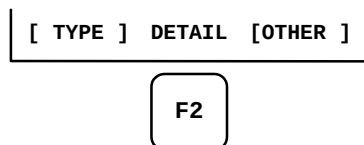
SETUP Frames JOINT 30%

User Frame	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	[]

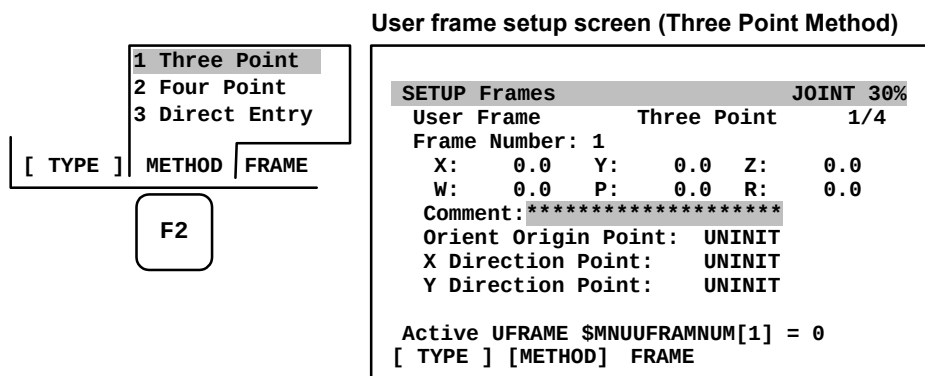
Active UFRAME \$MNUFRAMNUM[1] = 0
[TYPE] **DETAIL** [OTHER] **CLEAR** **SETIND**

- 6 Move the cursor to the line of the user frame number you want to set.

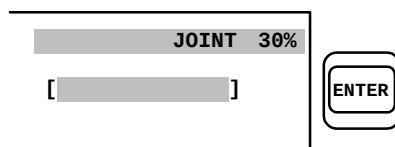
- 7 Press F2,DETAIL. The user frame setup screen of the selected frame number is displayed.



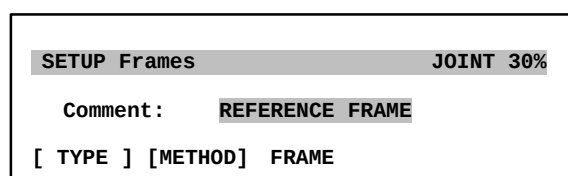
- 8 Press F2,METHOD and then select Three Point.



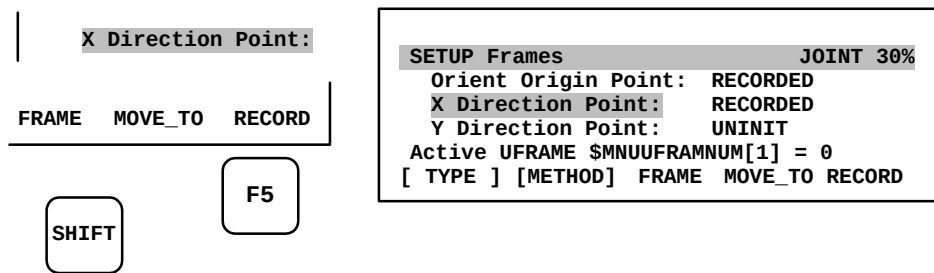
- 9 To add a comment:
- Move the cursor to the comment line and press the ENTER key.



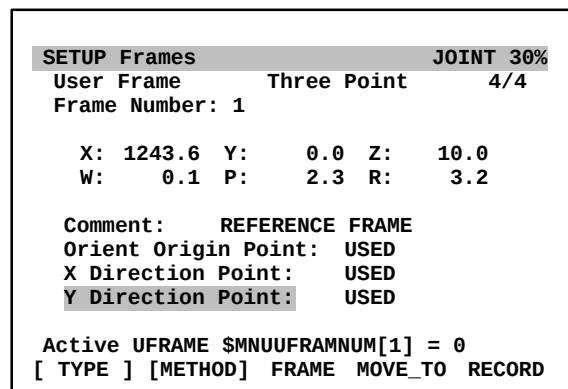
- Select the method of naming the comment.
- Press the appropriate function keys to add the comment.
- When you are finished, press ENTER key.



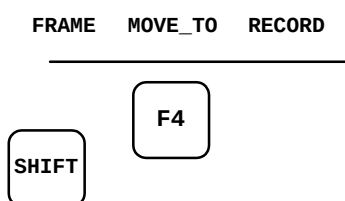
- 10 Record each approach point:
- Move the cursor to each Approach point.
 - Jog the robot to the position you want to record.
 - Press and hold the SHIFT key and press F5, RECORD to record the current position as the approach point. As for the taught reference point, RECORDED is displayed.



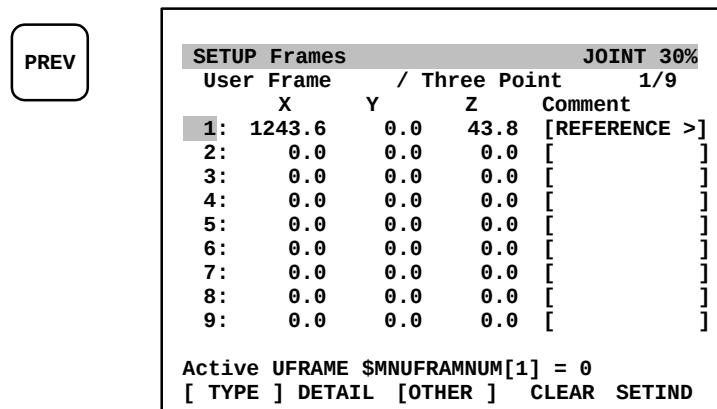
- d When all the reference points are taught, USED is displayed. The user frame has been set.



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



- 12 To see each recorded position data, move the cursor to each reference position item and press the ENTER key. The position detail screen of each position data is displayed. To return to the previous screen, press the PREV key.
- 13 To display the user frame list screen, press the PREV key. You can see the settings for all user frames.



- 14 To make the set user frame effective, press F5 (SETIND), then enter the frame number.

[OTHER] CLEAR SETIND

F5

⚠ CAUTION

- 1 If you do not press F5, SETIND, the user frame will not be effective.
- 2 After all coordinate systems are set, the setting information should be saved in external storage in case the information needs to be re-loaded. Otherwise, the current setting information would be lost when it is changed.

NOTE

To select the number of a coordinate system to be used, the jog menu can also be used. See Subsection 5.2.3 "Moving the Robot Jog Feed."

- 15 To delete the data of the set frame, move the cursor to the desired frame and press F4,CLEAR.

[OTHER] CLEAR SETIND

F4

Procedure 3-17 Setting User Frame Using Four Point Method

Step

- 1 Display the user frame list screen (Refer to the three point method)

SETUP Frames				JOINT 30%
User Frame	/ Three Point			2/9
	X	Y	Z	Comment
1:	1243.6	0.0	43.8	[REFERENCE >]
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 \$MNUFRAMNUM[1] = 0				
[TYPE] DETAIL [OTHER] CLEAR SETIND				

- 2 Move the cursor to the user frame number line you want to set.
- 3 Press F2,DETAIL. The user frame setup screen of the selected frame number is displayed.

[TYPE] DETAIL [OTHER]

F2

- 4 Press F2, METHOD
- 5 Select Four Point. The user frame setup / four point screen is displayed.

User frame setup screen (Four Point Method)

1 Three Point
2 **Four Point**
3 Direct Entry

[TYPE]
METHOD
FRAME

F2

SETUP Frames		JOINT 30%	
User Frame	Four Point	1/5	
Frame Number: 2			
X:	0.0	Y:	0.0
W:	0.0	P:	0.0
Z: 0.0 R: 0.0			
Comment:*****			
Orient Origin Point:		UNINIT	
X Direction Point:		UNINIT	
Y Direction Point:		UNINIT	
System Origin:		UNINIT	
Active UFRAME \$MNUUFRAMNUM[1] = 0			
[TYPE] [METHOD] FRAME			

- 6 Add a comment and teach the reference point. For details, refer to TCP auto set (Three Point Method).

SETUP Frames		JOINT 30%	
User Frame	Four Point	5/5	
Frame Number: 2			
X:	1243.6	Y:	525.2
W:	0.123	P:	2.34
Z: 43.9 R: 3.2			
Comment: RIGHT FRME			
Orient Origin Point:		USED	
X Direction Point:		USED	
Y Direction Point:		USED	
System Origin:		USED	
Active UFRAME \$MNUUFRAMNUM[1] = 0			
[TYPE] [METHOD] FRAME MOVE_TO RECORD			

- 7 Press the PREV key. The user frame list screen is displayed. You can see all the user frame settings.

PREV

SETUP Frames		JOINT 30%	
User Frame	/ Four Point	2/9	
X	Y	Z	Comment
1:	1243.6	0.0	43.8 [REFERENCE >]
2:	1243.6	525.2	43.8 [RIGHT FRAM>]
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 \$MNUUFRAMNUM[1] = 0			
[TYPE] DETAIL [OTHER] CLEAR SETIND			

- 8 To make the set user frame effective, press F5 (SETIND), then enter the frame number.

[OTHER] CLEAR SETIND

F5

⚠ CAUTION

- 1 If you do not press F5, SETIND, the tool frame will not be effective.
- 2 After all coordinate systems are set, the setting information should be saved in external storage in case the information needs to be re-loaded. Otherwise, the current setting information would be lost when it is changed.

NOTE

To select the number of a coordinate system to be used, the jog menu can also be used. See Subsection 5.2.3 "Moving the Robot Jog Feed."

- 9 To delete the data of the set frame, move the cursor to the desired frame and press F4,CLEAR.

[OTHER] CLEAR SETIND

F4

Procedure 3-18 Setting User Frame Using Direct Entry Method

Step

- 1 Display the user frame list screen (Refer to the three point method).

SETUP Frames				JOINT 30%
User Frame	/ Four Point			3/9
	X	Y	Z	Comment
1:	1243.6	0.0	43.8	[REFERENCE >]
2:	1243.6	525.2	43.8	[RIGHT FRAM>]
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 \$MNUFRAMNUM[1] = 0
[TYPE] DETAIL [OTHER] CLEAR SETIND

- 2 Move the cursor to the user frame number line you want to set.
- 3 Press F2,DETAIL or press the ENTER key. The user frame setup screen of the selected frame number is displayed.
- 4 Press F2,METHOD.
- 5 Select Direct Entry. The user frame setup / direct entry is displayed.

[TYPE] DETAIL [OTHER]

F2

1 Three Point

2 Four Point

3 Direct Entry

[TYPE] METHOD FRAME

F2

User frame setup screen (Direct Entry Method)

SETUP Frames JOINT 30%

User Frame Direct Entry 1/7

Frame Number: 3

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 D B,0,0,0

Active UFRAME \$UFRAMNUM[1] = 0

[TYPE] [METHOD] FRAME MOVE_TO RECORD

- 6 Add a comment and enter the coordinate values. For details, refer to tool frame (Direct Entry Method).

SETUP Frames JOINT 30%

User Frame Direct Entry 4/7

Frame Number: 3

1: Comment: LEFT FRAME

2: X: 1243.600

3: Y: -525.200

4: Z: 43.900

5: W: 0.123

6: P: 2.340

7: R: 3.200

Configuration: N D B,0,0,0

Active UFRAME \$UFRAMNUM[1] = 0

[TYPE] [METHOD] FRAME MOVE_TO RECORD

- 7 To display the user frame list screen, press the PREV key. You can see the settings of all the user frame.

PREV

SETUP Frames JOINT 30%

User Frame / Direct Entry 3/9

	X	Y	Z	Comment
1:	1243.6	0.0	43.8	[REFERENCE >]
2:	1243.6	525.2	43.8	[RIGHT FRAM>]
3:	1243.6	-525.2	43.8	[LEFT FRAME]
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 \$MNUFRAMNUM[1] = 0

[TYPE] DETAIL [OTHER] CLEAR SETIND

- 8 To get the set user frame as effective, press F5,SETIND.

[OTHER] CLEAR SETIND

F5

⚠ CAUTION

- 1 If you do not press F5, SETIND, the tool frame will not be effective.
- 2 After all coordinate systems are set, the setting information should be saved in external storage in case the information needs to be re-loaded. Otherwise, the current setting information would be lost when it is changed.

NOTE

To select the number of a coordinate system to be used, the jog menu can also be used. See Subsection 5.2.3 "Moving the Robot Jog Feed."

- 9 To delete the data of the set frame, move the cursor to the desired frame and press F4,CLEAR.

[OTHER] CLEAR SETIND

F4

Procedure 3-19 Method to change User Frame Number into "0"(World Frame)

Step

- 1 Display the user frame list screen.

SETUP Frames				JOINT 30%
User Frame	/ Four Point			3/9
	X	Y	Z	Comment
1:	1243.6	0.0	43.8	[REFERENCE >]
2:	1243.6	525.2	43.8	[RIGHT FRAM>]
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 \$MNUFRAMNUM[1] = 1
 [TYPE] DETAIL [OTHER] CLEAR SETIND

- 2 Press F->.

F→

[TYPE] CLRIND

- 3 Press F2, CLRIND. Then, \$MNUFRAMENUM[x] will be changed into 0.

```
Active UFRAME $MNUFRAMENUM[1] = 0
[ TYPE ] CLRIND
```

F2

3.9.3 Setting a Jog Coordinate System

A jog coordinate system is a Cartesian coordinate system defined in a work space by the user. It is used to efficiently move the robot by Cartesian jog in the work space. (See Subsection 5.2.3.)

The jog coordinate system is defined by (x, y, z) indicating the position of the zero point, and (w, p, r) indicating the angular displacement around the X-, Y-, and Z-axes on the world coordinate system.

NOTE

You need not take care of the jog frame origin, since it is used only when the jog frame is selected as the manual-feed coordinate system. The zero point of the jog coordinate system has no special meaning. Select any convenient position for defining the jog coordinate system.

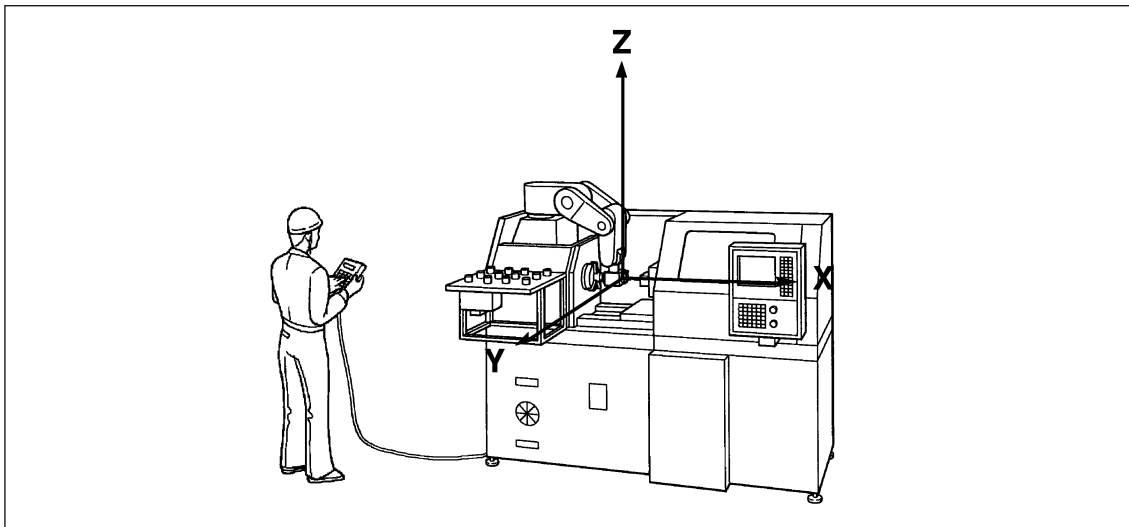


Fig. 3.9.3 Jog Coordinate System

The following system variables are changed by setting the jog frame with the frame setup screen.

- \$JOG_GROUP [1] . \$JOGFRAME is set the jog frame you want to used.

Five jog frames can be set and they can be switched according to the situation. It is substituted by the world frame when undefined.

Jog frame can be set by two methods.

Three Point Method

Three reference points need be taught. They are the start point of the x-axis, the positive direction of the x-axis, and one point on the x-y plane. The start point of the x-axis is used as the origin of the frame. Refer to Fig. 3.9.2 (b).

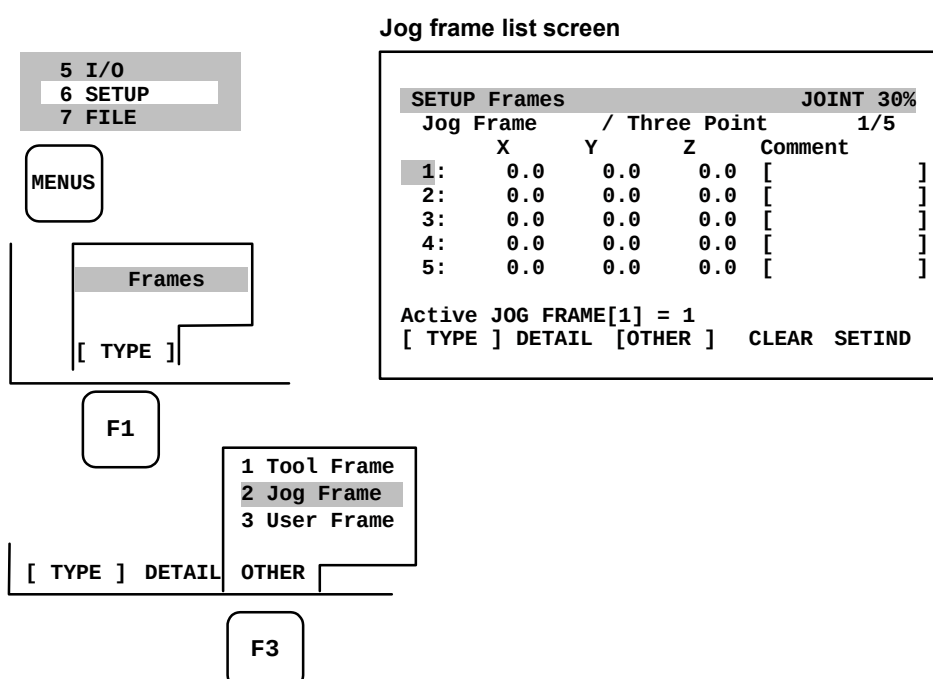
Direct Entry Method

The origin position x, y and z of the jog frame in the world frame and the rotating angle w, p, and r around the x-,y-,and z-axis of the world frame can be input directly. Refer to Fig. 3.9.2 (d).

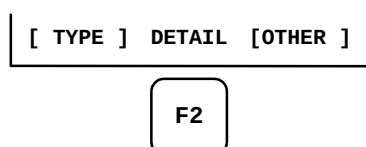
Procedure 3-20 Setting Up Jog Frame Using Three Point Method

Step

- 1 Press the MENUS key. The screen menu is displayed.
- 2 Select 6 (SETUP).
- 3 Press the F1 key, [TYPE]. The screen change menu is displayed.
- 4 Select Frames.
- 5 Press F3, OTHER
- 6 Select Jog Frame. Jog frame list screen is displayed.



- 7 Move the cursor to the line of the jog frame number you want to set.
- 8 Press F2,DETAIL. The jog frame setup screen of the selected frame number is displayed.



- 9 Press F2,METHOD.

- 10 Select Three Point.

Jog frame setup screen (Three Point Method)

1 Three Point
2 Direct Entry

[TYPE] METHOD FRAME

F2

SETUP Frames				JOINT 30%
Jog Frame		Three Point		1/4
Frame Number: 1				
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] = 1				
[TYPE] [METHOD] FRAME				

- 11 Add a comment and teach the reference point. For details, refer to TCP auto set (Three Point Method).

SETUP Frames				JOINT 30%
User Frame		Three Point		4/4
Frame Number: 1				
X:	1243.6	Y:	0.0	Z: 10.0
W:	0.1	P:	2.3	R: 3.2
Comment: WORK AREA 1				
Orient Origin Point:		RECORDED		
X Direction Point:		RECORDED		
Y Direction Point:		UNINIT		
Active JOG FRAME[1] = 1				
[TYPE] [METHOD] FRAME MOVE_TO RECORD				

- 12 Press the PREV key. The jog frame list screen is displayed. You can see all the jog frame settings.

PREV

SETUP Frames				JOINT 30%
Jog Frame		/ Three Point		1/5
	X	Y	Z	Comment
1:	1243.6	525.2	60.0	[WORK AREA 1]
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] = 1				
[TYPE] DETAIL [OTHER] CLEAR SETIND				

- 13 To make the set jog frame effective, press F5 (SETIND), then enter the frame number.

[OTHER] CLEAR SETIND

F5

⚠ CAUTION

- 1 If you do not press F5, SETIND the jog frame will not be effective.
- 2 After all coordinate systems are set, the setting information should be saved in external storage in case the information needs to be re-loaded. Otherwise, the current setting information would be lost when it is changed.

NOTE

To select the number of a coordinate system to be used, the jog menu can also be used. See Subsection 5.2.3 "Moving the Robot Jog Feed."

- 14 To delete the data of the set frame, move the cursor to the desired frame and press F4,CLEAR.

[OTHER] CLEAR SETIND

F4

Procedure 3-21 Setting Up Jog Frame Using the Direct Entry Method

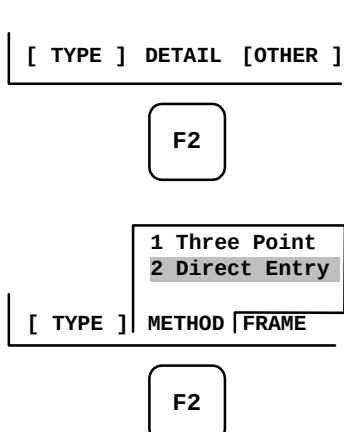
Step

- 1 Display the jog frame list screen (Refer to the three point method).

SETUP Frames				JOINT 30%
Jog Frame	/ Three Point			2/5
	X	Y	Z	Comment
1:	1243.6	525.2	60.0	[WORK AREA 1]
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] = 1
[TYPE] DETAIL [OTHER] CLEAR SETIND

- 2 Move the cursor to the jog frame number line you want to set.
- 3 Press F2,DETAIL or press the ENTER key. The jog frame setup screen of the selected frame number is displayed.
- 4 Press F2,METHOD.
- 5 Select Direct Entry.

**Jog Frame Setup Screen (Direct Entry Method)**

SETUP Frames				JOINT 30%
Jog Frame	Direct Entry			1/7
Frame Number: 2				
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 D B,0,0,0			

Active JOG FRAME[1] = 1
[TYPE] [METHOD] FRAME MOVE_TO RECORD

- 6 Add a comment and enter the coordinate value. For details, refer to TCP auto set (Direct Entry Method).

SETUP Frames		JOINT 30%
Jog Frame	Direct Entry	4/7
Frame Number: 2		
1: Comment:	WORK AREA 2	
2: X:	1003.000	
3: Y:	-236.000	
4: Z:	90.000	
5: W:	0.000	
6: P:	0.000	
7: R:	0.000	
Configuration:	N D B,0,0,0	
Active JOG FRAME[1] = 1		
[TYPE] [METHOD] FRAME MOVE_TO RECORD		

- 7 Press the PREV key. The jog frame list screen is displayed. You can see all the jog frame settings.

PREV

SETUP Frames		JOINT 30%
Jog Frame	/ Direct Entry	2/5
	X Y Z	Comment
1:	1243.6 525.2 60.0	[WORK AREA 1]
2:	1003.0 -236.0 90.0	[WORK AREA 2]
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] = 1		
[TYPE] DETAIL [OTHER] CLEAR SETIND		

- 8 To make the set jog frame effective, press F5 (SETIND), then enter the frame number.

[OTHER] CLEAR SETIND

F5

CAUTION

- 1 If you do not press F5, SETIND the jog frame will not be effective.
- 2 After all coordinate systems are set, the setting information should be saved in external storage in case the information needs to be re-loaded. Otherwise, the current setting information would be lost when it is changed.

NOTE

To select the number of a coordinate system to be used, the jog menu can also be used. See Subsection 5.2.3 "Moving the Robot Jog Feed."

- 9 To delete the data of the set frame, move the cursor to the desired frame and press F4,CLEAR.

[OTHER] CLEAR SETIND

F4

3.10 SETTING A REFERENCE POSITION

A reference position is a fixed (predetermined) position that is frequently used in a program or when the robot is moved by jog feed. The reference position is a safe position, which is usually distant from the operating area of the machine tool or peripheral equipment. Three reference positions can be specified.

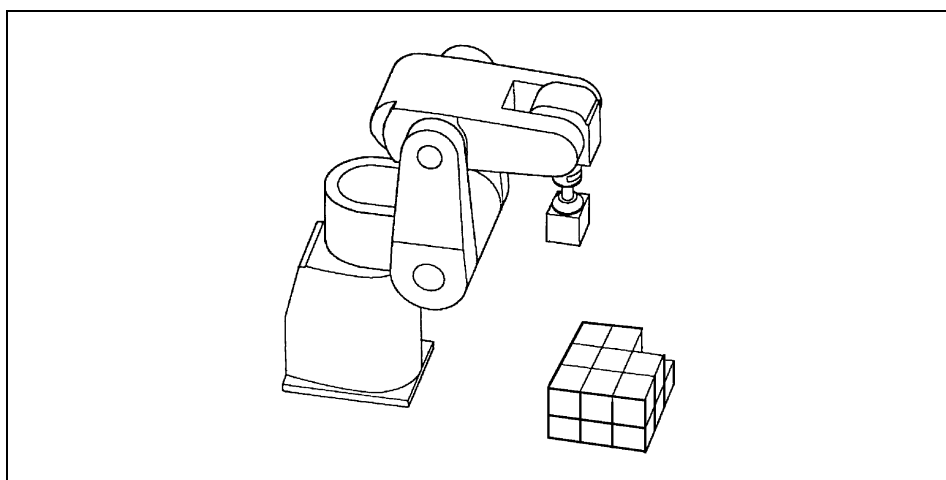


Fig. 3.10 Reference Position

When the robot is at the reference position, a predetermined digital signals, DO, is output. If the reference position is invalidated, the DO signal is not output.

When the robot is at reference position 1, the reference position output signal (ATPERCH) of the peripheral device I/O is output.

For this function, the reference position settings can be disabled so that the signal is not output.

To make the robot move to the reference position, make the program which specifies the return path and execute this program. At this time, also specify the order in which axes returns to the reference position in the program. Moreover, it is convenient to set the return program as a macro instruction.(See Section 9.1, "MACRO INSTRUCTION")

Specify the reference position on the reference position setting screen [6 (SETUP). Ref Position].

Procedure 3-22 Setting a reference position

Step

- 1 Press the MENUS key.
- 2 Select SETUP.
- 3 Press the F1 key, TYPE.

- 4 Select "Ref Position." The reference position selection screen is displayed.

Reference Position Selection Screen

5 I/O
6 **SETUP**
7 FILE

MENUS

Ref Position

[TYPE]

F1

REF POSN **JOINT 30%**

1/3

No.	Enb/Dsbl	@Pos	Comment
1	DISABLE	FALSE	[]
2	DISABLE	FALSE	[]
3	DISABLE	FALSE	[]

[TYPE] DETAIL ENABLE DISABLE

- 5 Press the F3 key, DETAIL. The detailed reference position screen is displayed.

Detailed Reference Position Screen

DETAIL ENABLE DIABLE

F3

REF POSN **JOINT 30%**

Reference Position 1/13

Ref.Position Number: 1

1 Comment: [*****]

2 Enable/Disable: **DISABLE**

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

[TYPE] RECORD

- 6 To enter a comment, follow these steps:
- Place the cursor on the comment line and press the ENTER key.
 - Determine whether the comment is entered by words, alphabetic characters, or katakana.
 - Press the corresponding function key and enter the desired comment.
 - After entering the comment, press the ENTER key.

JOINT 30%

[]

ENTER

REF POSN **JOINT 30%**

Reference Position 1/13

1 Comment: [Refpos1]

[TYPE]

- 7 In the “Signal definition” line, specify the digital output signal to be output when the tool is at the reference position.

JOINT 30% 4/13 DO [0] DO RO	JOINT 30% 4/13 4 Signal definition: RO [0] [TYPE]

F5

JOINT 30% 4/13 RO [0] 1 ENTER	JOINT 30% 4/13 4 Signal definition: RO [1] [TYPE]

- 8 To teach the reference position, place the cursor on the setting fields J1 to J9. While pressing the SHIFT key, press the F5 key, RECORD. The current position is recorded as the reference position.

RECORD SHIFT F5	JOINT 30% 5/13 5 J1: 0.000 +/- 0.000 [TYPE]

- 9 To enter the numeric value of the reference position directly, place the cursor on the setting fields J1 to J9 and enter the coordinates of the reference position. Enter the coordinates in the left column and allowable errors in the right column. Moreover, the value entered to the setting field which specifies an unused axis is ignored.

REF POSN Reference Position Ref.Position Number: 1 1 Comment: [Refpos1] 2 Enable/Disable: ENABLE 3 Is a valid HOME: FALSE 4 Signal definition: RO[1] 5 J1: 129.000 +/- 2.000 6 J2: -31.560 +/- 2.000 7 J3: 3.320 +/- 2.000 8 J4: 179.240 +/- 2.000 9 J5: 1.620 +/- 2.000 10 J6: 33.000 +/- 2.000 [TYPE]	JOINT 30% 1/13 RECORD

- 10 After the reference position is specified, press the PREV key. The reference position selection screen is displayed again.

PREV

REF POSN			JOINT 30%
			1/3
No.	Enb/Dsbl	@Pos	Comment
1	DISABLE	FALSE	[Refpos1]
2	DISABLE	FALSE	[]
3	DISABLE	FALSE	[]
[TYPE] DETAIL ENABLE DISABLE			

- 11 To enable or disable the reference position output signal, place the cursor on the ENABLE/DISABLE field and press the corresponding function key.

REF POSN		
No.	Enb/Dsbl	@Pos
1	DISABLE	FALSE
ENABLE DISABLE		

REF POSN			JOINT 30%
			1/3
No.	Enb/Dsbl	@Pos	Comment
1	ENABLE	FALSE	[Refpos1]
[TYPE] DETAIL ENABLE DISABLE			

F4

3.11 JOINT OPERATING AREA

The software restricts the operating area of the robot according to a specified joint operating area. The standard operating area of the robot can be changed by specifying the joint operating area.

Specify the joint operating area at [6 SYSTEM Axis Limits] on the joint operating area setting screen.

WARNING

- 1 The robot operating area should not be controlled only by the joint operating area function. Limit switches and mechanical stoppers should be used together with the function. Otherwise, injury or property damage would occur.
- 2 The mechanical stoppers should be adjusted to the software settings. Otherwise, injury or property damage would occur.

CAUTION

Changing the joint operating area will affect the robot operating area. Before the joint operating area is changed, the expected effect of the change should be carefully studied in order to prevent possible trouble. Otherwise, the change would produce unpredictable results. For example, an alarm might occur at a position programmed earlier.

UPPER

Specifies the upper limit of the joint operating area, which is the limit of the motion in the positive direction.

LOWER

Specifies the lower limit of the joint operating area, which is the limit of the motion in the negative direction.

Enabling the new setting

After a new joint operating area is specified, turn the controller off and on again to enable the new setting.

Procedure 3-23 Setting the joint operating area

Step

- 1 Press the MENUS key. The screen menu is displayed.
- 2 Select 6 (SYSTEM).
- 3 Press F1 (TYPE). The screen change menu is displayed.
- 4 Select Axis Limits. The joint operating area setting screen is displayed.

Joint Operating Area Setting Screen

5 POSITION
6 SYSTEM
7 USER2

MENUS

Axis Limits

[TYPE]

F1

SYSTEM Axis Limits				JOINT 30%
AXIS	GROUP	LOWER	UPPER	1/56
1	1	-160.00	160.00	dg
2	1	-30.00	160.00	dg
3	1	-156.50	206.10	dg
4	1	-120.00	120.00	dg
5	1	-200.00	200.00	dg
6	0	0.00	0.00	mm
7	0	0.00	0.00	mm
8	0	0.00	0.00	mm
9	0	0.00	0.00	mm
10	0	0.00	0.00	mm

[TYPE]

**WARNING**

The robot operating area should not be controlled only by the joint operating area function. Limit switches and mechanical stoppers should be used together with the function. Otherwise, injury or property damage would occur.

NOTE

A setting of 0.000 indicate that the robot does not have the corresponding axis.

- 5 Place the cursor on the target axis limits field, and enter a new value from the teach pendant.

SYSTEM Axis Limits

AXIS	GROUP	LOWER
2	1	-30.00

- 5 0 ENTER

SYSTEM Axis Limits

AXIS	GROUP	LOWER	UPPER	1/56
2	1	-50.00	100.00	dg

[TYPE]

- 6 Repeat the above step for all the axes.
- 7 To make the set information effective, turn the controller off and on again in cold start mode (Subsection 5.2.1).

**WARNING**

Power should be turned on again to make a new setting valid. Otherwise, injury or property damage would occur.

3.12 USER ALARM

In the user alarm setup screen, the message displayed when the user alarm is generated is set. The user alarm is the alarm which is generated when the user alarm instruction is executed. (See Subsection 4.14.2 "User Alarm Instruction")

Settings for user alarm is done in the user alarm setup screen [6 SETUP. User Alarm].

Procedure 3-24 Setting Up the User Alarm**Step**

- 1 Select the **MENUS** key. The screen menu is displayed.
- 2 Select **6(SETUP)**.
- 3 Press the **F1** key, **TYPE**. The screen change menu is displayed.
- 4 Select **User Alarm**. The user alarm setup screen is displayed.

User Alarm Setup Screen

5 I/O
6 **SETUP**
7 FILE

MENUS

User Alarm

[TYPE]

F1

Setting/User Alarm JOINT 30%

1/10

Alarm No.	User Message
[1]:	[]
[2]:	[]
[3]:	[]
[4]:	[]
[5]:	[]
[6]:	[]
[7]:	[]
[8]:	[]
[9]:	[]
[10]:	[]

[TYPE]

- 5 Move the cursor to the line of the user alarm number you want to set and press the **ENTER** key. Enter the message with the function keys.

[2]: []

[3]: []

[4]: []

ENTER

Setting/User Alarm JOINT 30%

--Insert--

Setting/User Alarm

Alarm No.	User Message
[1]:	[]
[2]:	[]
[3]:	[WORK]

Old Value:

ABCDEF GHIJK MNOPQR STUVW YZ_@*.

- 6 When you are finished to input the message of the user alarm, press the **ENTER** key. The user alarm message has been set.

Setting/User Alarm JOINT 30%

3/10

Alarm No.	User Message
[1]:	[]
[2]:	[]
[3]:	[NO WORK]
[4]:	[]
[5]:	[]
[6]:	[]
[7]:	[]
[8]:	[]
[9]:	[]
[10]:	[]

[TYPE]

3.13 VARIABLE AXIS AREAS

On the variable axis area setting screen, multiple (up to three) sets of stroke limits can be set for the J1 axis and an additional axis.

The variable axis area function allows the user to switch from one set of stroke limits to another during program execution.

* This function is offered by the specific robot only.

Upper limit

Indicates the upper limit for a joint operating area. Operating area in the plus direction.

Lower limit

Indicates the lower limit for a joint operating area. Operating area in the minus direction.

After changing an upper or lower limit, turn off the power to the controller and then turn it on with a cold start. With a cold start, the new upper or lower limit takes effect and the selected joint operating area is returned to the standard value (\$PARAM_GROUP.\$SLMT_**_NUM).



CAUTION

Changing a joint operating area affects the operating area of the robot. To avoid problems, it is necessary to thoroughly consider the effect of a change in the joint operating area before making the change.

Procedure 3-25 Setting a variable axis area

Step

- 1 Press MENU. The screen menu appears.
- 2 Select SETUP.
- 3 Press F1 "TYPE." The screen switching menu appears.
- 4 Select Stroke limit. The variable axis area setting screen appears.

```

5 I/O
6 SETUP
7 FILE
  
```

MENU

```

Stroke limit
[ TYPE ]
  
```

F1

Variable Axis Area Setting Screen

Stroke limit setup				JOINT 30%
				1/4
GROUP :1		AXIS :J1		
No.	LOWER	>-180.0	UPPER	< 180.0
1:	0.0	deg	0.0	deg
2:	0.0	deg	0.0	deg
3:	0.0	deg	0.0	deg
Default				
0:	-180.0	deg	180.0	deg
Active limit:				
\$MRR_GRP[1].\$SLMT_J1_NUM = 0				
[TYPE] GROUP# AXIS#				

- 5 Position the cursor to the desired axis area. Enter new values using the numeric keys on the teach pendant.
 - The upper and lower limits must be within the stroke limits of the system. (->Section 3.11, "Joint Operating Area"). If an attempt is made to set a value outside the limits, the upper or lower limit is fixed to the system default value.
 - To switch from one motion group to another, use the F2 key (group #).
 - To set an additional axis, press the F3 key (axis #) to switch to the additional axis setting screen.

- 6 To make the settings effective, turn off the power and then back on. When the power is turned on for the first time after the settings have been changed, a cold start is automatically performed.

Procedure 3-26 Using a variable axis area

Condition

- A proper axis area has been set and is effective.

Step

- 1 To switch to the joint operating area that has been set on the variable axis area setting screen during program execution, use the parameter instruction (→ Subsection 4.14.8, "Parameter instruction"). For example, after the following program has been executed

```

PRG1                                JOINT 30%

1:  $MRR_GRP[1].$SLMT_J1_NUM = 1
2:  $PARAM_GROUP[1].$SLMT_J1_NUM = 1

[ INST ]                            [EDCMD]>

```

Value No.1 is used for the joint operating area for the J1 axis. To switch to another joint operating area for the additional axis, use the following command:

```

PRG1                                JOINT 30%

3:  $MRR_GRP[1].$SLMT_E1_NUM = 2
4:  $PARAM_GROUP[1].$SLMT_E1_NUM = 2

[ INST ]                            [EDCMD]>

```

3.14 INTERFERENCE PREVENTION AREA FUNCTION

The interference prevention area function is a function that automatically stops the robot when a move instruction that causes the robot to enter the preset interference area is issued, the robot will stop if another robot or peripheral device is located in that interference area and, after confirming that the other robot or peripheral device has moved out of the interference area, automatically releases the robot from the stopped state to restart its operation.

Communication between a robot and a peripheral device is accomplished with a set of interlock signals (one signal for each of input and output). One set of interlock signals is allocated to one interference area. Up to three interference areas can be defined.

The relationship between the interlock signals and the robot is as described below.

Output signal

The output signal is off when the tool endpoint is located inside the interface area. It is on when it is located outside the area.

State	Output signal
Safe (tool endpoint located outside the interference area)	On
Dangerous (tool endpoint located inside the interference area)	Off

Input signal

When the input signal is off, and the robot attempts to enter the interference area, the robot enters the hold state. When the input signal is turned on, the robot is released from the hold state, automatically restarting its operation.



CAUTION

The robot decelerates to stop at the point where the tool endpoint enters the interference area, so that the robot actually stops at a position inside the interface area. The faster the operating speed of the robot, the deeper the robot enters the interference area. Consider this and other factors, such as the tool size, to ensure that a sufficiently large interface area is set.

To set up the interference prevention area function, use the SETUP Space function.

To set up the following items, use the Rectangular Space/DETAILED screen.

Table 3.14 (a) Items of the Interference prevention Area Function (Area Details Screen)

Item	Description
Enable/disable	Enables and disables this function. To change the settings of the other items, this function must be disabled for the area for which the settings of the items are to be changed.
Comment	Allows the user to enter a comment of up to 10 characters.
Output signal	Sets up the output signal.
Input signal	Sets up the input signal.
Priority	When two robots use this function, this item specifies which robot is to enter the interference area first if the two robots attempt to enter the interference area at the same time. The robot for which High is set enters the interface area first. When the robot completes its operation and moves out of the interference area, the robot for which Low is set enters the interference area. The setting for one robot must be different from that for the other. NOTE If High or Low is set for both robots, and the robots attempt to enter the interference area at the same time, they both enter the stopped (deadlock) state. If this occurs, perform the recovery operation described below and check that the settings are correct. 1 Perform an emergency stop on both robots. WARNING If an emergency stop is not performed on both robots, one robot will automatically start its operation when the other moves out of the interference area. This is very dangerous. 2 Check that there are no objects or by standers that a robot could hit. 3 Disable this function. 4 Move either robot out of the interference area, using a jog operation.
inside/outside	Specifies whether the inside or outside of a rectangular parallelepiped is to be an interference area.

To set up the following items, use the Rectangular Space/SPACE SETUP screen.

Table 3.14 (b) Items of the Interference prevention Area Function (Area Setting Screen)

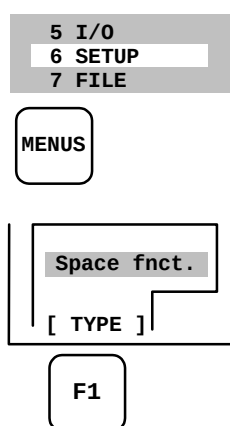
Item	Description
BASIS VERTEX	Position of the vertex of a rectangular parallel pipe that is to become the reference.
SIDE LENGTH/SECOND	If SIDE LENGTH is selected, specify the lengths of the sides of a rectangular parallelepiped from the reference vertex along the X, Y, and Z axes in the user coordinate system. (The

Item	Description
VERTEX	sides of the rectangular parallel pipe must be parallel to the respective axes of the user coordinate system.) If SECOND VERTEX is selected, the rectangular parallel pipe having the reference vertex and the diagonal vertex, specified here, becomes an interference area.

Procedure 3-27 Setting up the interference prevention area function

Step

- 1 Press **MENUS**. The screen menu appears.
- 2 Select **SETUP**.
- 3 Press **F1** "TYPE." The screen switching menu appears.
- 4 Select **Space fnct.** The area list screen appears.

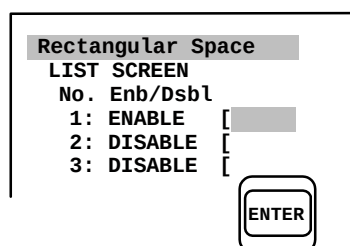


Area List Screen

Rectangular Space			JOINT 30%
LIST SCREEN			
No.	Enb/Dsbl	Comment	Usage
1:	ENABLE	[]	Common Space
2:	DISABLE	[]	Common Space
3:	DISABLE	[]	Common Space

[TYPE] GROUP DETAIL ENABLE DISABLE

- 5 The area list screen allows the user to enable and disable each interface area with the appropriate function key. To enter a comment, use the procedure below:
 - a. Move the cursor to the desired comment line and press the Enter key.
 - b. Specify which of alphabetic or katakana characters are to be used to enter a comment.
 - c. Press the appropriate function key to enter a comment.
 - d. When a comment is entered, press the Enter key.



- 6 To set up an item other than Enb/Dsbl or Comment, press **F3** (DETAIL). The details screen appears.

DETAIL	ENABLE	DISABLE
F3		

Rectangular Space		JOINT 30%
DETAILED SCREEN		1/7
SPACE:1	GROUP:1	
USAGE: Common Space		
1 Enable/Disable:	DISABLE	
2 Comment:	[*****]	
3 Output Signal:	DO [0]	
4 Input Signal:	DI [0]	
5 Priority:	High	
6 Inside/Outside	Inside	
7 Common Space Num:	0	
[TYPE] SPACE SETUP ENABLE DISABLE		

- 7 Position the cursor to the desired item. Change the setting of the item using the function or numeric keys.
- 8 To set an area, press SPACE. The area setting screen appears.

[TYPE] SPACE	
F2	

Rec		JOINT 30%
SPACE SETUP		1/4
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	
[TYPE] OTHER		RECORD

- 9 The reference vertex and the side lengths or diagonal vertex can be set in either of two ways:
 - a. Position the cursor to the X, Y, and Z coordinate fields and enter the desired coordinates directly using the numeric keys.
 - b. Move the robot to a vertex of a rectangular parallelepiped, then read the current position of the robot with SHIFT key +F5 RECORD.

RECORD	
SHIFT	F5

NOTE

- 1 If UF or UT is to be changed, perform operation b first. By means of this operation, the current UF or UT value is selected.
- 2 When the user coordinate system values are changed, the spatial position of the interference area does not change. When the user coordinate system values have been changed and an interference area is to be defined in the new user coordinate system, use SHIFT key +F5 RECORD to set an interference area again.

- 10 After setting the area, press PREV. The area details screen reappears. To return to the area list screen, press PREV again.



3.15 SYSTEM CONFIG MENU

The System Config Menu includes some important components which should be set when the system is established.

In the system config menu, the following items can be set.

Table 3.15 System config menu

Items	Descriptions
Use HOT START (Hot Start)	When the hot start is set to TRUE, hot start is done at turning on the controller. (Default setting = TRUE)
I/O power fail recovery	Specifies whether or how to perform I/O power failure recovery if the hot start function is enabled and how to perform simulated recovery if the hot start function is disabled. There are four power failure recovery modes, as described below. - NOT RECOVER I/O power failure recovery is not performed regardless of whether the hot start function is enabled. All outputs are turned off, and the simulated state is reset. - RECOVER SIM Simulated-state recovery is performed regardless of whether the hot start function is enabled, and the simulated state is reset, but all actual outputs and simulated inputs/outputs are turned off. - UNSIMULATE I/O power failure recovery is performed, but all the simulated states are reset. This is equivalent to NOT RECOVER if the hot start function is disabled, because the output states are not recovered. - RECOVER ALL I/O power failure recovery is performed if the hot start function is enabled. The output and simulated states are recovered to the states that existed immediately before the power is turned off. If the hot start function is disabled, RECOVER ALL is equivalent to RECOVER SIM, because the output states are not recovered.  CAUTION Even if power failure handling is enabled, the output signal is turned off without being recovered in the following cases: • When the I/O allocation is changed before the power is turned off. • When the fuse of an I/O unit blows, or when an I/O unit is turned off. • When the I/O unit configuration is changed.

Items	Descriptions
Autoexec program for Cold start Autoexec program for Hot start	Specifies the name of the auto-start program for the cold start and the hot start. The specified program is executed immediately after the power is turned on. If it does not end within 15 seconds, it will be aborted.  CAUTION The program automatically executed at power on is executed just before the servo power is turned on. Therefore the robot can not be moved by this program. Set the program which initializes the condition of setup and I/O of the system. You should set the name of program which sets up the system, initializes I/O...etc. Moreover, the attributes should be set as follows on the program detail screen. Group Mask: [* , * , * , * , * , * , * , *] Ignore pause: [ON]
HOT START done signal	Specifies the digital signal (DO) that is to be output at the hot start. If the hot start is not performed, the digital signal is turned off. This function is disabled if 0 is specified.
Restore selected program	Specifies whether the program selected at turning off the controller is selected after turning on the controller when the cold start is done. When this is set to TRUE, the program selected at the power off is selected after the power on again. When this is set to FALSE, the program is not selected after power on again. This is set to TRUE in standard setting.
Enable UI signals	Selects whether a UI signal is valid or invalid. When this is set to FALSE, the peripheral input signals (UI[1 to 18]) is disabled. See Section 3.3 "PERIPHERAL I/O".
START for CONTINUE only	If this item is enabled, the external start signal (START) starts only those programs that have been paused. → See Section 3.3 "PERIPHERAL I/O".
CSTOPI for ABORT	If this item is enabled, those programs that are currently running are forcibly terminated immediately upon the input of CSTOPI. → See Section 3.3 "PERIPHERAL I/O".
Abort all programs by CSTOPI	Specifies whether all programs are to be forcibly terminated with the CSTOPI signal in a multitasking environment. If this item is set to TRUE, the CSTOPI input signal terminates all programs forcibly: If this item is set to FALSE, the CSTOPI input signal causes only the currently selected program to be forcibly terminated. (Default setting)
PROD-START depend on PNSTROBE	If this item is enabled, the PROD_START input is enabled only when the PNSTROBE input is on. By enabling this item, it is possible to prevent a program that should not be started from being started accidentally due to noise or a sequence error when that program is displayed on the teach pendant.
Detect FAULT_RESET signal	Specifies whether the reset signal is detected the instant it rises or falls. When this setting is changed, turn the controller off and on again to use the new information. At this time the cold start is done automatically. The falling edge is detected by standard setting.
Use PPABN signal	Specifies if the pneumatic pressure alarm(*PPABN) is detected for each motion group. Move the cursor to this line and press ENTER key. The setup screen for each motion group is displayed. When *PPABN signal is not used, set this invalid. When this setting is changed, turn off the controller, and turn on the controller to use the new information. At this time it start when HOT START is disable.
WAIT timeout	Specifies the period of time used in the conditional wait instruction(WAIT ..., TIMEOUT LBL[...]). The period of time is 30 second.

Items	Descriptions
RECEIVE timeout	For this item, set the limit time for register receive instruction RCV R[...] LBL[...] (can be specified only when the sensor interface option is specified).
Return to top of program	Specifies whether the cursor is moved to the top of the program or not when it is finished to be executed. When this setting is TRUE, the cursor stays at the last line without returning to the top of the program when the program is finished to be executed. The default setting is TRUE. It is effective in the standard.
Original program name (F1 to F5)	Specifies the words which is displayed as the soft key at registering a program. It is convenient to set the words used many times as the program name to this.
Default logical command	It is possible to enter the screen to which standard instruction function key is set by pushing the input key from the condition that there is a cursor in setting a standard instruction. - Name Specifies the name which is displayed as the function key title.(Up to 7 characters) - Lines Specifies the number of the logic command registered in on function key. The default logical command up to four can be registered in one function key. When the Lines is set to 0,the function of teaching the default logical command is invalid.
Maximum of ACC instruction	Specifies the maximum of the override value used in the acceleration override motion option(ACC ...). The default value is 150.
Minimum of ACC instruction	Specifies the minimum of the override value used in the acceleration override motion option(ACC ...).
WJNT for default motion	Adds the Wjnt motion option to all linear and circular default motion instructions or delete it from them. - Pressing the F4 (ADD) key adds the Wjnt motion option to all the linear and circular default motion instructions and changes the screen display from "DELETE" (or *****) to "ADD." - Pressing the F5 (DELETE) key deletes the Wjnt motion option from all the linear and circular default motion instructions and changes the screen display from "ADD" (or *****) to "DELETE."
Auto display of alarm menu	Toggles the function for automatically displaying the alarm screen between FALSE and TRUE. The default setting is FALSE. If the setting of this item is changed, the power must be turned off and then back on. • FALSE : Does not display the alarm screen automatically. • TRUE : Displays the alarm screen automatically.
Force Message	Specifies whether the user screen is to appear automatically when a message instruction is executed in a program.
Reset CHAIN FAILURE detection	Resets a chain abnormality alarm (servo 230 or 231) when it is issued. For details on the chain abnormality alarm and for how to make hardware checks, refer to the maintenance manual. <Reset procedure> 1) Check for any hardware problem. 2) Press the emergency stop button on the teach pendant. (Input an emergency stop signal other than the emergency stop signal currently generated.) 3) Turn the emergency stop button on the teach pendant to release the emergency stop condition. 4) Move the cursor to this line, then press the F4 (TURE) key. 5) Press the reset button on the teach pendant.
Allow force I/O in AUTO mode	Enables or disables signal setting from TP when AUTO mode is set. By default, setting is enabled. - Yes : Enables signal setting. - No : Disables signal setting.
Allow chg. ovr. in AUTO mode	Enables or disables override change from TP when AUTO mode is set. By default, change is enabled. - Yes : Enables override change. - No : Disables override change.

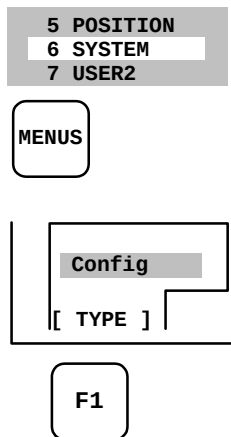
Items	Descriptions
Signal to set in AUTO mode	If the three-mode switch is set to AUTO mode, a specified DO is turned on. When 0 (default) is set, this function is disabled. When the setting has been changed, the power must be turned off then back on.
Signal to set in T1 mode	If the three-mode switch is set to T1 mode, a specified DO is turned on. When 0 (default) is set, this function is disabled. When the setting has been changed, the power must be turned off then back on.
Signal to set in T2 mode	When the three-mode switch is set to T2 mode, a specified DO is turned on. When 0 (default) is set, this function is disabled. When the setting has been changed, the power must be turned off then back on.
Signal to set if E-STOP	When an emergency stop (TP external emergency stop, operator's panel) is applied, a specified DO is output. When 0 (default) is set, this function is disabled. When the setting has been changed, the power must be turned off then back on.
Simulated status signal (Set if INPUT SIMULATED)	Monitors to see if an input signal set as a simulated one exists and outputs it as an output signal. For this item, set the number of the output signal that will turn on when a digital, group, robot, or analog input signal is set as a simulated one. When the setting has been changed, the power must be turned off then back on.
Simulated status signal (Set if OUTPUT SIMULATED)	Monitors to see if an output signal set as a simulated one exists and outputs it as an output signal. For this item, set the number of the output signal that will turn on when a digital, group, robot, or analog input signal is set as a simulated one. When the setting has been changed, the power must be turned off then back on.
Simulated input wait time (Sim. Input Wait Delay)	Sets the time after which a wait command causes a timeout if the simulated signal skip function is enabled. When the setting is changed, the change is immediately applied.
Setting to be assumed when the simulated signal skip function is enabled (Set if Sim. Skip enabled)	Monitors to see if there is any input signal for which the simulated signal skip function is enabled and outputs the signal as an output signal. For this item, set the number of the output signal that will turn on when the simulated signal skip function is enabled for a digital or robot input signal. When the setting has been changed, the power must be turned off then back on.
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.
Wait for DI range	Below 3 items setup "Monitoring WAIT for DI function" This function output DO ON when the timeout time is passed during WAIT instruction of program is waiting for specified DI. The range of monitoring DI is set by this item. Timeout time is set by the item of "Wait for DI time". Output signal is set by the item of "Wait for DI output". This function is usable from 7DA4 or later.
Wait for DI time	This item setup timeout time of "Monitoring WAIT for DI function"
Wait for DI output	This item setup output signal of "Monitoring WAIT for DI function"
Signal to be output at an override of 100% (Signal if OVERRIDE= 100)	Sets the number of the digital output signal for notifying that the override is set to 100%. For the digital output, on is output if the override is 100% and otherwise, off is output. When the setting has been changed, the power must be turned off then back on.

Items	Descriptions
Hand broken	Enables and disables hand breakage (*HBK) detection. When multiple robots are used, hand breakage detection can be enabled and disabled for two robots. Press the Enter key with the cursor positioned on this line. Then, the screen for enabling or disabling hand breakage detection for each robot appears. On this screen, move the cursor to ENABLE or DISABLE, then press the ENABLE (F4) or DISABLE (F5) key to enable or disable hand breakage detection. When hand breakage detection is enabled, and the *HBK signal is off, alarm "SRVO-006 SERVO Hand broken" is issued. Refer to the Alarm Code List Operator's Manual to release the alarm. When the *HBK signal is off, and this signal is not to be used, disable the hand breakage detection. When hand breakage detection is disabled although a hand is installed, and the *HBK signal is used, "SRVO-302 SERVO Set Hand broken to ENABLE" is displayed if the *HBK signal is on. Enable hand breakage detection. If the *HBK signal is turned off when hand breakage detection is disabled, "SRVO-300 SERVO Hand broken/HBK disabled" is issued. In this case, this alarm can be released by pressing the reset key. By default, hand breakage detection is enabled.
Remote/Local setup	Select the method for setting the remote signal (SI[2]) that switches between remote mode and local mode of the system. • Remote : Keeps SI[2] on (remote mode) at all times. • Local : Keeps SI[2] off (local mode) at all times. • External I/O : Reflects the external signal status on SI[2]. When selecting this item, specify an external signal for "External I/O (ON : Remote)" on the next line. • OP panel key : When the R-30iA Mate controller is used, this item cannot be selected.
External I/O (ON : Remote)	When "External I/O" is selected in "Remote/Local setup" above, specify an external signal to be used here. Choose from "DI, DO, RI, RO, UI, and UO".
Automatic UOP allocation	Performs automatic UOP allocation. Select the allocation type from the following. - None: Clears UOP allocation. - Full: Allocates the UOP with 18 input points and 20 output points, which is the same as that for the R-30iA, to an I/O link master etc. - Full (Slave): Allocates the UOP with 18 input points and 20 output points, which is the same as that for the R-30iA, to an I/O link slave. - Full (CRMA16): Allocates the UOP with 18 input points and 20 output points, which is the same as that for the R-30iA, to CRMA15 and CRMA16. - Simple: Allocates the UOP with 8 input points and 4 output points, which is the same as that for the R-J3iB Mate, to an I/O link master etc. - Simple (Slave): Allocates the UOP with 8 input points and 4 output points, which is the same as that for the R-J3iB Mate, to an I/O link slave. - Simple (CRMA16) : Allocates the UOP with 8 input points and 4 output points, which is the same as that for the R-J3iB Mate, to CRMA16. For the types and positions of the signals to be allocated, see Section 3.3, "PERIPHERAL I/O". Turn the power off and back on again for the changed settings to take effect.
Multiple-program selection (Multi Program Selection)	This setting is for switching the program selection method between single task mode and multitask mode. If this setting is enabled, multitask mode is selected, and if it is disabled, single task mode is selected. 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.

Procedure 3-28 Setting The System

Step

- 1 Select the MENUS key. The screen menu is displayed.
- 2 Select 6(SYSTEM) in the next page.
- 3 Press the F1 key, [TYPE]. The screen change menu is displayed.
- 4 Select Config. The system configuration screen is displayed.



System Configuration Screen

```

System/Config                               Joint 30%
                                           1/48
1 Use HOT START:                           FALSE
2 I/O power fail recovery: RECOVER ALL
3 COLD START Autoexec program:
[*****]
4 HOT START Autoexec program:
[*****]
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 CSTPI: 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 Return to top of program: TRUE
17 Original program name(F1): [RSR ]
18 Original program name(F2): [PNS ]
19 Original program name(F3): [STYLE ]
20 Original program name(F4): [JOB ]
21 Original program name(F5): [TEST ]
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. ovr. in AUTO mode: TRUE
31 Signal to set in AUTO mode DOUT: DO[ 0]
32 Signal to set in T1 mode DOUT: DO[ 0]
33 Signal to set in T2 mode DOUT: DO[ 0]
34 Signal to set if ESTOP DOUT: DO[ 0]
35 Set if INPUT SIMULATED: DO[ 0]
36 Set if OUTPUT SIMULATED: DO[ 0]
37 Sim. Input Wait Delay: 0.00 sec
38 Set if Sim. Skip Enabled: DO[ 0]
39 Set when prompt displayed: DO[ 0]
40 WAIT for DI range: DI[ 0- 0]
41 WAIT for DI time: 0.00 sec
42 WAIT for DI output: DO[ 0]
43 Signal if OVERRIDE = 100: DO[ 0]
44 Hand broken: <*GROUPS*>
45 Remote/Local setup: Remote
46 External I/O(ON:Remote): DI[ 0]
47 UOP auto assignment: FULL
48 Multi Program Selection: FALSE

[ TYPE ]

```

- 5 Move the cursor to the field you want to set and enter the new value by using the numerical key or using the function key on the teach pendant. As for the field which should be set character string, move the cursor to it and press the ENTER key. Then the character input becomes possible.

NOTE

As for the setting of "Use PPABN signal:", "Hand Broken:" or "Default logical command:", move the cursor to "<*GROUPS*>" or "<*DETAIL*>" and press the ENTER key. Then each setting screen is displayed. Press the PREV key to get out of these screens.

13 Use PPABN signal: <*GROUPS*>
 14 Wait timeout: 30.00 sec
 15 RECEIVE timeout: 30.00 sec

ENTER

- 6 When you change the setting that the cold start must be done after a setting is changed, the following message is displayed. In that case perform the cold start. (See Section 5.2, "TURNING ON THE POWER AND JOG FEED")

Please power on again
 [TYPE]

3.16 SETTING THE GENERAL ITEMS

[6 SETUP General] has the following items.

- Break on hold
- Current language
- Ignore Offset command
- Ignore Tool-offset

Table 3.16 Setting the general items

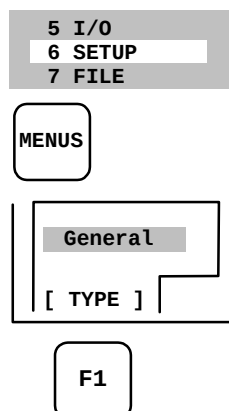
Items	Descriptions
Break on hold	Specifies whether to issue an alarm and turn off the servo alarm when the HOLD key is pressed. • If the function is DISABLED, no alarm is issued when the operation is halted by the HOLD key (standard setting). • If the function is ENABLED, an alarm is issued and the servo power is turned off, when the operation is halted by the HOLD key. To be ENABLE this function power need to be on again. ⚠ WARNING Not all axes are equipped with a brake. The brake on hold function has no effect on an axis without brake even if the function is enabled. Before the brake on hold function is enabled, it should be checked which axis has a brake. Otherwise, injury would occur.
Current language	The current language is set to "DEFAULT" by standard setting. Changing the current language requires special work. Usually, the standard setting should be used.
Ignore Offset command	Specifies whether to ignore the offset command (Subsection 4.3.5 "Additional motion instructions"). • If the function is DISABLED, the robot moves to the position for which the offset command has been executed (standard setting). • If the function is ENABLED, the robot moves to the taught position (for which the offset command has not been executed). See Subsection 4.3.5 "Additional Motion Instructions" for details of the Offset command.

Items	Descriptions
Ignore Tool-offset	Specifies whether to ignore the tool offset command (Subsection 4.3.5 "Additional motion instructions"). • If the function is DISABLED, the robot moves to the position for which the tool offset command has been executed (standard setting). • If the function is ENABLED, the robot moves to the taught position (for which the tool offset command has not been executed). See Subsection 4.3.5 "Additional Motion Instructions" for details of the Tool-offset command.
Enable VOFFSET	Specifies whether to enable the vision offset command. • If the setting is ENABLED, the robot moves to the position to which the vision offset is applied. (Standard setting) • If the setting is DISABLED, the robot moves to the originally taught position.

Procedure 3-29 Setting the general items

Step

- 1 Press the MENUS key. The screen menu is displayed.
- 2 Select 6 (SETUP).
- 3 Press F1 [TYPE]. The screen change menu is displayed.
- 4 Select General.



General item setting screen

SETUP General		JOINT 30%
		1/4
1 Break on hold:	DISABLED	
2 Current language:	DEFAULT	
3 Ignore Offset command:	DISABLED	
4 Ignore Tool_offset:	DISABLED	
5 Enable VOFFSET:	DISABLED	
[TYPE]		ENABLED DISABLED

- 5 Place the cursor on the target field, and select the function key menu.
- 6 If the value for the break on hold function is re-set, to make the new setting effective, turn the controller off and on again in cold start mode. The setting of the other functions is made effective immediately when they are reset.

3.17 OTHER SETTINGS

The other settings are specified at [6 SYSTEM Variables] on the system variable screen.

- Override restore function

Override restore function

The override restore function is a function that decreases the speed override to a prescribed value when a safety fence is opened and the *SFSPD input is turned off, but restores the speed override immediately when the safety fence is closed. This function is effective under the following conditions:

- \$SCR.\$RECOV_OVRD = TRUE. (A control start is required.)
- The system is in remote control state.
- The speed override is not changed while the safety fence is open.

Other items are set up on the system variable screen, [6 SYSTEM Variables]. To specify system variables, see the appropriate appendix (→ Appendix C, "SYSTEM VARIABLES").

4 PROGRAM STRUCTURE

This chapter describes the program structure and program instructions.

Contents of this chapter

- 4.1 PROGRAM DETAIL INFORMATION
- 4.2 LINE NUMBER, PROGRAM END SYMBOL, AND ARGUMENT
- 4.3 MOTION INSTRUCTIONS
- 4.4 PALLETIZING INSTRUCTIONS
- 4.5 REGISTER INSTRUCTIONS
- 4.6 I/O INSTRUCTIONS
- 4.7 BRANCH INSTRUCTIONS
- 4.8 WAIT INSTRUCTIONS
- 4.9 SKIP CONDITION INSTRUCTION
- 4.10 OFFSET CONDITION INSTRUCTION
- 4.11 TOOL OFFSET CONDITION INSTRUCTIONS
- 4.12 FRAME INSTRUCTIONS
- 4.13 PROGRAM CONTROL INSTRUCTIONS
- 4.14 OTHER INSTRUCTIONS
- 4.15 MULTIAXIS CONTROL INSTRUCTIONS
- 4.16 OPERATION GROUP INSTRUCTIONS

A robot application program consists of commands described by the user for the robot to perform operations and other incidental information.

In addition to program information that describes how the robot should perform operations, a program contains program detail information that defines program attributes.

Program detail		JOINT 30 %
		1/7
Creation Date:	10-MAR-1998	
Modification Date:	11-MAR-1998	
Copy Source:	[*****]	
Positions: FALSE	Size:	312 Byte
1 Program name:	[SAMPLE3]	
2 Sub Type:	[None]	
3 Comment:	[SAMPLE PROGRAM 3]	
4 Group Mask:	[1, *, *, *, *, *]	
5 Write protect:	[OFF]	
6 Ignore pause:	[OFF]	
7 Stack size	[300]	
END	PREV	NEXT

Fig. 4 (a) Program Information Screen

Program detail information consists of the following information items:

- Attribute-related information items such as a creation date, modification date, a copy source file name, presence/absence of position data, and program data size.
- Information items related to an execution environment such as a program name, sub type, comment, group mask, write protection, interruption disable and stack size.

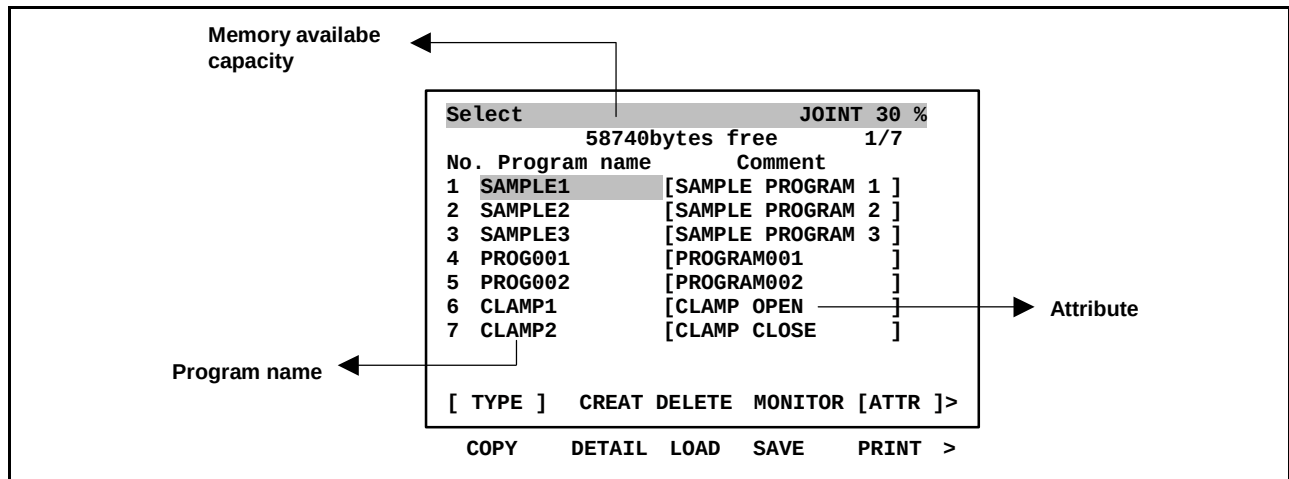


Fig. 4 (b) Program List Screen

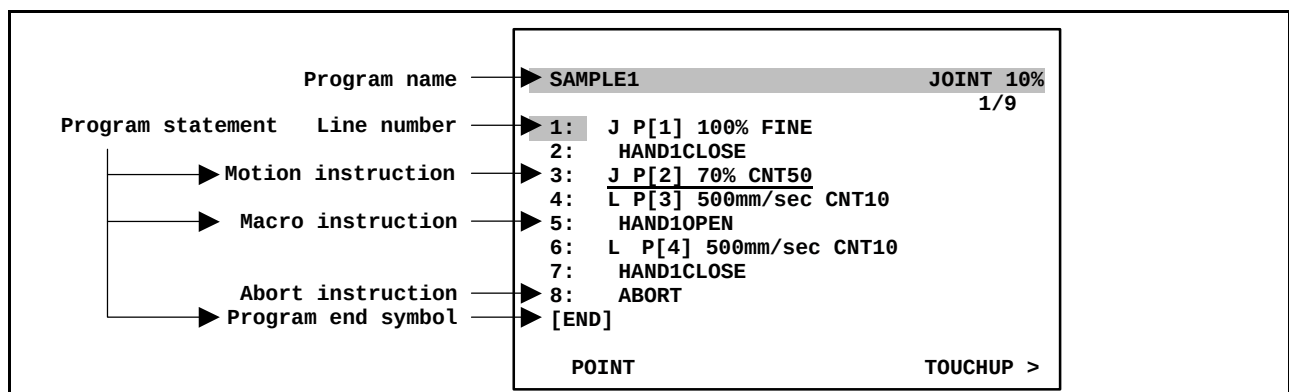


Fig. 4 (c) Program Edit Screen

A program consists of the following information:

- Line number assigned to each program command
- Motion instructions specifying how and where the robot is to move
- Program instructions including the following:
 - Palletizing instructions for performing the palletizing function
 - Instructions for storing numerical data in registers (register instructions)
 - Instructions for storing robot position data in position registers (position register instructions)
 - I/O instructions to output and input signals to and from peripheral devices
 - Branch instructions for changing the flow of program control when a defined condition is satisfied (IF, JMP/LBL, CALL/END)
 - Wait instructions for suspending program execution
 - Skip condition instruction for operating the robot until a signal is received. If the signal is not received, a branch to a specified command occurs. If the signal is received, the next command is executed, canceling the operation.
 - Program comments
 - Other instructions
- Program end symbol indicating that the program contains no more instructions

Program detail information is set on the program information screen. (See Subsection 5.3.1 and 5.5.) A program is registered on the program registration screen (See Subsection 5.3.1.) A program is created and changed on the program edit screen. (See Sections 5.3 and 5.4.)

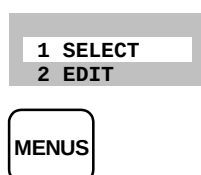
4.1 PROGRAM DETAIL INFORMATION

Program detail information names a program and defines the attributes of the program. Program detail information consists of the following items:

- Attribute-related information items such as a creation date, modification date, a copy source file name, presence/absence of position data, and program data size.
- Information items related to an execution environment such as a program name, subtype, comment, group mask, write protection and interruption disable.

The program information screen is used to set program detail information. The program information screen is displayed by selecting F2 (DETAIL) on the program list screen. (For program detail information setting, see Subsection 5.3.1 and 5.5)

Moreover, program comment, a write protection, a modification date, a memory size of the program, and a copy source settings, can be displayed on the list screen by pressing F5,[ATTR] and selecting the item from a pull up menu.



Program detail		JOINT 30 %
		1/7
Creation Date:	10-MAR-1998	
Modification Date:	11-MAR-1998	
Copy Source:		
Positions:	FALSE	Size: 312 Byte
Program name:		
1	SAMPLE3	
2	Sub Type:	[None]
3	Comment:	[SAMPLE PROGRAM 3]
4	Group Mask:	[1,*,*,*,*,*,*]
5	Write protect:	[OFF]
6	Ignore pause:	[OFF]
7	Stack size	[300]
END PREV NEXT		

4.1.1 Program Name

A program name is specified to identify a program stored in the memory of the controller. A single controller cannot contain two or more programs with the same program names.

Length

A program name must consist of one to eight characters. A unique name must be assigned to each program. In software 7DA4 or later, it is possible to use program name's length up to 36 characters

⚠ CAUTION

In software 7DA4 or later, since the program name's length can be used up to 36 characters. So the screen displayed program name is changed for 36 characters.

- Width of window screen is expanded to be able to display 36 characters
- If 36 characters cannot be displayed, the last character of program name is ">" example for "ABCDEF>" in this situation, complete program name is displayed in prompt line.

Usable characters

Character: Alphabetic characters, Number: 0 to 9. No program name can start with a number. Symbol: Underscore (_) only. The at mark (@) and asterisk (*) cannot be used.

Unusable program name

The following program names are not usable.

CON, PRN, AUX, NUL

COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9

LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, LPT9

Informative name

A program should be named so that purpose or function of the program can be known from its name.

When a program for handling workpiece A is to be named, for example, "HAND-A" is a good name to assign to the program since it indicates the function of the program.

NOTE

Observe the following when writing a program for automatic operation using RSR or PNS. Otherwise, the program will not run.

- A program using RSR must be named RSRnnnn, where nnnn is a 4-digit number. Example: RSR0001.
- A program using PNS must be named PNSnnnn, where nnnn is a 4-digit number. Example: PNS0001.

4.1.2 Program Comment

When a new program is created, a program comment can be added to the program name. A program comment is used to describe additional information to be displayed on the list screen together with the program name.

Length

A program comment must consist of one to sixteen characters.

Usable characters

Character: Alphabetic characters, Number: 0 to 9 Symbol: Underscore (_), at mark (@), and asterisk (*)

Informative comment

A program comment should describe the purpose or function of the program.

4.1.3 Subtype

Subtype is used to set a type of program. The following subtypes are available:

- **Job (JB):**
This represents a main program that can be started using a device such as a teach pendant. Process programs are called in a main program for execution.
- **Process (PR):**
This represents a subprogram that is called by a job program for execution of a particular job.
- **Macro (MR):**
This represents a program for executing a macro instruction. The subtype of a program registered on the macro instruction setting screen is automatically set to MR.
- **State:**
Specify this when creating a conditional program with the state monitoring function.

4.1.4 Group Mask

A motion group sets up an operation group of a program. An operation group represents a group of different axes (motors) used for independent robots, positioning tables, and other jigs.

NOTE

A motion group must be set before the program is executed.

The Robot controller can divide up to 56 axes (when a multifunction board is inserted) into up to 8 operation groups and control those groups simultaneously. A single group can control up to nine axes (multimotion function). If the system has only one operation group, the default motion group is group 1 (1, *, *, *, *, *, *, *, *).

For a program that has no motion group (that is, a program involving no robot motion), this item is to be specified as (*, *, *, *, *, *, *, *). A program that has no motion group can be started even when the system is not ready for operation.

The system is ready for operation when the following ready conditions are satisfied:

- The peripheral I/O, ENBL input, is on.
- The peripheral I/O, SYSRDY output, is on (With the servo power is on).

4.1.5 Write Protection

Write protection specifies whether the program can be modified.

- When this item is set to ON, no data can be added to the program, and the program cannot be modified; that is, the program is write protected. When a program has been created, and its operation is confirmed, the user can set this item to ON to prevent the program from being modified by the user or someone else.

NOTE

When this item is set to ON, other items in the program detail information (Program name, Comment, Sub Type, Group Mask, Ignore pause) cannot be changed.

- When this item is set to OFF, the program can be modified; that is, program instructions can be added to the program, and existing instructions can be modified. Write protection is normally set to OFF as standard.

4.1.6 Interruption Disable

Interruption disable (ignore pause) prevents a program being executed and not having the motion group from being interrupted by an alarm (with a severity of SERVO or lower), emergency stop, or halt. When these signals are to be ignored, set interruption disable to ON.

When interruption disable is set to ON, a program being executed can only interrupted by an abort instruction in the program or an alarm with a severity higher than SERVO. (See Subsection 4.13.2.)



WARNING

When interruption disable is set to ON, a program being executed cannot be interrupted by pressing the emergency stop or halt button on the teach pendant or operator's panel.

4.1.7 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 the following stack overflow alarm occurs, the program needs more stack. Please increase stack size of the main program that calls the sub program. Please set the stack size to 300 and try to execute program at first. If the following alarm occurs, please increase the stack size to 500, 800 and 1000 in order.

Options such as TCP Mate and Cell Finder require that teach pendant programs have increased stack size.

INTP-222Call program failed.

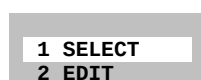
INTP-302Stack overflow.

The minimum and default stack size is 300. The maximum stack size is 4000.

Procedure 4-1 Program Detail Information

Step

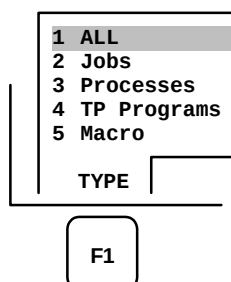
- 1 Press the MENUS key. The screen menu is displayed.
- 2 Select 1(SELECT). The program list screen is displayed.
The program list screen can be displayed by pressing the SELECT key without using above steps.



Select		JOINT 30 %
58740 bytes free		1/7
No.	Program name	Comment
1	SAMPLE1	[SAMPLE PROGRAM 1]
2	SAMPLE2	[SAMPLE PROGRAM 2]
3	SAMPLE3	[SAMPLE PROGRAM 3]
4	PROG001	[PROGRAM001]
5	PROG002	[PROGRAM002]
6	CLAMP1	[CLAMP OPEN]
7	CLAMP2	[CLAMP CLOSE]
[TYPE] CREATE DELETE MONITOR [ATTR]>		
COPY DETAIL LOAD SAVE PRINT >		

- Switching the screen using sub type

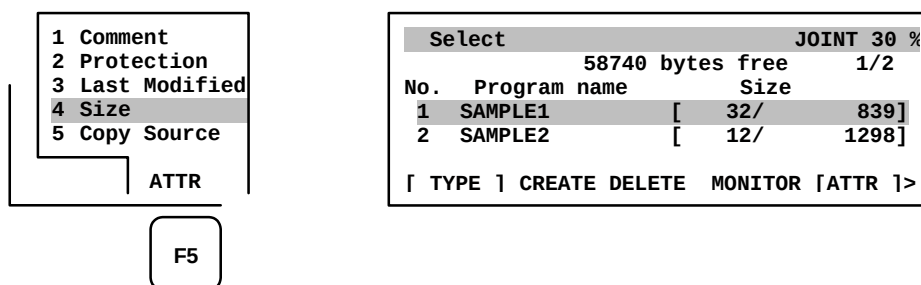
- 3 To specify the program to be displayed for the sub type, press F1,[TYPE] and select the sub type of the program you want to display.
 - All : All the programs are displayed.
 - Job : Only job programs are displayed.
 - Process : Only process programs are displayed.
 - Program : All the programs except the macro are displayed.
 - Macro : Only macro programs are displayed.



SAMPLE1	LINE 1	ABORT
Select		JOINT 30 %
61276 bytes free		1/2
1	SAMPLE1	[SAMPLE PROGRAM 1]
2	SAMPLE2	[SAMPLE PROGRAM 2]
[TYPE] CREATE DELETE MONITOR [ATTR]>		

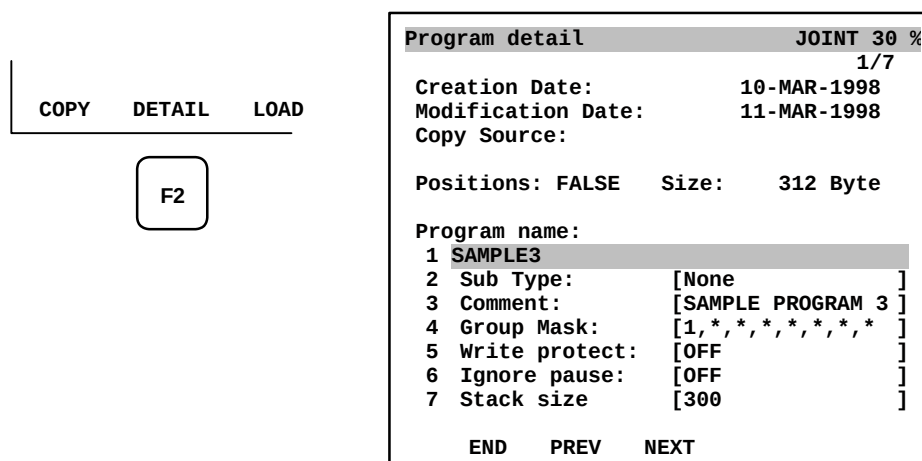
- Switching the display using the attribute

- 4 To specify the program attribute to be displayed, press F5,[ATTR] and select the attribute type of the program you want to display.
 - Comment : The comment is displayed.
 - Protection : The setting of the write protection is displayed.
 - Last Modified : The latest date of the modification is displayed.
 - Size : The number of the line and the program size are displayed.
 - Copy Source : The name of the copy source program is displayed.

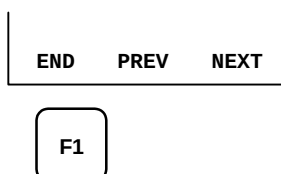


- Program Detail Screen

- 5 Press NEXT, > and press F2,DETAIL in the next page. The program detail screen is displayed.



- 6 When you finish setting the program header information, press F1,END.



4.2 LINE NUMBER, PROGRAM END SYMBOL, AND ARGUMENT

Line number

A line number is automatically inserted in front of an instruction when it is added to a program. When an instruction is deleted, or an instruction is moved to another location, the lines of the program are renumbered in ascending order; that is, the first line is numbered as 1, the second line is numbered as 2, and so forth.

When a program is to be modified, the cursor can be used to specify a line or a range of lines for movement or deletion by line number.

The user can make the cursor move to a desired line number by specifying a line number (with the ITEM key).

Program end symbol

The program end symbol ([End]) is automatically displayed on the line after the last instruction of a program. Whenever a new instruction is added, the program end symbol moves downward on the screen. As a result, it is always displayed on the last line.

When the execution of a program reaches the program end symbol after the last instruction in the program is executed, the program execution automatically returns to the first line of the program for termination. However, when the setting of "Return to top of program" is FALSE, the cursor stays at the last line of the program after program execution is completed. (See Section 3.15 "SYSTEM CONFIG MENU".)

A description of the program instructions required to create and change a program follows. (For how to create a program, see Section 5.3. For how to change a program, see Section 5.4.)

Argument i

Argument i is an index used in teaching control instructions (program instructions other than motion instruction). Some arguments are specified directly; others are specified indirectly. In direct specification, an integer from 1 to 32766 is usually specified. The range of values used depends on the type of instruction. In indirect specification, the register number of a register is specified.

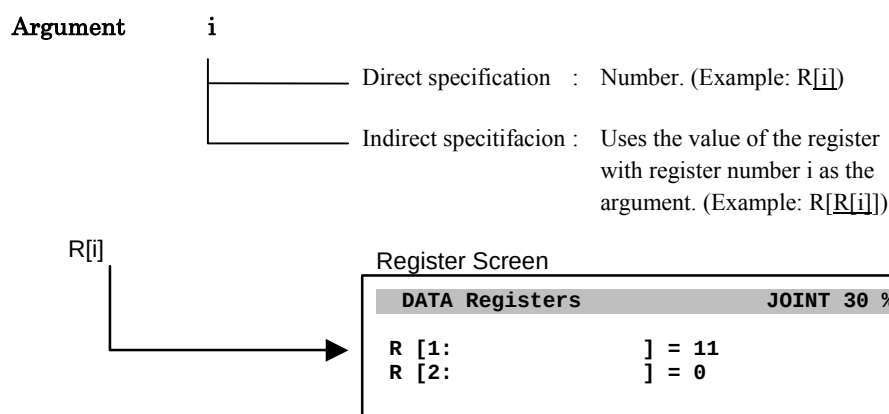


Fig. 4.2 Format of Argument i

Procedure 4-2 Program Edit Screen

Condition

- The teach pendant must be enabled.

Step

- 1 Display the program list screen.
- 2 Move the cursor to the program you want to edit and press ENTER key. The program edit screen is displayed.

Select 61276 No. Program name 1 SAMPLE1 2 SAMPLE2 3 SAMPLE3 4 PROG001 ENTER	SAMPLE1 JOINT 30 % 1/6 1: J P[1] 100% FINE 2: J P[2] 70% CNT50 3: L P[3] 100cm/min CNT30 4: L P[4] 500mm/sec FINE 5: J P[5] 100% FINE [END] POINT TOUCHUP >

- Moving the cursor

- 3 To move the cursor, use the arrow keys such as up, down right, and left. To move quickly through the information, press and hold the SHIFT key and press the down or up arrow keys.
- 4 To select the line number, press the ITEM key and enter the line number you want to move the cursor.

ITEM 5 ENTER	SAMPLE1 JOINT 30 % 5/6 4: L P[4] 500mm/sec FINE 5: J P[5] 100% FINE [END]

- Entering the numerical value

- 5 To enter the numerical value, move the cursor to the argument and press the numerical value keys. When you are finished, press the ENTER key.

PROG2 9: L P[5] 100% FINE 10: DO[...]=... 1 ENTER	PROG2 JOINT 30 % 10/11 10: DO[1]=... [END] Enter value DIRECT INDIRECT [CHOICE]

- 6 To use the indirect addressing with the register, press F3,INDIRECT.

DIRECT INDIRECT F3	PROG2 JOINT 30 % 10/11 10: DO[R[1]]=... [END] Enter value DIRECT INDIRECT [CHOICE]

4.3 MOTION INSTRUCTIONS

A motion instruction moves a robot tool to a specified point within the operating area at a specified feedrate and in a specified traveling mode. The items listed below must be specified in a motion instruction. The format of a motion instruction is shown in Fig. 4.3.

- Motion format: Specifies how to control the path of motion to a specified position.
- Position data: Teaches a position to which the robot is to move.
- Feedrate: Specifies the feedrate of the robot.
- Positioning path: Specifies whether to position the robot at a specified point.
- Additional motion instruction: Specifies the execution of an additional instruction while the robot is in motion.

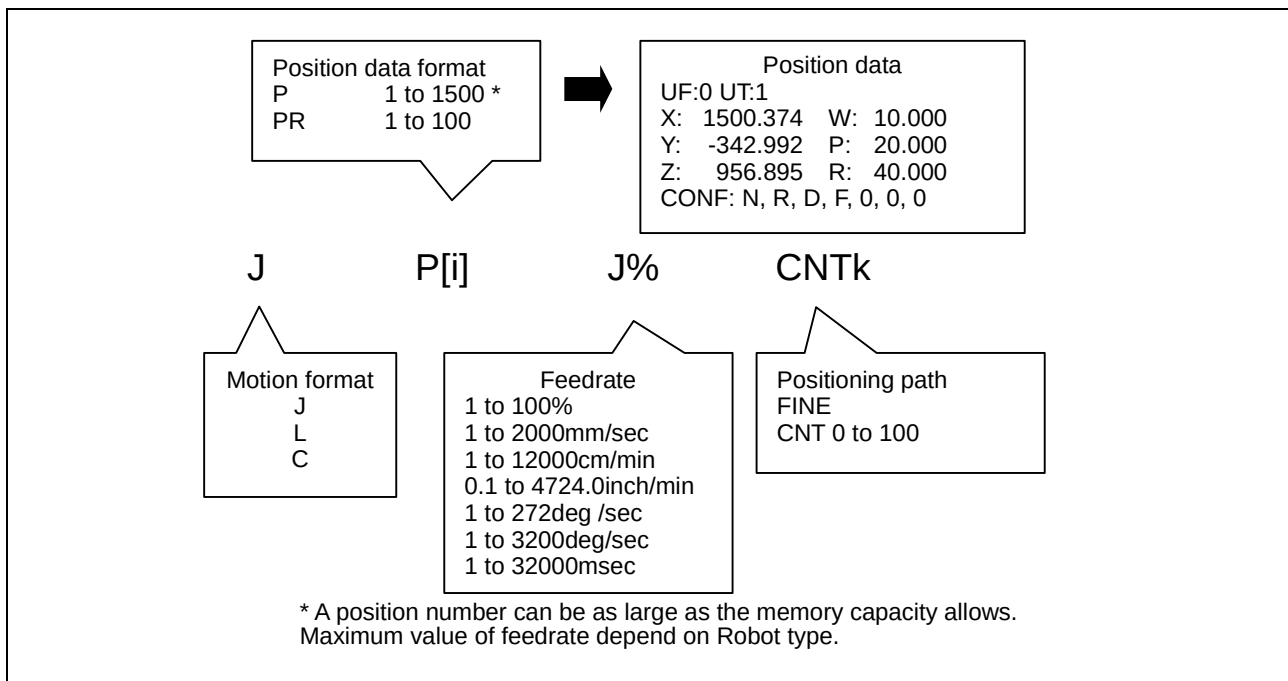


Fig. 4.3 Motion Instructions

In teaching a motion instruction, a standard motion instruction is selected using either the F1 or F5 function key. (For modifying a standard motion instruction, see Subsection 5.3.2. For teaching a motion instruction, see Subsection 5.3.3. For changing a motion instruction, see Subsection 5.4.2.)



- Press F1 POINT to program an operation instruction.
- Press F5 TOUCHUP to re-program programmed position data.

4.3.1 Motion Format

For the motion format, the path of motion to a specified position is specified. Three options are available: joint motion, which does not exercise path/attitude control and linear motion and circular motion, which exercise path/attitude control.

- Joint motion (J)
- Linear motion (including the rotation motion)(L)
- Circular motion (C)

Joint motion J

The joint motion mode is the basic mode for moving the robot to a specified position. The robot accelerates along or about all axes, moves at a specified feedrate, decelerates, and stops at the same time. The path of motion is usually non-linear. The motion format is specified to teach an end point. A percentage of a maximum feedrate is specified as the feedrate of joint motion. The attitude of a tool being moved is not controlled.

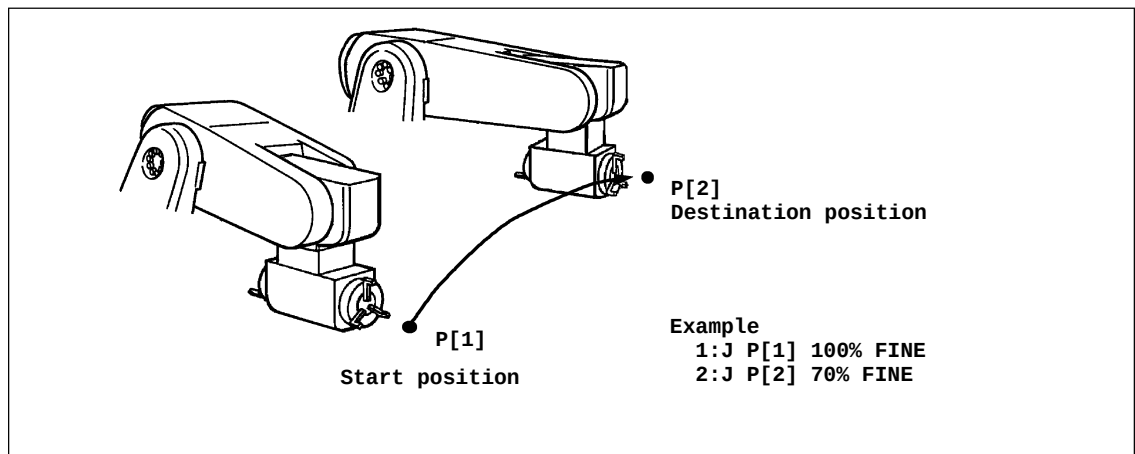


Fig. 4.3.1 (a) Joint Motion

Linear motion L

The linear motion mode controls the path of tool center point (TCP) motion from a start point to an end point; the tool center point moves linearly. The motion format is specified to teach an end point. For linear feedrate specification, a desired option must be chosen from mm/sec, cm/min, and inch/min. The attitude of a tool being moved is controlled by dividing the attitude at a start point from the attitude at a target point.

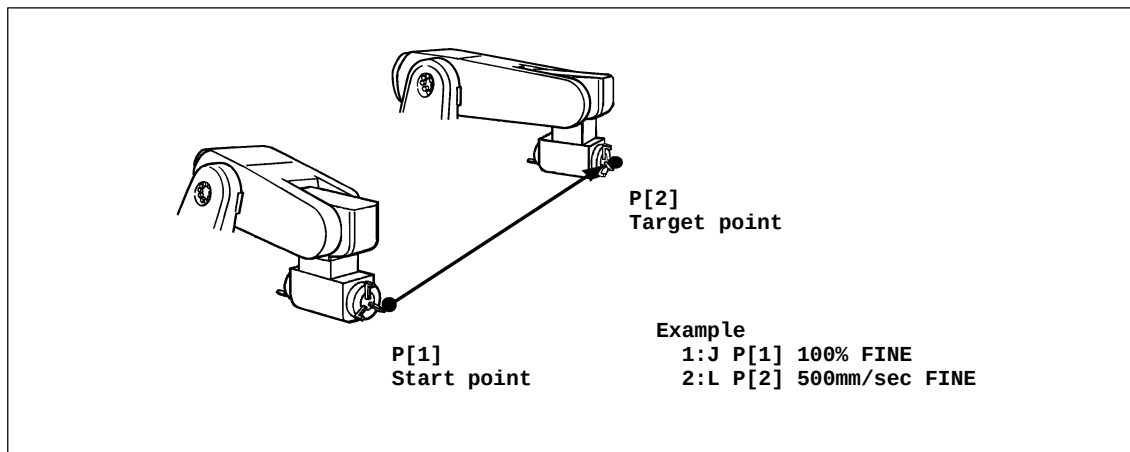


Fig. 4.3.1 (b) Linear Motion

Rotary operation is a method of travel in which the tool is rotated about the tool endpoint from the start position to the end position by using linear operation. The orientation of the tool during travel is controlled by dividing the orientation at the start position and that at the destination position. The feedrate is specified in deg/sec. The path is controlled linearly (if the tool endpoint moves).

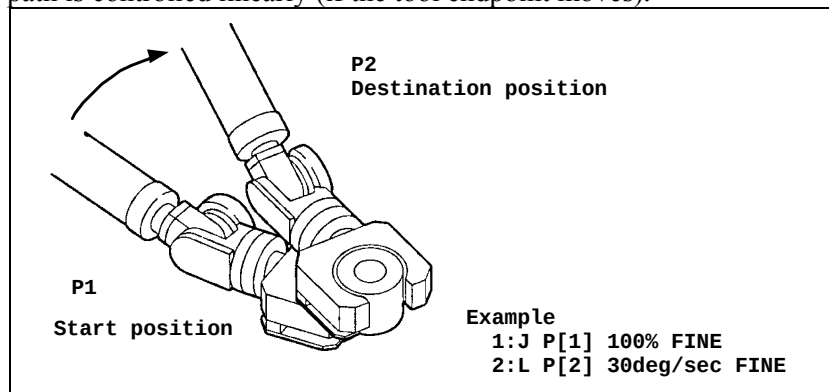


Fig. 4.3.1 (c) Rotation Motion

Circular motion

The circular motion mode controls the path of tool center point motion from a start point to an end point through a passing point.

Both a passing point and a target point are taught in one instruction.

For circular feedrate specification, a desired option must be chosen from mm/sec, cm/min and inch/min.

The attitude of a tool being moved is controlled by dividing the attitude at a start point from the attitude at a target point.

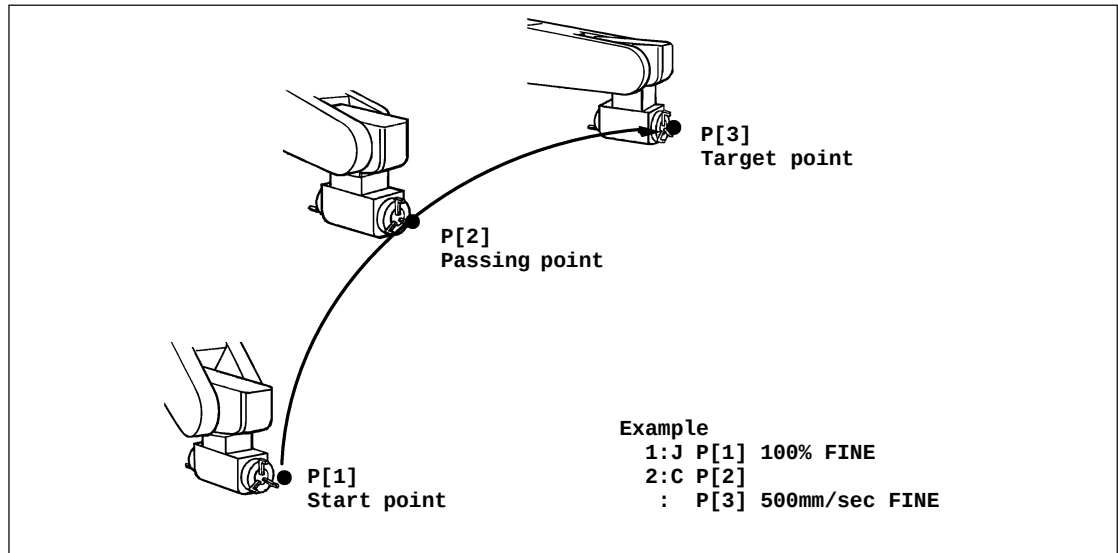


Fig. 4.3.1 (d) Circular motion

Motion of 5-axis robot

Since 5-axis robot cannot move under a full control of its Tool attitude, there are many attitudes which the robot cannot reach. For this reason, the motion of 5-axis robot has some unique characteristics.

- 1 During Linear motion or Circular motion, 5-axis robot can exactly control its Tool attitude ONLY IF the flange surface faces in the vertical direction. If not, 5-axis robot moves in an alternative “reachable” Tool attitude. This means the robot does not keep the same Tool attitude during the motion. On the other hand, 5-axis robot always keep its TCP position exactly.
- 2 5-axis robot has a special definition of Configuration; Joint placement of wrist axis. (See section 4.3.2 for Configuration.)
 - (1) M-410iB/140H: If the flange surface faces downward, the configuration is NOFLIP. If not, the configuration is FLIP.
 - (2) F-100iA 5-axis: The definition of FLIP / NOFLIP is special.
 - (3) Other 5-axis robot : If the flange surface faces inward, the configuration is NOFLIP. If not, the configuration is FLIP.

If Configuration of a start point is different from that of a destination point, 5-axis robot cannot perform any Cartesian motion (Linear motion and Circular motion). In this case, use a Joint motion instead or add Wrist joint motion instruction to the Cartesian motion. (See section 4.3.5 for Wrist joint motion instruction.)

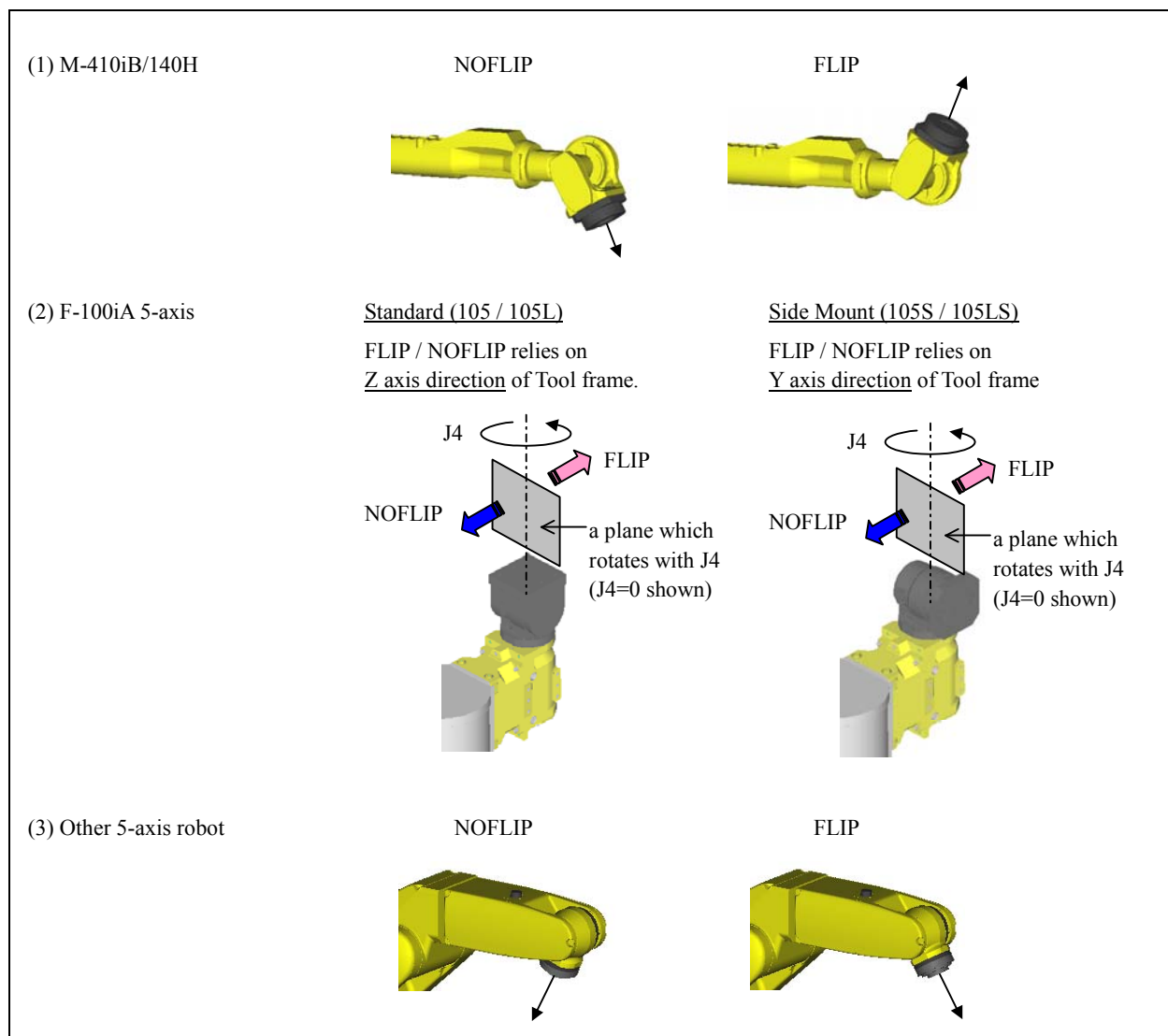


Fig. 4.3.1 (e) Definition of configuration for 5-axis robot

- 3 Group (3) 5-axis robot switches the attitude control method according to the following condition;
 (A) : The flange surface faces in the vertical direction.
 (B) : Other than condition (A).

For this reason, 5-axis robot cannot perform a Cartesian motion (Linear motion and Circular motion) if the condition changes from (A) to (B) or from (B) to (A) during the motion. If you teach and start such a motion, an alarm "MOTN-063 Position config change" or "MOTN-310 Pos. Cfg. change 2" occurs.

When the servo power turns off due to an emergency stop input etc., sometimes robot axes slip slightly and the condition changes from (A) to (B). Then, you may see the alarm described above. In this case, resume the program as follows;

Step1: Abort the program.

Step2: Jog the robot to recover condition (A).

Step3: Resume the program.

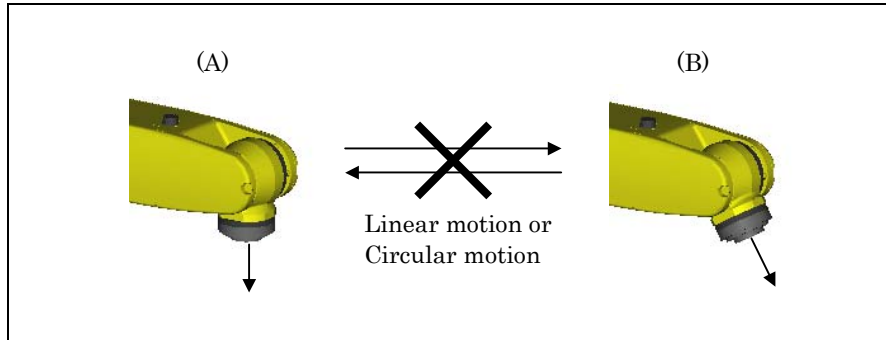


Fig. 4.3.1 (f) Prohibition of linear and circular motion that the attitude control method is switched

4.3.2 Position Data

Position data includes the positions and attitudes of the robot. When a motion instruction is taught, position data is written to the program at the same time.

Position data is classified into two types. One type consists of joint coordinates in a joint coordinate system. The other type consists of Cartesian coordinates representing tool positions and attitudes in work space. Standard position data uses Cartesian coordinates.

Cartesian coordinates

Position data consisting of Cartesian coordinates is defined by four elements: the position of the tool center point (origin of the tool coordinate system) in a Cartesian coordinate system, the inclination of the axis along which the tool moves (tool coordinate system), configuration, and a Cartesian coordinate used. A Cartesian coordinate system may be a world coordinate system. How to select the coordinate systems is explained later in this subsection.

<u>UF</u>	<u>UT</u>	(<u>X, Y, Z</u>	<u>W, P, R</u>	<u>Configuration</u>
User coordinate system number	Tool coordinate system number	Position	Attitude	Configuration

Fig. 4.3.2 (a) Position Data (Cartesian Coordinates)

Position and attitude

- The position (x,y,z) represents the three-dimensional position of the tool center point (origin of the tool coordinate system) in the Cartesian coordinate system.
- The attitude (w,p,r) represents angular displacements about the X-axis, Y-axis, and Z-axis in the Cartesian coordinate system.

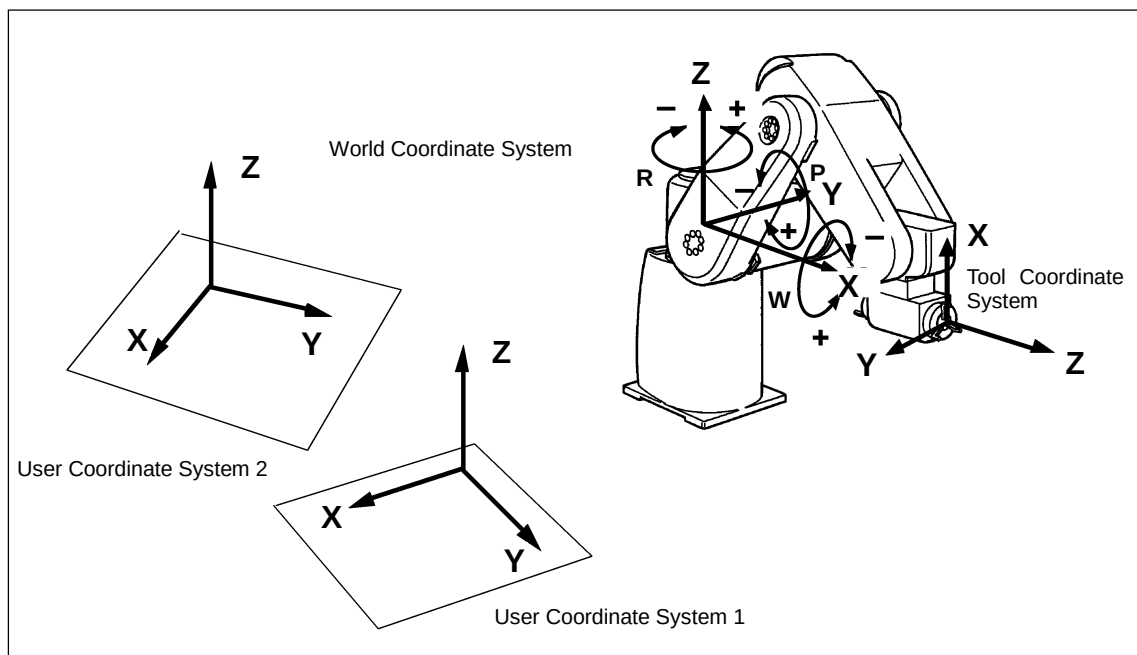
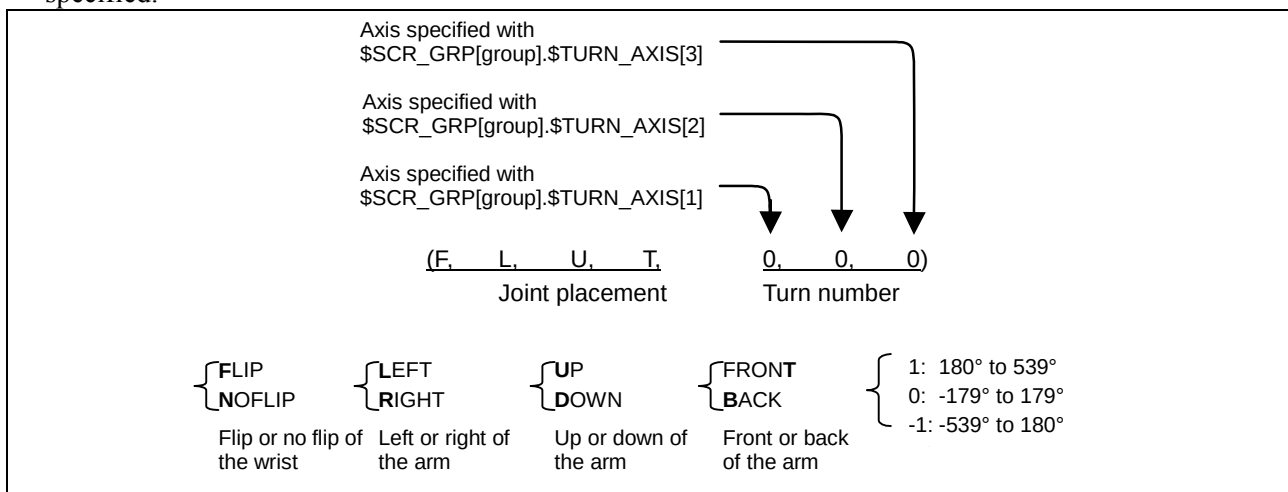


Fig. 4.3.2 (b) World Coordinate System/Tool Coordinate System

Configuration

A configuration represents the attitude of the robot. Several configurations are available which meet the condition of Cartesian coordinates (x,y,z,w,p,r). The turn number and joint placement of each axis must be specified.



- Joint placement

Joint placement specifies the placement of the wrist and arm. This specifies which side the control point of the wrist and arm is placed on against the control plane. When a control point is placed on the control plane, the robot is said to be placed at a singular, or to be taking a peculiar attitude. At the singular, since the configuration can not be decided to one by the specified cartesian coordinate values, the robot can not move.

- An operation that ends at a singular point cannot be programmed. (In some cases, the most feasible configuration can be selected.) To specify such an operation, define the joint coordinate values.
- During linear or circular motion, the tool cannot pass through a singular point (the joint placement cannot be changed). In this case, execute a joint motion. To pass through a singular point on the wrist axis, a wrist joint motion (Wjnt) can also be executed.

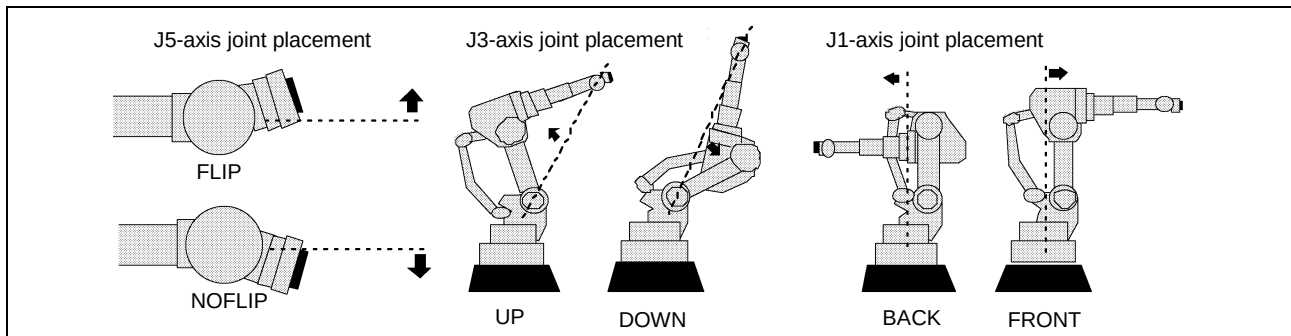


Fig. 4.3.2 (d) Joint Placement

Turn number

Turn number represents the number of revolutions of the wrist axis (J4, J5, J6). Each axis returns to the original position after one revolution. So, specify how many turns have been made. Turn number is 0 when each axis is at an attitude of 0.

The turn numbers for up to three axes can be displayed. The axis number to correspond to each field is specified with system variable \$SCR_GRP[i].\$TURN_AXIS[j] (where i is a group number), as follows:

- Left field : Axis number specified with \$SCR_GRP[i].\$TURN_AXIS[1]
- Middle field : Axis number specified with \$SCR_GRP[i].\$TURN_AXIS[2]
- Right field : Axis number specified with \$SCR_GRP[i].\$TURN_AXIS[3]

When programmed linear motion or circular motion is executed, the robot tool moves toward the target point while adopting an attitude very similar to that at the start point. The number of revolutions performed at the target point is selected automatically. The actual number of revolutions performed at the target point may differ from the number specified in the position data.

Cartesian coordinate system reference

In playback of position data consisting of Cartesian coordinates, a Cartesian coordinate system reference checks the coordinate system number of a Cartesian coordinate system to be used.

If the coordinate system number (a number from 0 to 10) specified in the position data does not match the coordinate system number currently selected, the program is not executed for safety, and an alarm is issued. A coordinate system number is written into position data in position teaching.

To change a coordinate system number after it has been written, use the tool replacement/coordinate replacement shift function.

Tool coordinate system number (UT)

The tool coordinate system number specifies the coordinate system number of a mechanical interface coordinate system or tool coordinate system. Thus, the coordinate system of the tool is determined.

- 0 : The mechanical interface coordinate system is used.
- 1 to 10 : The tool coordinate system of a specified tool coordinate system number is used.
- F : The coordinate system of the tool coordinate system number currently selected is used.

User coordinate system number (UF)

The user coordinate system number specifies the coordinate system number of a world coordinate system or user coordinate system. Thus, the coordinate system of work space is determined.

- 0 : The world coordinate system is used.
- 1 to 9 : The user coordinate system of a specified user coordinate system number is used.
- F : The coordinate system of the user coordinate system number currently selected is used.

Detail position data

To display the detail position data, position the cursor to the position number, then press the F5 (POSITION) key.

SAMPLE1			Position Detail						JOINT 30 %
1: J P[1]	100%		P[2]	GP:1	UF:0	UT:1	CONF:N	T, 0	
2: J P[2]	70%		X:	1500.374 mm	W:	40.000 deg			
			Y:	-242.992 mm	P:	10.000 deg			
			Z:	956.895 mm	R:	20.000 deg			
COMMENT CHOICE POSITION			SAMPLE1						

F5

Joint coordinates

Position data consisting of joint coordinates is defined using angular displacements with respect to the joint coordinate system on the base side of each joint.

(J1, J2, J3, J4, J5, J6, E1, E2, E3)
 Main axis Wrist axis Additional axis

Fig. 4.3.2 (e) Position Data (Joint Coordinates)

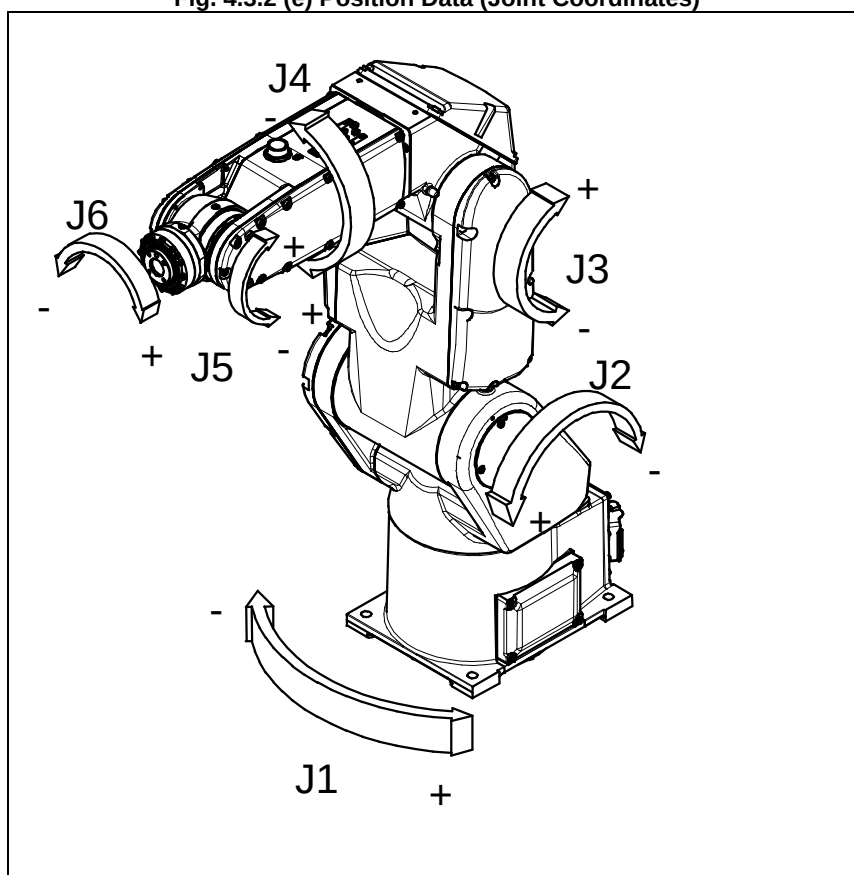
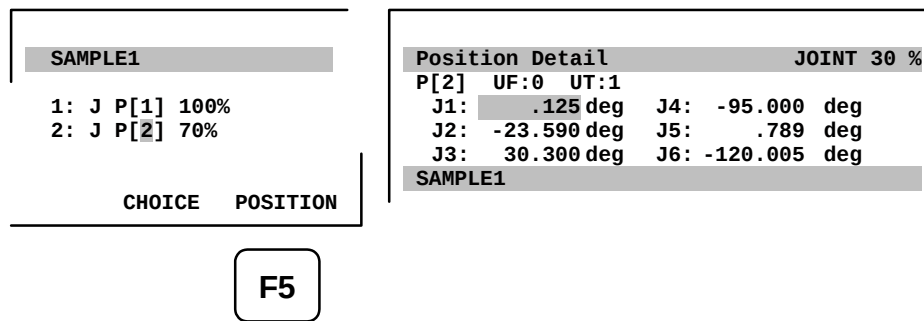


Fig. 4.3.2 (f) Joint Coordinate System

Detail position data

Detailed position data is displayed when F5 POSITION is pressed. You can choose between Cartesian coordinates and joint coordinates by pressing F5 [REPRE].



Position variable and position register

In a motion instruction, position data is represented by a position variable (P[i]) or position register (PR[i]). Usually, a position variable is used.

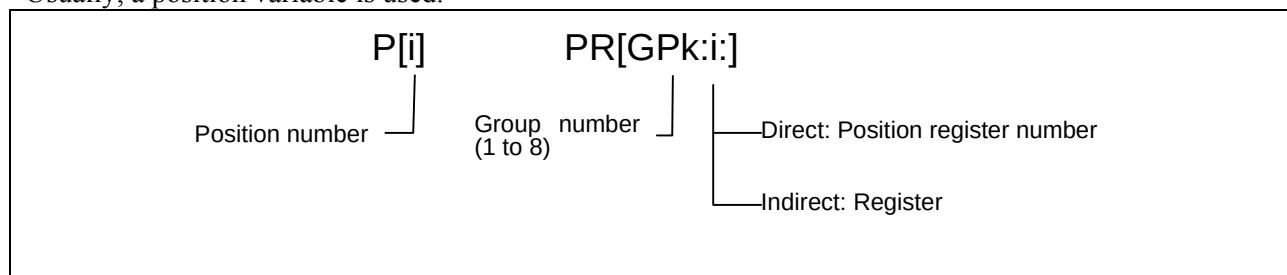


Fig. 4.3.2 (g) Position Variable and Position Register

Example 1: J P[12] 30% FINE
 2: L PR[1] 300mm/s CNT50
 3: L PR[R[3]] 300mm/s CNT50

- Position variable

The position variable is the variable usually used to hold position data. In motion instruction teaching, position data is automatically saved.

When Cartesian coordinates are taught, the following Cartesian coordinate system and coordinate system number are used:

- Coordinate system of the tool coordinate system number currently selected (UT = 1 to 10)
- Coordinate system of the user coordinate system number currently selected (UF = 0 to 9)

In playback, the following Cartesian coordinate system and coordinate system number are used:

- Coordinate system with the specified tool coordinate system number (UT = 1 to 10)
- Coordinate system with the specified user coordinate system number (UF = 0 to 9)

- Position register

The position register functions as a general-purpose register for holding position data. (For position teaching using a position register, see Section 7.4.)

When Cartesian coordinates are taught, the following Cartesian coordinate system and coordinate system number are used:

- Coordinate system of the tool coordinate system number currently selected (UT = F)
- Coordinate system of the user coordinate system number currently selected (UF = F)

In playback, the following Cartesian coordinate system and coordinate system number are used:

- Coordinate system of the tool coordinate system number currently selected (UT = F)
- Coordinate system of the user coordinate system number currently selected (UF = F)

It is possible to move only the specified motion group by specifying the group number in the position register.

- Position number

The position number is used to reference a position variable. A position number is automatically assigned each time a motion instruction is taught and it is reflected in the program. For example, the first position number assigned is P[1], the second P[2], and so on.

When a motion instruction is added, it is assigned the position number obtained by incrementing the position number assigned to the most recently added motion instruction by one, regardless of where the newly added instruction is placed in the program. However, this is not the case when a position number is changed.

When a position is deleted, the position numbers of other taught points remain unchanged. However, this is not the case when a position number is changed. (For changing a position number, see Section 5.4 "Changing a Program".)

A comment consisting of up to 16 characters can be described for a position number or position register number. To add a comment, press the ENTER key when the cursor is at the position number or position register number.

Example

```
4: J P[11: APPROACH POS ] 30% FINE
5: L PR[1: WAIT POS ] 300mm/s CNT50
```

4.3.3 Feedrate

The feedrate specifies the speed at which the robot moves. During program execution, the feedrate is controlled by feedrate override. A feedrate override value of 1% to 100% can be used.

The unit used to specify a feedrate depends on the motion format taught with a motion instruction.

NOTE

The programmed traveling speed cannot exceed the allowable range of the robot. If a speed exceeding the range is programmed, a warning alarm would be issued.

J P[1] 50% FINE

When the motion type is joint, a feedrate is specified as the following:

- A percentage from 1% to 100% of the maximum feedrate is to be specified.
- When the unit is sec, specify the value from 0.1 to 3200sec as the time took for motion. This specification is required, when the time took for motion is important. An operation cannot sometimes takes place in a specified time.
- When the unit is msec, specify the value from 1 to 32000msec as the time took for motion.

L P[1] 100mm/sec FINE

If the specified motion format is linear motion or circular motion, specify a feedrate as follows:

- When the unit is mm/sec, specify a feedrate from 1 to 2000 mm/sec.
- When the unit is cm/min, specify a feedrate from 1 to 12000 cm/min.
- When the unit is inch/min, specify a feedrate from 0.1 to 4724.4 inch/min.
- When the unit is sec, specify the value from 0.1 to 3200sec as the time took for motion.
- When the unit is msec, specify the value from 1 to 32000msec as the time took for motion.

L P[1] 50deg/sec FINE

When the mode of motion is rotation about the tool center point, specify an angular displacement as follows:

- When the unit is deg/sec, specify an angular displacement from 1 to 272 deg/sec.
- When the unit is sec, specify the value from 0.1 to 3200sec as the time took for motion.
- When the unit is msec, specify the value from 1 to 32000msec as the time took for motion.

Specifying the feedrate with a register

The feedrate can be specified with a register. This allows the user to specify the feedrate for an operation instruction after calculating the feedrate using a register. The feedrate can also be specified externally, using group input (GI) or data transfer, for example.



CAUTION

This function allows the user to change the feedrate of a robot freely by setting a register. This means that the robot may operate at an unexpected speed depending on the specified register value. When using this function, therefore, specify the register value with great care during both teaching and operation.

Format in which an operation instruction is displayed when the feedrate is specified with a register

- Joint J P[1] R[i]% FINE
- Linear L P[1] R[i]mm/sec FINE
- Circular C P[1]
 P[2] R[i]mm/sec FINE
- Pallet operation instruction
 J PAL_1[A_1] R[i]% FINE
 J PAL_1[BTM] R[i]% FINE
 J PAL_1[R_1] R[i]% FINE

NOTE

The pallet operation instruction is a software option of palletizing.

Operation group instruction

Asynchronous operation group

GP1 JP[1] R[i]% FINE

GP2 JP[1] R[i]% FINE

NOTE

The operation group instruction is a software option of multimotion.

The feedrate for a standard operation instruction is also supported.

Search/replace functions

- Search function

The search function is not supported.

Search using register items cannot be performed.

- Replace function

Replacement is possible with the operation statement modification item.

Replacement using register items cannot be performed.

The additional axis feedrate for an operation addition instruction is not supported.

In program editing, a range check is not performed on the feedrate (register value).

The feedrate (register value) is not automatically converted when the feedrate unit is changed.

If the feedrate specification for an operation statement is made with a register, the read-ahead of execution is stopped. (It is possible to specify whether to stop read-ahead using a system variable. This is described later.)

If the value entered in the register is not within the upper and lower limits, or if the value is of a type other than those appropriate to a feedrate (integer/real), an alarm is generated during execution.

Unit	Allowable range	
%	1 to 100	Integer
sec	0.1 to 3200.0	(*1) Real/effective up to the first decimal place.
Msec	1 to 32000	(*1) Integer
mm/sec	1 to 2000	(*1) Integer
cm/min	1 to 12000	(*1) Integer
inch/min	0.1 to 4724.2	(*2) Real/effective up to the first decimal place.
deg/sec	1 to 272	(*3) Integer

The allowable range (maximum value) differs depending on the robot type.

*1: System variable \$MPR_GRP.\$SPPEDLIM

*2: System variable \$MPR_GRP.\$SPPEDLIM/10

*3: System variable \$MPR_GRP.\$ROTSPEEDLIM * 180/3.1415

Read-ahead can be enabled.

If the feedrate specification for an operation statement is made with a register, the read-ahead of execution is stopped. It is possible to specify whether to stop read-ahead using the following system variable. The default is FALSE (read-ahead is stopped).

\$RGSPD_PREXE = TRUE: Enables read-ahead.
= FALSE: Disables read-ahead.

NOTE

If the read-ahead of the register feedrate is enabled with the above system variable, it is possible that the new value is not reflected in the operating speed, causing the robot to move with the old value, depending on the timing at which the register value is changed. If read-ahead of the register feedrate is enabled, it is necessary to take appropriate measures such as interlocking or not changing the value of the register used for the feedrate during program execution.

```
10: R [1] = 100
11: J P[5] R[1]% FINE
12: R[1]=10
13: J P[6] R[1]% FINE
```

If read-ahead is enabled, 100 on line 10, not 10 on line 12, is used for the operating speed on line 13.

4.3.4 Positioning Path

The positioning path defines the method of ending robot operation in a motion instruction. Two positioning path modes are available:

- FINE positioning path
- CNT positioning path

FINE positioning path J P[i] 50% FINE

When the FINE positioning path is specified, the robot stops at a target point before moving to the next target point.

CNT positioning path J P[i] 50% CNT50

When the CNT positioning path is specified, the robot approaches a target point but does not stop at the point and moves to the next point.

How closely the robot must approach a target point can be defined by specifying a value from 0 to 100.

When 0 is specified, the robot moves the nearest path to the destination position but moves to the next target point without stopping at the target point. When 100 is specified, the robot moves along the farthest path to

the target point because the robot does not decelerate near the target point and it starts to move to the next target point soon.

NOTE

- 1 When an instruction such as a wait instruction is taught, the robot stops at the target point to execute that instruction.
- 2 Several short-distance, high-speed motions that are performed continuously with CNT specified may be decelerated, even if the specified CNT value is 100.

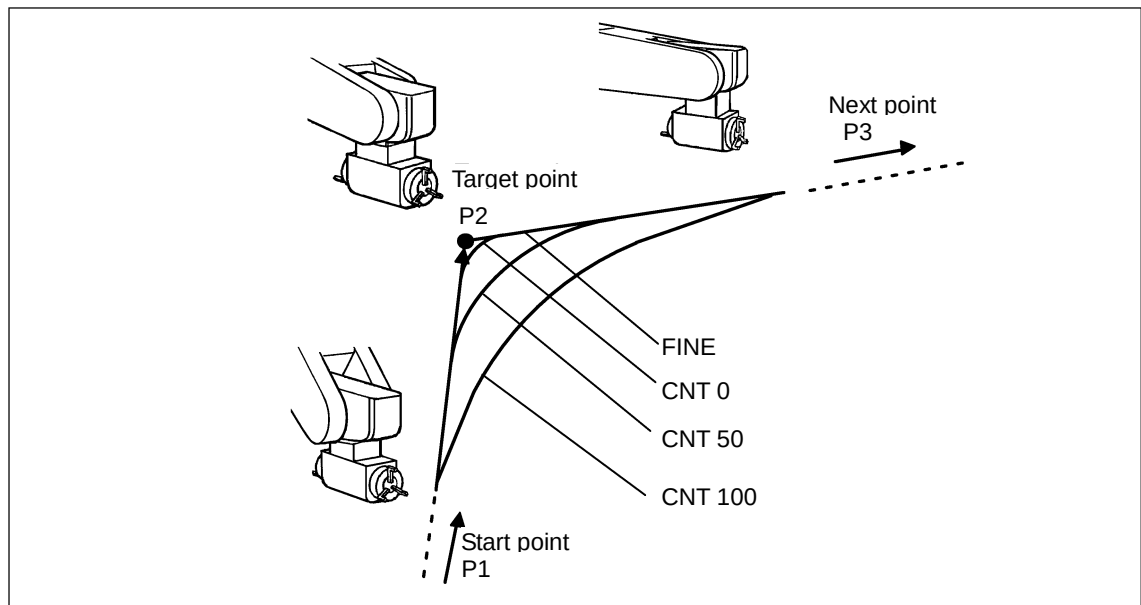


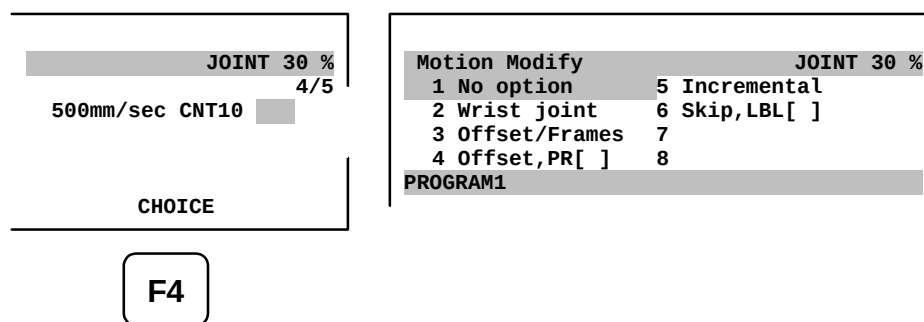
Fig. 4.3.4 Robot motion path using continuous termination type

4.3.5 Additional Motion Instructions

An additional motion instruction causes the robot to perform a particular job. The following additional motion instructions are available:

- Wrist joint motion instruction (Wjnt)
- Acceleration override instruction (ACC)
- Skip instruction (Skip,LBL[i])
- Offset condition instruction (Offset)
- Direct offset condition (Offset,PR[i])
- Tool offset instruction (Tool_Offset)
- Direct tool offset instruction (Tool_Offset, PR[i])
- Incremental instruction (INC)
- Simultaneous EV instruction (EV i%)
- Independent EV instruction (Ind.EV i%)
- Path instruction (PTH)
- Pre-execution instruction (pre-execution/post-execution)
(→Section 9.8, "Pre-execution Instruction")
- BREAK instruction (BREAK)

When teaching an additional motion instruction, move the cursor after the motion instruction, then press the F4 [CHOICE] to display the list of additional motion instructions. Then select a desired additional motion instruction.



Wrist joint motion instruction L P[i] 50% FINE Wjnt

Motion Modify	
1 No option	5 In
2 Wrist joint	6 Sk.
3 Offset/Frames	7
4 Offset, PR[]	8

The wrist joint motion instruction specifies a path control operation that does not control the attitude of the wrist. (In the standard mode, the attitude of the wrist is controlled until the end of the motion.) The wrist joint motion instruction is used when a linear motion or circular motion is specified.

When the wrist joint motion instruction is used, the attitude of the wrist changes during the motion. However, the tool center point can move along a programmed path without causing the wrist axis to invert due to a wrist axis singular point.

Acceleration override

J P[1] 50% FINE ACC80

Motion Modify	
1 No option	
2 Wrist joint	
3 ACC	
4 Skip, LBL[]	

This instruction specifies the percentage of the acceleration/deceleration rate during motion. Basically, this function is for making motion slower. When the acceleration override is reduced, acceleration time will be long (Acceleration and deceleration are done slowly). To perform a potentially dangerous operation such as hot water scooping, use a value less than 100%. When acceleration override is raised, acceleration time will be short (Acceleration and decelerate are done quickly). For portions where the operation is felt to be very slow, or when cycle time needs to be reduced, use a value greater than 100%.

The time used for motion from a starting point to a destination point depends on the acceleration override. The acceleration override value ranges from 0 to 150%. Acceleration override is programmed at the destination position.

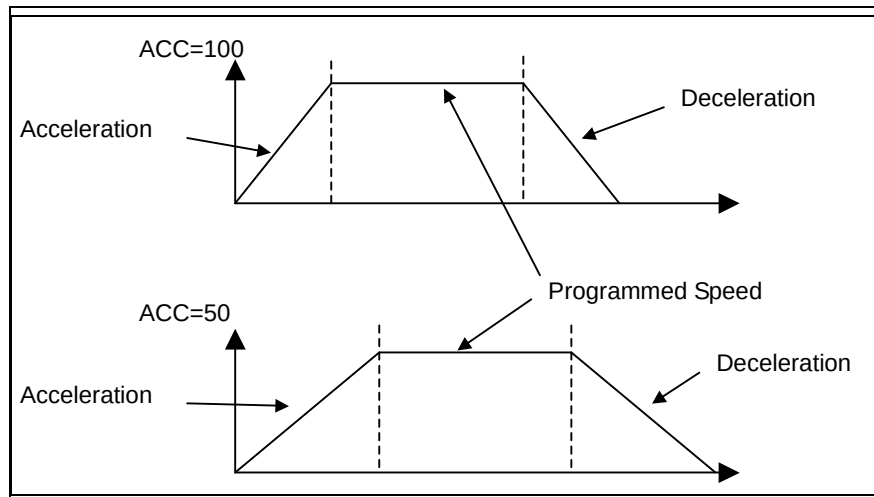


Fig. 4.3.5 (a) Acceleration Override

**CAUTION**

If the acceleration override value is set to the value more than 100%, awkward movement and vibration may occur. Because primary power source has instant large current, the input voltage may drop depending on equipment power capacity, and this may cause a servo alarm. Such as power alarm, position error excess, and servo amplifier low voltage. If this occurs either reduce the acceleration/deceleration override value or delete the accelerate/deceleration override instruction.

Skip instruction

SKIP CONDITION [I/O] = [value] J P[1] 50 FINE Skip,LBL[3]

Motion Modify

- 1 No option
- 2 Wrist joint
- 3 ACC
- 4 Skip,LBL[]

A skip instruction causes a jump to a branch destination label if the skip condition is not satisfied. If the skip condition is satisfied while the robot is moving to a target point, the robot cancels the motion and program execution proceeds to the program statement on the next line. If the skip condition is not satisfied, program execution skips (jumps) to the line of the branch destination label after completion of the robot motion.

The skip condition instruction specifies, in advance, a skip condition (condition for executing a skip instruction) to be used with it. Before a skip instruction can be executed, a skip condition instruction must be executed. A skip condition once specified is valid until the execution of the program is completed, or the next skip condition instruction is executed. (For the branch instructions, see Section 4.7. For the skip condition instruction, see Section 4.9.)

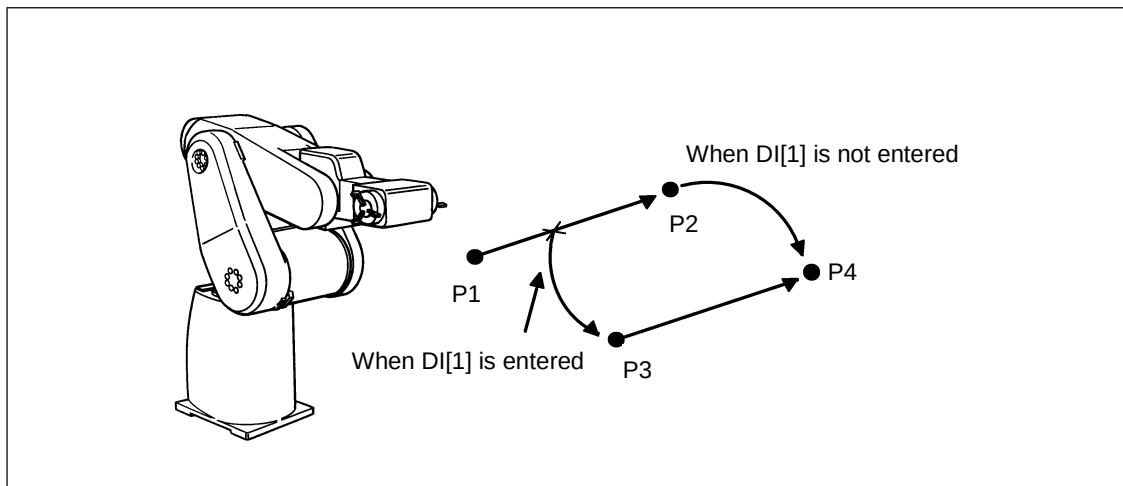


Fig. 4.3.5 (b) Skip Instruction

Example

```

1: SKIP CONDITION DI[1] = ON
2: J P[1] 100% FINE
3: L P[2] 1000mm/sec FINE Skip, LBL[1]
4: L P[3] 50% FINE
5: LBL[1]
6: J P[4] 50% FINE

```

High-speed skip

- Function outline

- (1) The position of the robot when the skip conditions are met can be stored in programmed position registers.
- (2) 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.

- Use method

The high-speed skip function can be used in program teaching. There is no need to set system variables.

Program teaching

a) Teaching skip conditions

The skip conditions for the high-speed skip function are taught in the same way as the ordinary skip function.

b) Teaching a high-speed 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	←		High-speed skip instruction
3 Skip, LBL,	←		Ordinary skip instruction

c) Specify the label, position register, and position storage format.

```

Skip, LBL[10], PR[5]=LPOS
or JPOS

```

[Sample program]

```

:
8: SKIP CONDITION DI[3]=ON
:
10: LP[2]500mm/sec FINE
11: LP[3]100mm/sec
: SKIP, LBL[10], PR[5]=LPOS

```

```

      :
      :
30: LBL[10]

```

Explanation of the execution example

When DI[3] is turned on during execution of the 11th line, the current position is stored in a form of Cartesian coordinates.

When DI[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].

- Limitations and notes

<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.)

OFFSET instruction

Offset,PR[2] (UFRAME [1])

J P[1] 50% FINE Offset

```

                                JOINT 30 %
5 Offset
6 Offset,PR[ ]
7 Incremental
8---next page---

```

The OFFSET instruction alters positional information programmed at the destination position by the offset amount specified by a position register, and moves the robot to the altered position.

The offset condition is specified by the OFFSET CONDITION instruction.

The OFFSET CONDITION instruction specifies the offset amount used by the OFFSET instruction in advance. The OFFSET CONDITION instruction has to be specified before the OFFSET instruction is executed. The specified offset condition is available until the program is finished or the next OFFSET CONDITION instruction is executed:

As for the offset condition, the following elements should be specified:

- The position register specifies the shifting direction and the shift amount.
- When the positional information is expressed in the joint frame, the shift amount of each axis is applied.
- When the positional information is expressed in the Cartesian coordinate system, the user frame by which the offset condition is decided should be specified.(See Section 4.12, "FRAME INSTRUCTION) When it is not specified, the user frame (UF) being selected now is used.(See Section 4.10, "OFFSET CONDITION INSTRUCTION")

⚠ CAUTION

If teaching is made by joint coordinates, changing the user coordinate system does not affect the position variables and position registers. However, note that both position variables and registers are affected by the user coordinate systems when the robot is taught in the Cartesian format.

The setting values of the tool frame number (UT) and the configuration (CONF:) are ignored.

When you teach or edit the positional information of the motion instruction with the OFFSET option, you teach the position minus the offset amount.

When you teach or edit the positional information of the motion instruction with the OFFSET option, the following prompt message is displayed.

- Subtract offset data from current pos?
 - Yes The positional information subtracted the offset data is taught.
 - No The positional information is directly taught.
- Enter PR index of offset data :
 - Enter the number of the position register specified in the OFFSET CONDITION instruction.
- Enter uframe no of offset data :
 - Enter the number of the user frame which is used when the offset amount is subtracted.

When the positional information is manually edited with the numerical keys, you can not teach the positional information minus the offset amount.

Moreover, even if the position teaching by which the amount of the subtracted correction is effective, the current position will be taken in the following cases.

- The specified position register is non-initialization.
- "Ignore Offset command" is set to ENABLED. (See Section 3.15 "SETTING THE GENERAL ITEMS")

When "Ignore Offset command" is set to ENABLED, the current position is directly taught as the positional information (The prompt message is not displayed) and the robot stops at the teaching position even if the OFFSET instruction is executed.

When the offset amount is changed after the program is paused while the OFFSET instruction is in progress, this change is reflected to the motion after the program is resumed. But, when you change the number of a position register in the OFFSET CONDITION instruction, this change is not reflected in the motion.

The robot moves to the offset position at the backward execution. (See Subsection 6.3.2, "Step Test")

This is the same as the following explanation for the direct offset condition instruction.

Direct offset condition instruction

J P[1] 50% FINE Offset,PR[2]

```

JOINT 30 %
5 Offset
6 Offset,PR[ ]
7 Incremental
8---next page---
```

The direct offset condition instruction alters positional information by the offset amount directly specified in the position register without using the offset condition specified in the OFFSET CONDITION instruction. The reference frame is specified by the number of the user frame currently selected.



CAUTION

If teaching is made by joint coordinates, changing the user coordinate system does not affect the position variables and position registers. However, note that both position variables and registers are affected by the user coordinate systems when the robot is taught in the Cartesian format.

When you change or edit the motion instruction with the direct offset condition option, you can teach the positional information minus the offset amount.

When you teach or edit the motion instruction with the direct offset condition option, the following prompt message is displayed:

- Subtract offset data from current pos?
 - Yes Subtract the offset data from the taught position.
 - No The positional information from the directly taught position.

When the positional information is manually edited with the numerical keys, you can not teach the positional information minus the offset amount.

Moreover, even if the position teaching by which the subtracted offset amount is effective, the current position will be taught as it is in the following cases:

- The specified position register is non-initialized.
- The position register number used by direct offset condition instruction is non-initialized.
- "Ignore Offset command" is set to ENABLED.(See Section 3.16 "SETTING THE GENERAL ITEMS")

When "Ignore Offset command" is set to ENABLED, the current position is directly taught as the positional information (The prompt message is not displayed) and the robot stops at the teaching point even if the offset instruction is executed.

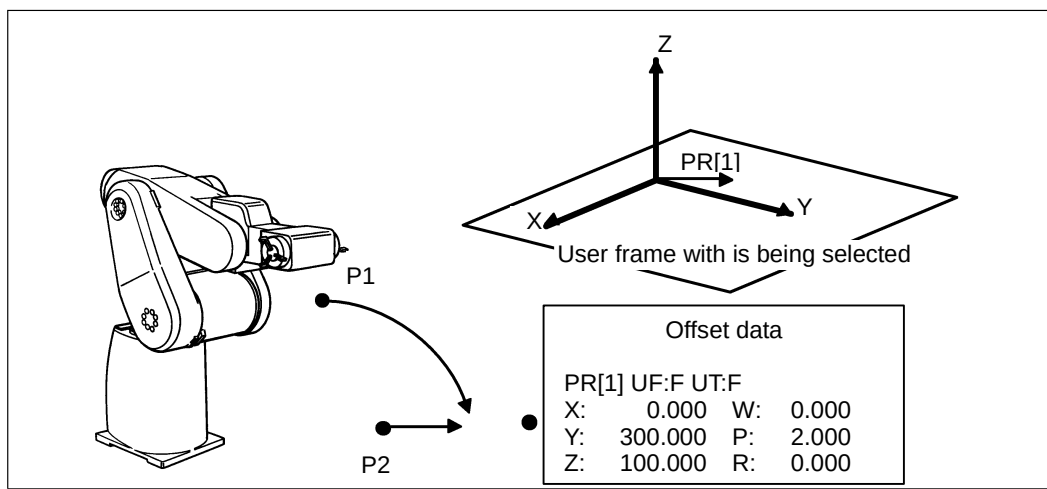


Fig. 4.3.5 (c) Offset Instruction

Example 1
 1: OFFSET CONDITION PR[1]
 2: J P[1] 100% FINE
 3: L P[2] 500mm/sec FINE Offset

Example 2
 1: J P[1] 100% FINE
 2: L P[2] 500mm/sec FINE Offset, PR[1]

Tool offset instruction

TOOL_OFFSET_CONDITION PR[2] (UTOOL[1])
 J P[1] 50% FINE Tool_offset

```

JOINT 30 %
5 Tool_Offset
6 Tool_Offset,PR[
7 Incremental
8---next page---
```

A tool offset instruction moves the robot to the position shifted from the target position, recorded in the position data, by the offset specified in the tool offset conditions. The condition when the offset is applied is specified by a 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. Once the tool offset condition has been specified, it remains effective until the program terminates or the next tool offset condition instruction is executed.

Note the following when specifying tool offset conditions.

- The position register specifies the direction in which the target position shifts, as well as the amount of the shift.
- The tool coordinate system is used for specifying 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 to which the offset is not to be applied can be taught.

When a motion statement which includes a tool offset instruction is taught or a certain position is modified, the system prompts the operator to respond to enter data in response to the following messages.

- Subtract tool offset data?
 - Pressing the YES soft key subtracts the tool offset from the position data and the robot is taught the new position.
 - Pressing the NO soft key stores the current position as the position data.
- Enter PR index of tool offset data?
 - Specify the position-register number specified by the tool offset condition instruction.
- Enter tool no. of tool offset data?
 - Specify the number of the tool coordinate system in which the offset is to be specified.

When the position data is manually modified with the numeric keys, the position is taught without subtracting the offset.

Even when teaching of the position from which the offset is subtracted is enabled, the current position is stored in the following cases.

- When the specified position register has not yet been initialized
- When "Ignore Tool-offset" is set to ENABLED. (See Section 3.16 "SETTING THE GENERAL ITEMS".)

When "Ignore Tool-offset" is set to ENABLED, the current position is taught as position data (no prompt messages are output) and the robot is moved to the taught position, even if a tool offset instruction is executed.

When the robot is temporarily stopped during the execution of a tool offset instruction and the shift distance is modified, the modified distance is used in the resumed movement. When a position register number specified by a tool offset condition instruction is modified, the modified number is not used.

In backward execution (See Subsection 6.3.2, "Step Test"), 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.

Direct tool offset instruction

J P[1] 50% FINE Tool_Offset, PR[2]

```

JOINT 30 %
5 Tool_Offset
6 Tool_Offset,PR[
7 Incremental
8---next page---
```

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 a motion statement which includes a direct tool offset instruction is taught or a certain position is modified, the position to which the offset is not to be applied can be taught.

When a motion statement which includes a direct tool offset instruction is taught or a certain position is modified, the system prompts the operator to enter data in response to the following messages.

- Subtract tool offset data?

- Pressing the YES soft key subtracts the tool offset from the position data and the robot is taught the new position.
- Pressing the NO soft key stores the current position as position data.

When the position data is manually modified with the numeric keys, the position is taught without subtracting the offset.

When teaching of the position from which the offset is subtracted is enabled, the current position is stored in the following cases.

- When the specified position register has not yet been initialized
- When the direct tool offset instruction has not specified the number of a position register
- When "Ignore Tool-offset" is set to ENABLED. (See Section 3.16 "SETTING THE GENERAL ITEMS".)

When "Ignore Tool-offset" is set to ENABLED, the current position is taught as position data (no prompt messages are output) and the robot is moved to the taught position even if a tool offset instruction is executed.

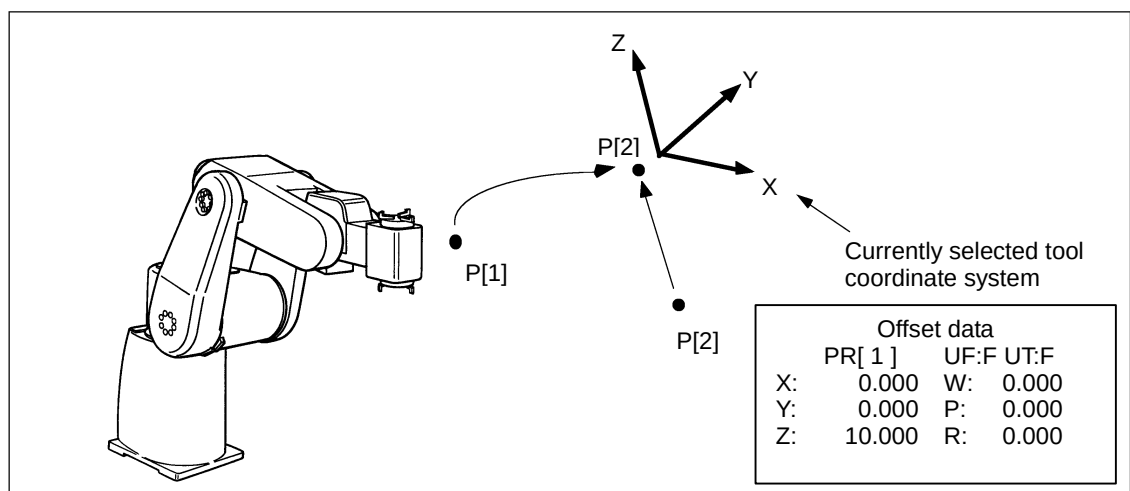


Fig. 4.3.5 (d) Tool Offset Instruction

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]

Incremental instruction

J P[1] 50% FINE INC

JOINT 30 %
5 Tool_Offset
6 Tool_Offset,PR[
7 Incremental
8---next page---

The incremental instruction uses the positional data in the motion instruction as the incremental amount from the current position, and causes the robot to move to the destination position that the incremental amount is added to the current position. This means that the incremental motion amount from the current position is recorded in the positional data in the motion instruction.

The incremental condition is specified by the following elements:

- When the positional data is joint frame value, the incremental amount of each axis is applied.

- When the positional variable (P[]) is used as the positional data, the reference user frame is specified by the number of the user frame which is specified in the positional data. However, the frame is verified.(Cartesian coordinate system reference)
- When the position register is used as the position data, the reference frame is the current user frame.
- When the INC instruction is used with the OFFSET instruction, the type of the positional data in the motion instruction should be corresponding to the type of the positional register for the offset. In this case, the offset amount is used as the offset amount of the specified incremental amount.

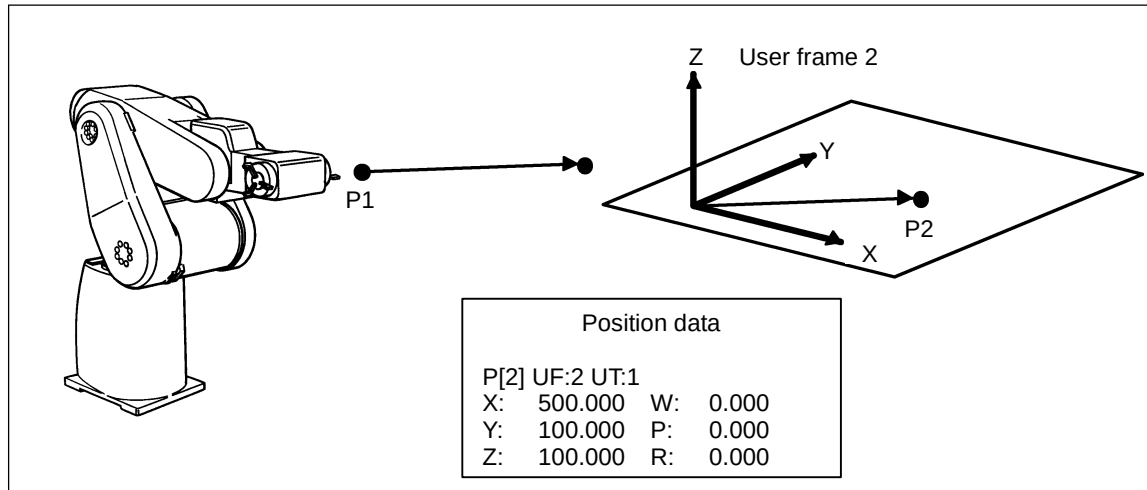


Fig. 4.3.5 (e) Incremental instruction

Example 1 1: J P[1] 100% FINE
 2: L P[2] 500mm/sec FINE INC

Note the following when teaching the incremental instruction (See Subsection 5.3.4, "Teaching an Supplementary Motion Instruction"):

- Adding the INC option causes the positional data to be non-initialized.
- When the motion instruction with the INC option is taught, the positional data is set to be non-teaching.
- Editing the position in the motion instruction with the INC option removes the INC option automatically.

When the motion instruction with the INC option is paused and the position data is changed, that changes is not immediately reflected. To move the robot to the changed position, resume the program from the just previous motion instruction.

Simultaneous EV instruction

J P[1] 50% FINE EV 50%

The additional axis speed instruction (synchronous) moves the robot in sync with the additional axis. When this instruction is used, the robot and additional axis operations are synchronized as follows:

- If the robot operation time is longer than the additional axis operation time, the additional axis operation is synchronized with the robot operation.
- If the additional axis operation time is longer than the robot operation time, the robot operation is synchronized with the additional axis operation.

The extended axis speed is specified as a ratio (1% to 100%) to the maximum travel speed of the extended axis.

Independent EV instruction (Ind.EV i%)

J P[1] 50% FINE Ind.EV 50%

Motion Modify	
1	Independent EV
2	Simultaneous EV
3	
4	
PROGRAM1	

The additional axis speed instruction (asynchronous) moves the robot asynchronously with the additional axis.

When this instruction is used, the robot and the additional axis start moving at the same time, but stop at different times because they are not synchronized.

The extended axis speed is specified as a ratio (1% to 100%) to the maximum travel speed of the extended axis.

If a motion statement is not accompanied with either extended axis speed instruction, the extended axis moves in synchronization with the speed of the robot.

Path instruction

J P[1] 50% Cnt10 PTH

Motion Modify	
1	Independent EV
2	Simultaneous EV
3	PTH
4	
PROGRAM1	

This function is designed to improve the performance of continuous motion (the termination type is Cnt1 to Cnt100) when the robot moves through a short distance.

In a motion where the robot moves through a short distance, the robot speed cannot be increased to the speed specified by a motion statement. For this reason, in an operation statement for which the positioning format is "FINE," operation planning for such an operation must be based on the "attainable speed," the speed that the robot can actually attain, rather than the specified speed. (Motion planning entails calculating the path along which the robot will travel, before actual operation.)

By using this instruction, operation planning is performed using the "attainable speed" in a CNT operation. The use of this function enables the following effects in normal operation:

- Improvement in cycle time
- Improvement in path accuracy

This function is more effective as the movement distance is shorter and the Cnt value is smaller (the value n in Cntn is smaller).

When using this function, note the following:

In the following cases, use of the PTH instruction may actually incur a longer cycle time:

Before using this function, therefore, confirm its effect.

- A large Cnt value is specified in a motion statement.
- A motion statement causes the robot to move through a long distance.
- Successive Cnt motion statements appear.

**CAUTION**

Some motion instructions that use the PTH switch might cause jerky motion or vibration. If the motion is attached to PTH has a vibration, delete the PTH motion option.

BREAK instruction**L P[1] 2000 mm/sec CNT100 BREAK**

By using BREAK instruction, user can have the robot waiting at taught position even if the termtype before WAIT instruction is CNT.

When BREAK instruction is added to the motion instruction with CNT before WAIT instruction. The next motion does not start until the condition of WAIT is satisfied. Therefore the robot moves to the taught position until then and the path of the corner is changed depending on WAIT time. If WAIT time is long enough, the robot waits on the taught position until the next motion starts.

Ex: With BREAK instruction
 1:L P[1] 2000mm/sec FINE
 2:L P[2] 2000mm/sec CNT100 BREAK
 3: WAIT DI[1]=ON or WAIT x(sec)
 4:LP P[3] 2000mm/sec FINE

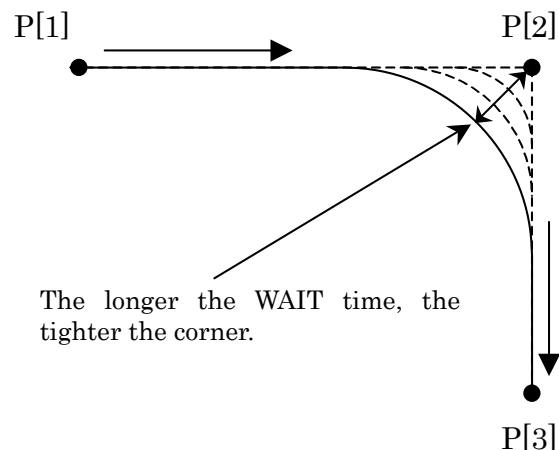


Fig. 4.3.5 (f) Break instruction

[Reference]

Normally, when there is WAIT instruction next to the motion instruction with CNT, the robot waits on the corner until the condition of WAIT is satisfied. The path of the corner is constant regardless of WAIT time.

Ex: Without BREAK instruction
 1:L P[1] 2000mm/sec FINE
 2:L P[2] 2000mm/sec CNT100
 3: WAIT DI[1]=ON or WAIT x (sec)
 4:LP3] 2000mm/sec FINE

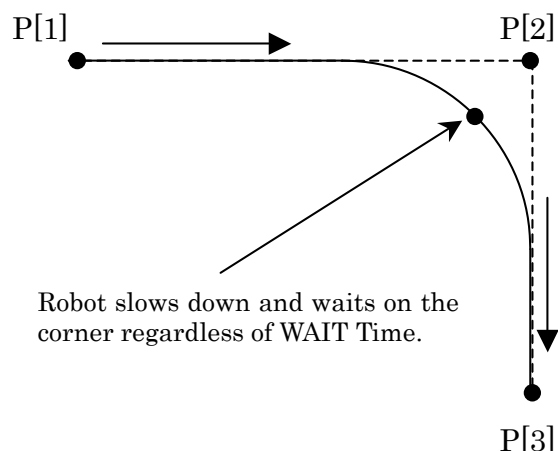


Fig. 4.3.5 (g) Break instruction (In case that break instruction is not taught)

4.4 PALLETIZING INSTRUCTIONS

Palletizing is a function for orderly stacking of workpieces by only teaching several representative points. (See Section 10.1.)

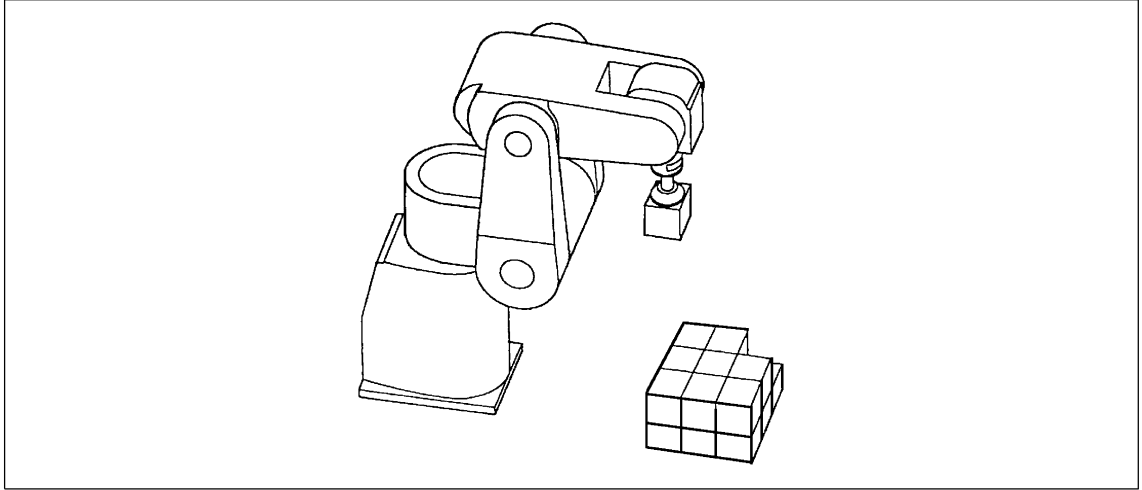


Fig. 4.4 Palletizing

The following palletizing instructions are available:

- Palletizing instruction PALLETIZING-B, BX, E, EX
- Palletizing motion instruction J PAL_i [BTM] 100 % FINE
- Palletizing end instruction PALLETIZING-END-_i

4.4.1 Palletizing Instruction

Based on the value held in the palletizing register, the palletizing instruction calculates the position of the current stack point from a stacking pattern and the position of the current path from a path pattern. It then writes the found values into the position data of a palletizing motion instruction.

PALLETIZING-[pattern]_i

B, BX, E, EX

Palletizing number (1 to 16)

Fig. 4.4.1 Palletizing Instruction

Palletizing is divided into four palletizing patterns according to the stacking and path patterns (See Section 10.1).

Palletizing patterns

For pallet instructions, palletizing is divided into the following palletizing patterns.

Table 4.4.1 Palletizing Patterns

Pattern	Allowable palletizing
B	Simple stacking pattern and single path pattern
BX	Simple stacking pattern and multiple path patterns
E	Complex stacking pattern and single path pattern
EX	Complex stacking pattern and multiple path patterns

4.4.2 Palletizing Motion Instruction

The palletizing motion instruction is a motion instruction that uses three path points - an approach point, stack point, and retraction point - as position data. This instruction is dedicated to palletizing. Each palletizing instruction rewrites such position data.

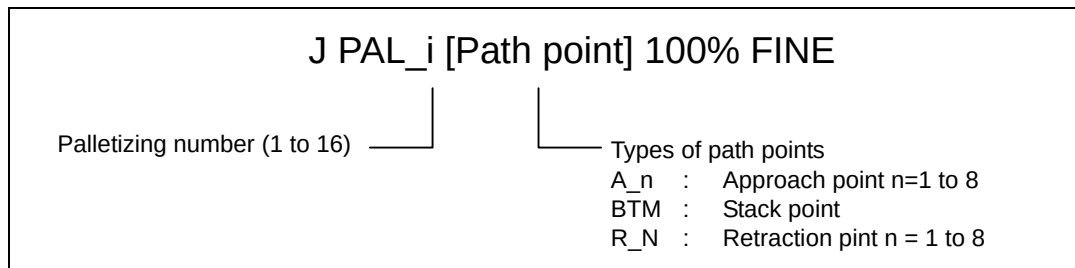


Fig. 4.4.2 Palletizing Motion Instruction

4.4.3 Palletizing End Instruction

The palletizing end instruction increments or decrements the value of the palletizing register.

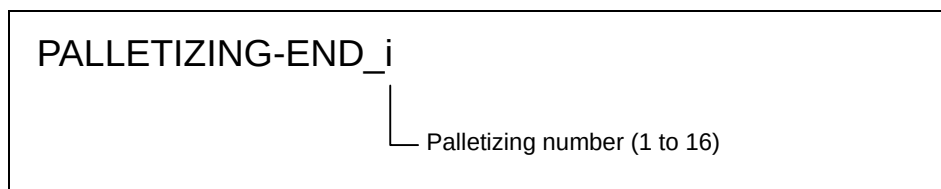


Fig. 4.4.3 Palletizing End Instruction

Example

```

1:   PALLETIZING-B_3
2: L PAL_3[ A_1 ] 100mm/sec CNT10
3: L PAL_3[ BTM ] 50mm/sec FINE
4:   HAND1 OPEN
5: L PAL_3[ R_1 ] 100mm/sec CNT10
6:   PALLETIZING-END_3
  
```

4.5 REGISTER INSTRUCTIONS

The register instructions perform arithmetic operations on registers. The following register instructions are available:

Instruction		JOINT 30 %
1 Registers	5 JMP/LBL	
2 I/O	6 CALL	
3 IF/SELECT	7 Palletizing	
4 WAIT	8---next page---	
PROGRAM1		

- Register instructions
- Position register instructions
- Position register axis instructions
- Palletizing register instructions
- String register, string instructions

In register operations, polynomial operations such as those shown below are possible:

Example 1: $R[2]=R[3]-R[4]+R[5]-R[6]$
 2: $R[10]=R[2]*[100/R[6]]$

The following restrictions are imposed:

- Up to five operators can be written on a single line.

Example 1: $R[2]=R[3]+R[4]+R[5]+R[6]+R[7]+R[8]$

Up to five operators

- The "+" and "-" operators can be mixed on a single line. So can the "*" and "/" operators. "+" and "-" cannot, however, be mixed with "*" and "/".

4.5.1 Register Instructions

A register instruction performs an arithmetic operation on registers. A register is a variable for holding an integer or a decimal fraction. (For registers, See Section 7.3.) Two hundred registers are provided.

$R[i] = (\text{value})$

The instruction, $R[i] = (\text{value})$, loads a value into a specified register.

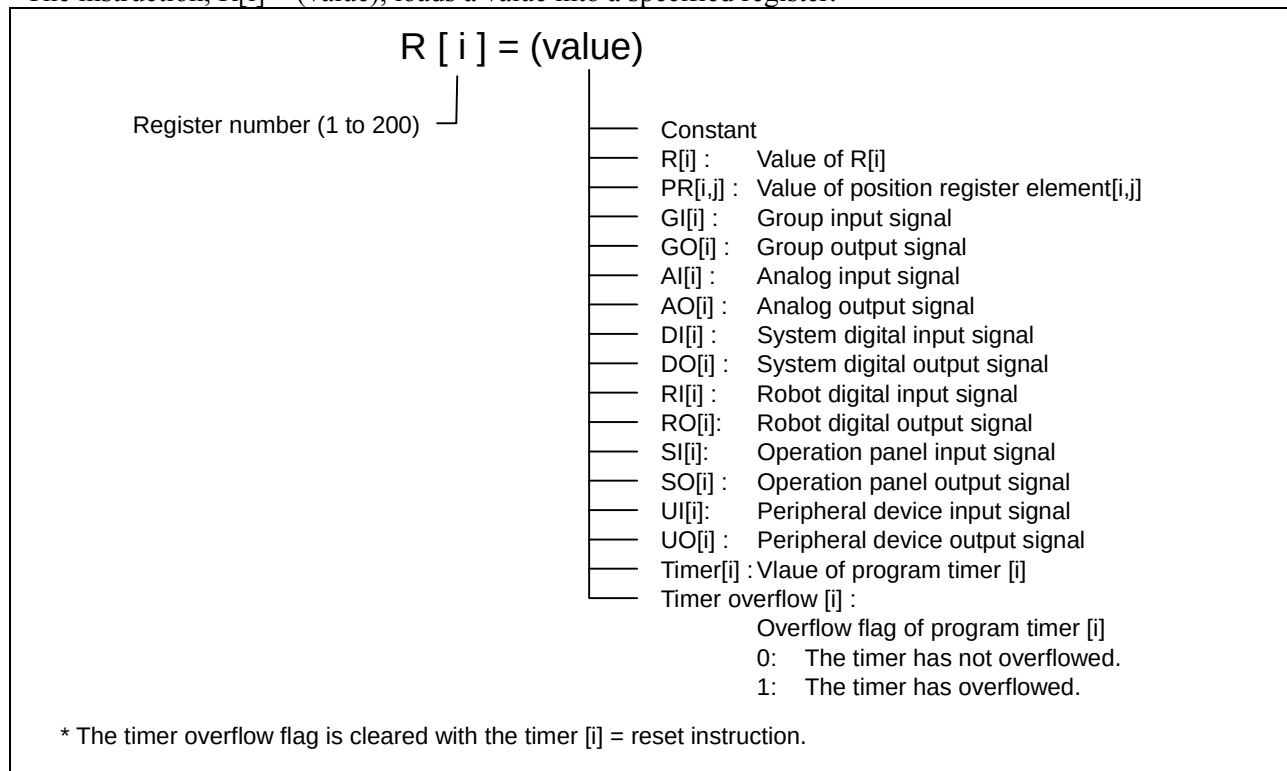


Fig. 4.5.1 (a) Instruction $R[i] = (\text{value})$

Example 1: $R[1] = RI[3]$
 2: $R[R[4]] = AI[R[1]]$

$R[i] = (\text{value}) + (\text{value})$

The instruction, $R[i] = (\text{value}) + (\text{value})$, loads the sum of two values into a specified register.

$R[i] = (\text{value}) - (\text{value})$

The instruction, $R[i] = (\text{value}) - (\text{value})$, loads the difference between two values into a specified register.

$R[i] = (\text{value}) * (\text{value})$

The instruction, $R[i] = (\text{value}) * (\text{value})$, loads the product of two values into a specified register.

$R[i] = (\text{value}) / (\text{value})$

The instruction, $R[i] = (\text{value}) / (\text{value})$, loads the quotient of two values into a specified register.

R[i] = (value) MOD (value)

The instruction, R[i] = (value) MOD (value), loads the remainder (value after decimal point) of the quotient of two values into a specified register.

R[i] = (value) DIV (value)

The instruction, R[i] = (value) DIV (value), loads the integer of the quotient of two values into a specified register.

$$R[i] = (x - (x \text{ MOD } y)) / y$$

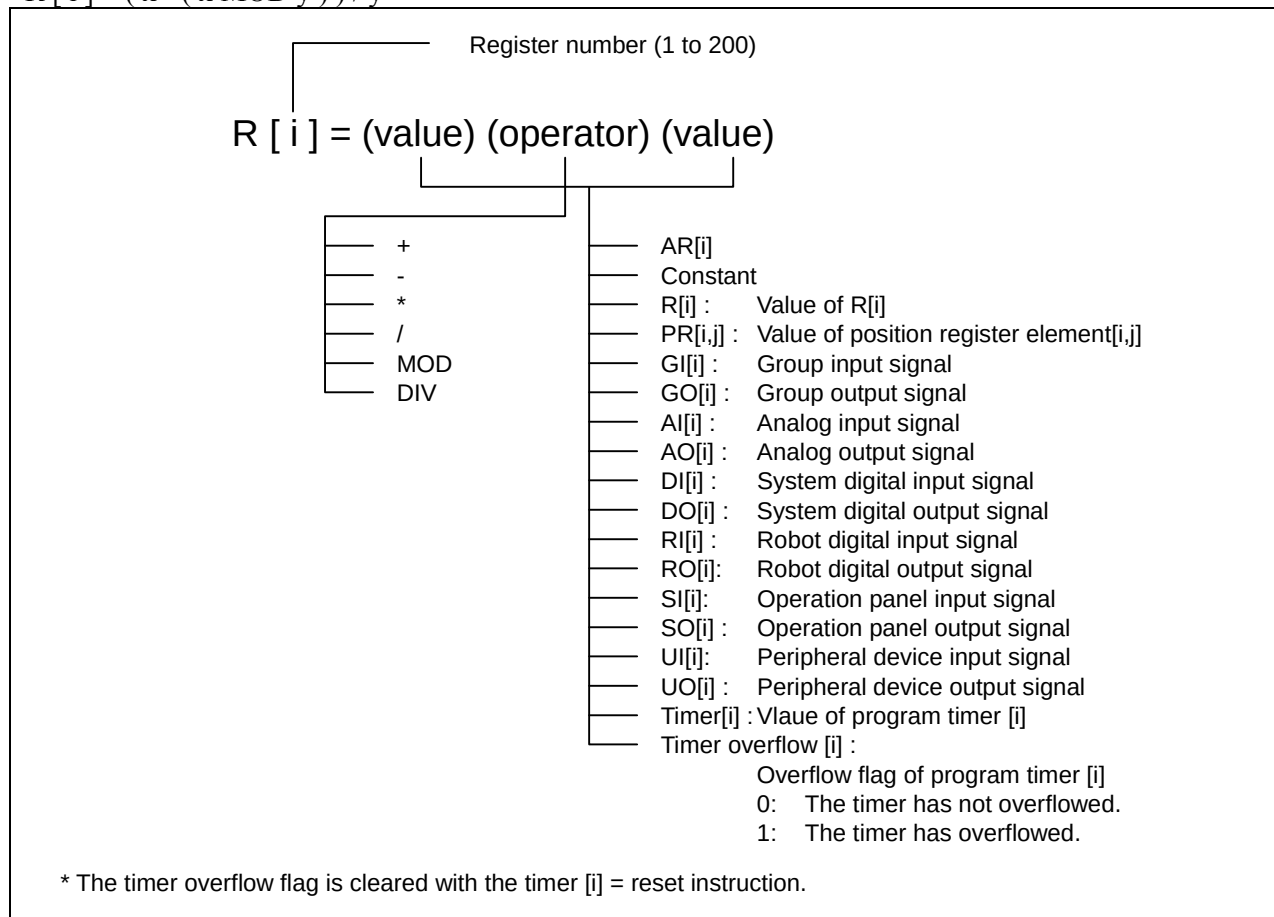


Fig. 4.5.1 (b) Arithmetic Register Instruction

Example 3: R[3:flag] = DI[4]+PR[1, 2]
 4: R[R[4]] = R[1]+1

4.5.2 Position Register Instructions

A position register instruction performs an arithmetic operation on position registers. A position register instruction can load position data, the sum of two values, or the difference of two values, into a specified position register. A position register instruction uses the same format as a register instruction.

A position register is a variable for holding position data (x,y,z,w,p,r). (For position registers, See Section 7.4.) One hundred position registers are provided.

NOTE

Before using position register instructions, lock position registers by specifying LOCK PREG. When position register instructions are used with the position registers unlocked, operation may become tight. For the LOCK PREG instruction, see Section 9.6, "POSITION REGISTER LOOK-AHEAD EXECUTION FUNCTION."

PR[i] = (value)

The instruction, PR[i] = (value), loads position data into a specified position register.

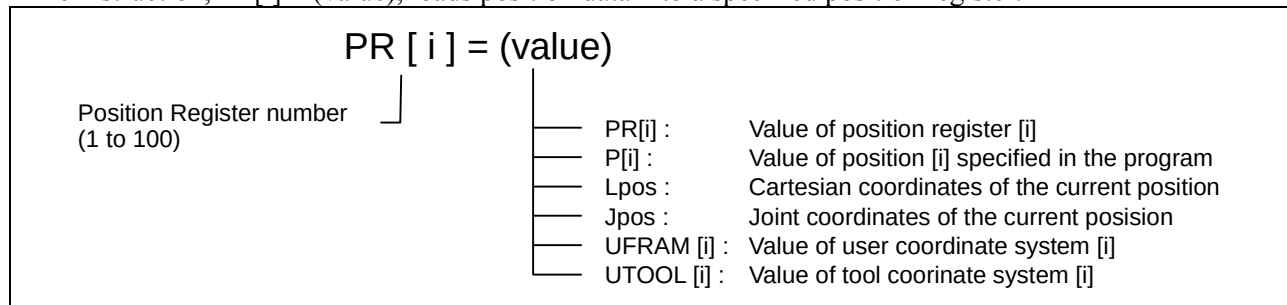


Fig. 4.5.2 (a) Instruction PR[i] = (value)

Example 1: PR[1] = Lpos
 2: PR[R[4]] = UFRAM[R[1]]
 3: PR[9] = UTOOL[1]

PR[i] = (value) + (value)

The instruction, PR[i] = (value) + (value), loads the sum of two values into a specified register.

The instruction, PR[i] = (value) - (value), loads the difference of two values into a specified register.

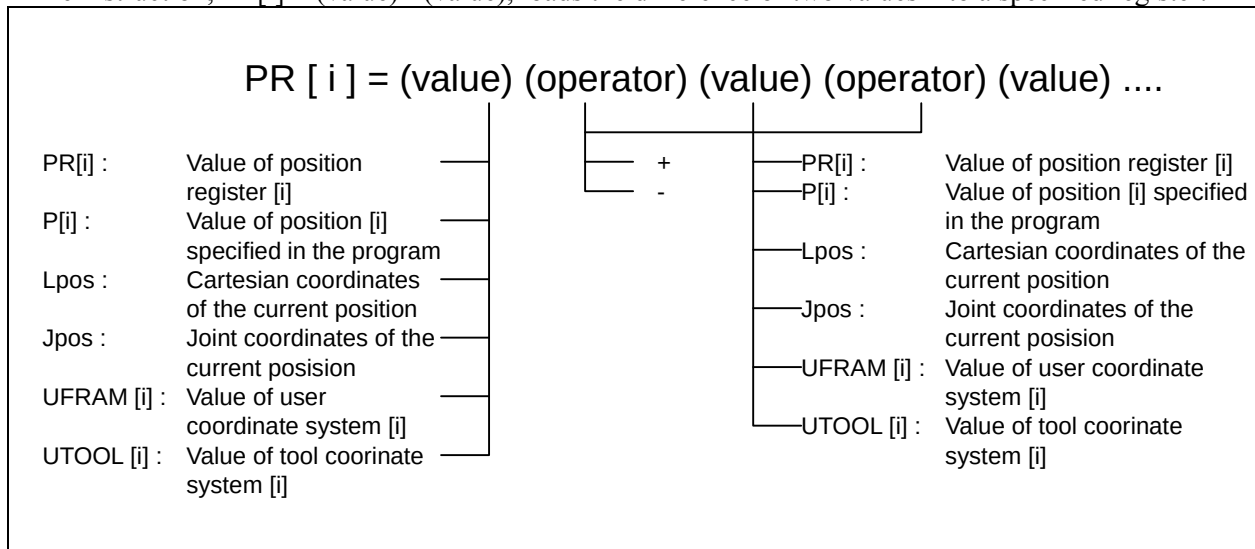


Fig. 4.5.2 (b) PR[i] Arithmetic Instruction

Example 4: PR[3] = PR[3]+Lpos
 5: PR[4] = PR[R[1]]

4.5.3 Position Register Axis Instructions

A position register axis instruction performs an arithmetic operation on position register elements. i of PR[i,j] represents a position register number, and j of PR[i,j] represents a position register element number. The position register axis instructions can load the value of one position data element, or the sum, difference, product, or quotient of two values into a specified position register element. A position register axis instruction uses the same format as a register instruction.

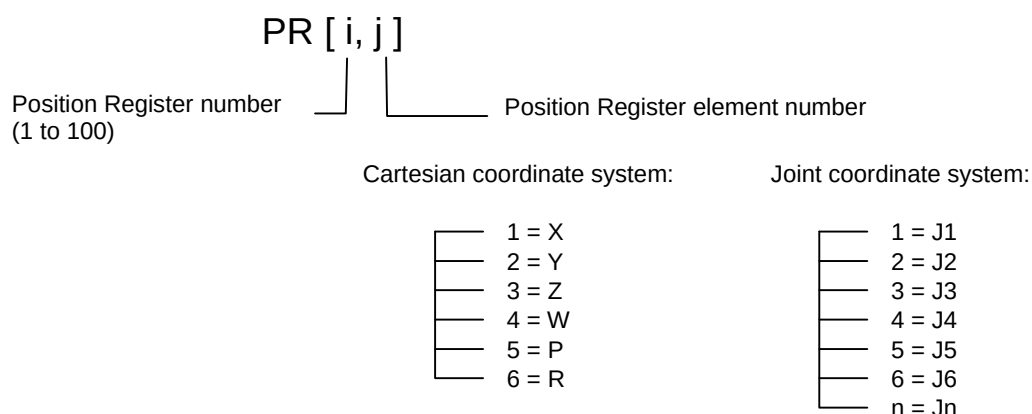


Fig. 4.5.3 (a) Format of PR[i,j]

PR[i,j] = (value)

The instruction, PR[i,j] = (value), loads the value of a position data element into a position register element.

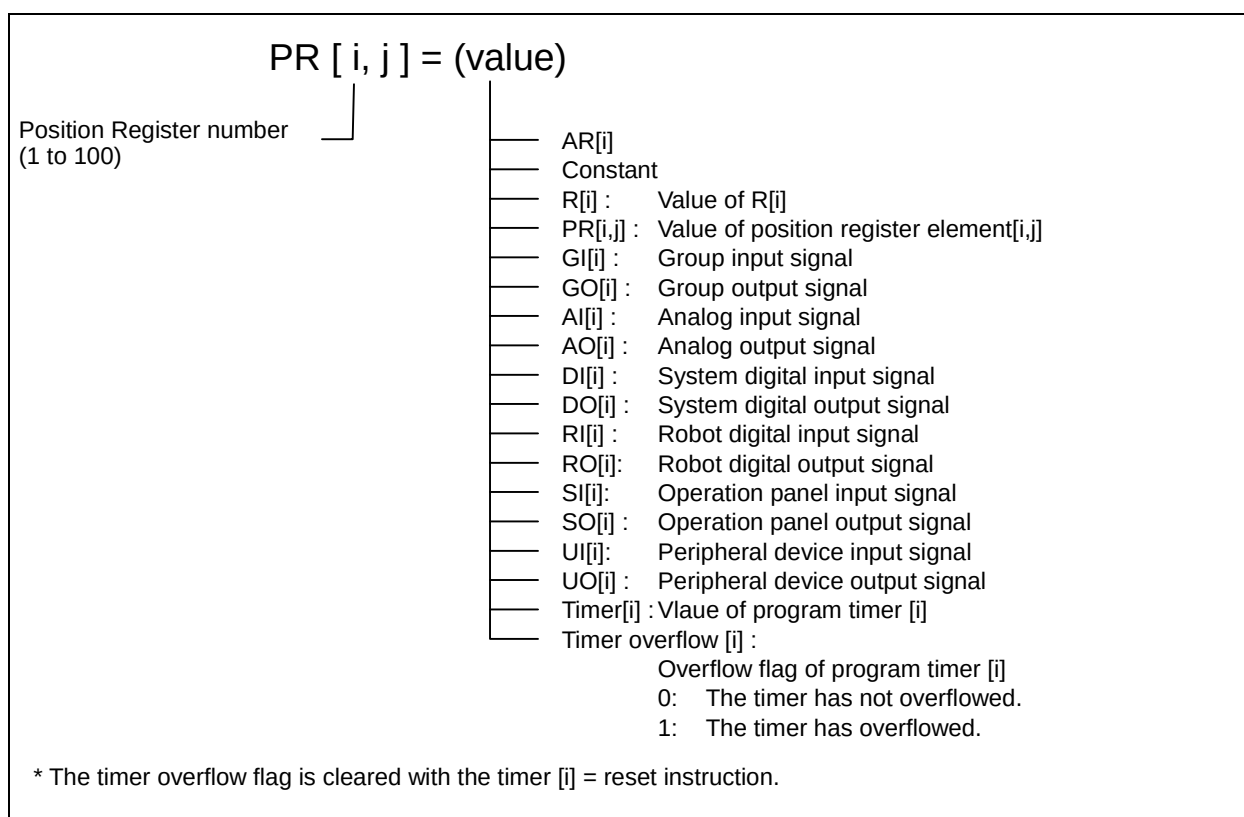


Fig. 4.5.3 (b) Instruction PR[i,j] = (value)

Example 1: PR[1, 2] = R[3]
 2: PR[4, 3] = 324.5

PR[i] = (value) + (value)

The instruction, PR[i,j] = (value) + (value), loads the sum of two values into a specified position register element.

PR[i] = (value) - (value)

The instruction, PR[i,j] = (value) - (value), loads the difference of two values into a specified position register element.

PR[i] = (value) * (value)

The instruction, $PR[i,j] = (value) * (value)$, loads the product of two values into a specified position register element.

PR[i] = (value) / (value)

The instruction, $PR[i,j] = (value) / (value)$, loads the quotient of two values into a specified position register element.

R[i] = (value) MOD (value)

The instruction, $R[i] = (value) \text{ MOD } (value)$, loads the remainder (value after decimal point) of the quotient of two values into a specified register.

R[i] = (value) DIV (value)

The instruction, $R[i] = (value) \text{ DIV } (value)$, loads the integer of the quotient of two values into a specified register.

$$R[i] = (x - (x \text{ MOD } y)) / y$$

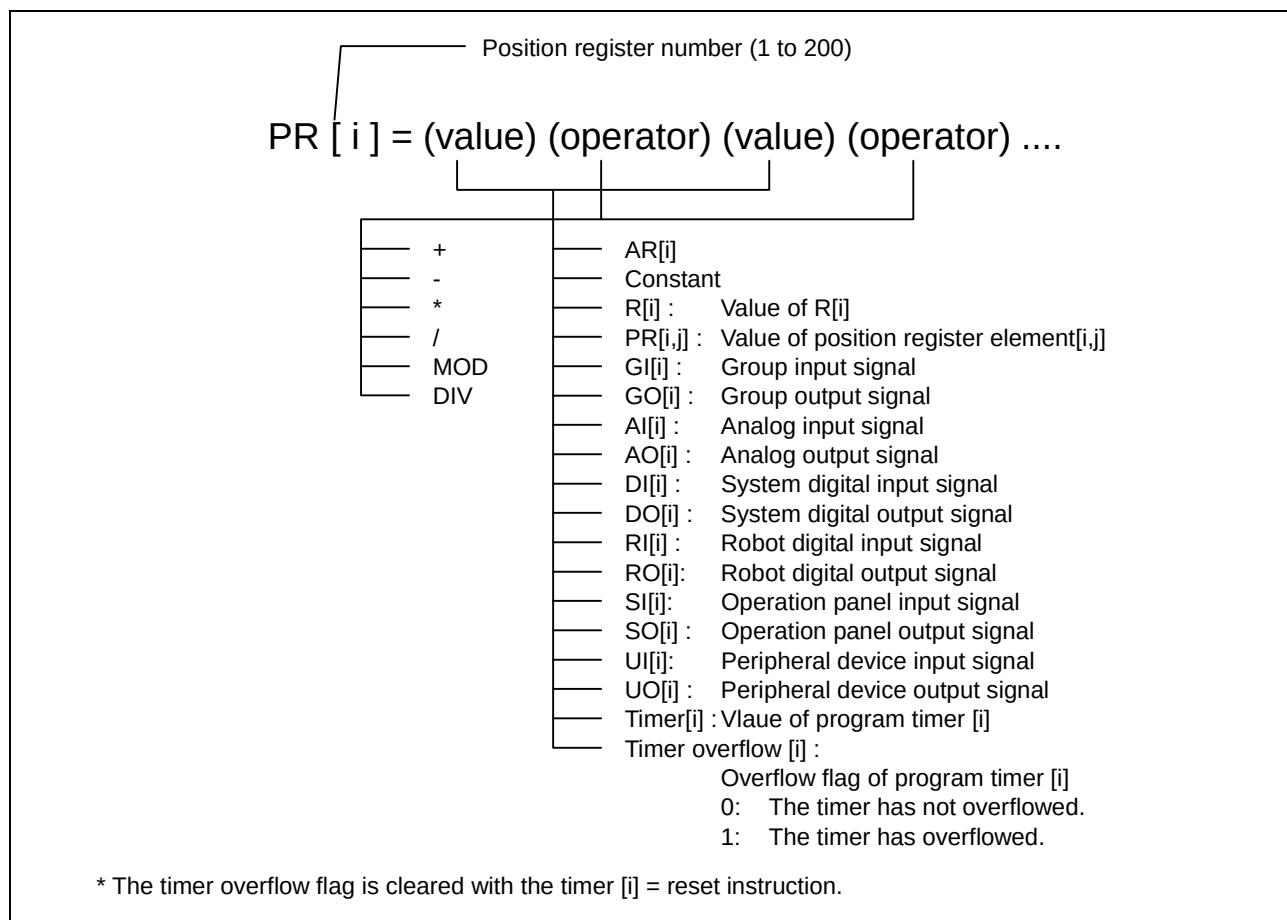


Fig. 4.5.3 (c) PR[i,j] Arithmetic Instruction

Example 1: $PR[3, 5] = R[3] + DI[4]$
 2: $PR[4, 3] = PR[1, 3] - 3.528$

4.5.4 Arithmetic Palletizing Register Instructions

An arithmetic palletizing register instruction performs an arithmetic operation on palletizing registers. The arithmetic palletizing register instructions are load data, the sum of two values, or the difference between two values. An arithmetic palletizing register instruction uses the same format as a register instruction.

A palletizing register holds palletizing register elements (j,k,l). A program can use up to 32 palletizing registers. (See Section 7.5.)

Palletizing register element

For palletizing register elements, elements to be loaded into a palletizing register, or elements to be operated on are specified. Three methods of element specification are available; - Direct specification: A numerical value is directly specified. - Indirect specification: The value of register [i] is specified. - Omitted: The asterisk (*) specifies that no modification is to be made.

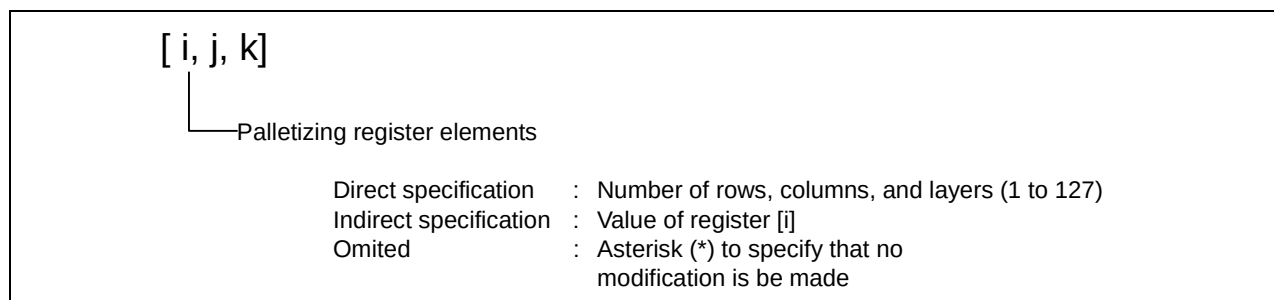


Fig. 4.5.4 (a) Format of Palletizing Register Elements

PL[i] = (value)

The instruction, PL[i] = (value), loads (assigns) palletizing register elements into a specified palletizing register.

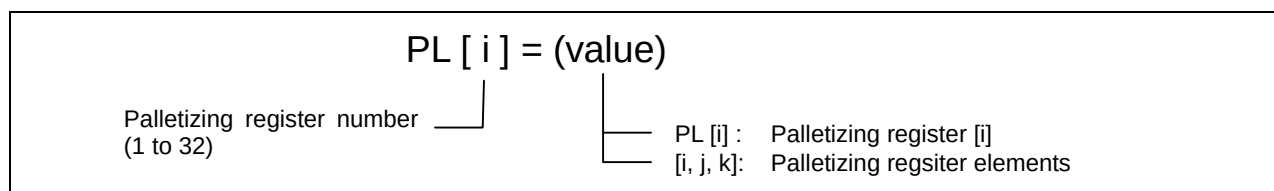


Fig. 4.5.4 (b) Instruction PL [i] = (value)

Example 1: PL[1] = PL[3]
 2: PL[2] = [1, 2, 1]
 3: PL[R[3]] = [*, R[1], 1]

PL[i] = (value) (operator) (value)

The instruction, PL[i] = (value) (operator) (value), performs an arithmetic operation, then loads the result of the operation into a specified palletizing register.

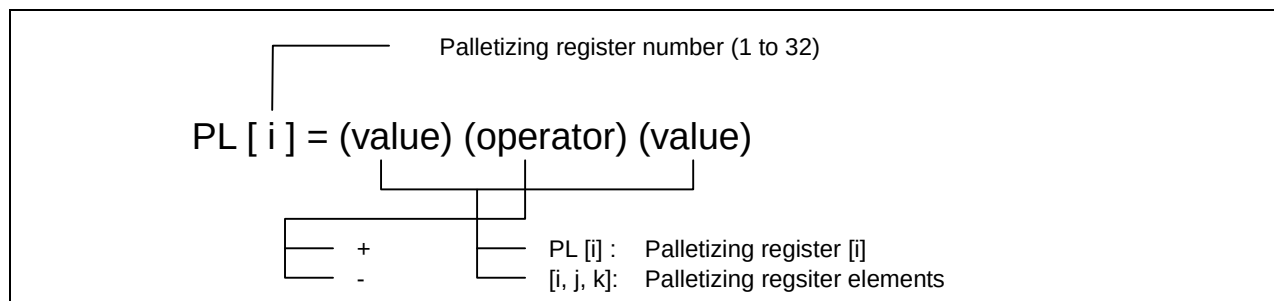


Fig. 4.5.4 (c) Instruction PL [i] = (value) (operator) (value)

Example 1: PL[1] = PL[3] + [1, 2, 1]
 2: PL[2] = [1, 2, 1] + [1, R[1], *]

4.5.5 String Register, String Instructions

NOTE

This item 4.5.5 is available only on 7DA5 series or later.

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.

SR[i] = (value)

The instruction, SR[i]=(value), loads a value into a specified string register.

Conversion from either integer or floating point numeric data is allowed. Floating point data will be rounded to 6 decimal places.

Conversion from strings to numeric data is allowed. Conversion from string to numeric data will stop when the first alpha character is found.

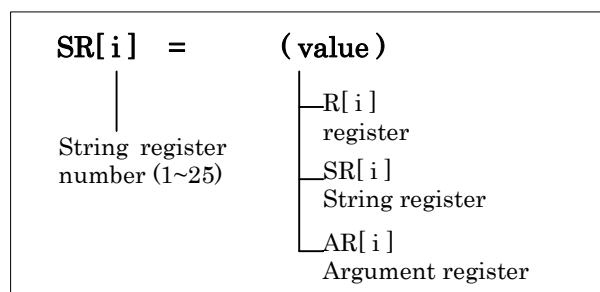


Fig. 4.5.5 (a) Instruction SR[i]=(value)

Example: SR[i] = R[j]

R[j] Value	SR[i] Result
R[j]=1234	SR[i]='1234'
R[j]=12.34	SR[i]='12.34'
R[j]=5.123456789	SR[i]='5.123457'

Example: R[i] = SR[j]

SR[j] Value	R[i] Result
SR[j]='1234'	R[i]=1234
SR[j]='12.34'	R[i]=12.34
SR[j]='765abc'	R[i]=765
SR[j]='abc'	R[i]=0

SR[i] = (value) (operator) (value)

The instruction, SR[i] = (value) (operator) (value), concatenate two values and loads the result into a specified string register.

The data is converted to the data type of the left instruction for each operation.

When the data type of the left instruction is string data, concatenation will be performed.

When the data type of the left instruction is numeric data, arithmetic operation will be performed. Then, if the data type of the right instruction is string data, conversion will stop when the first alpha character is found.

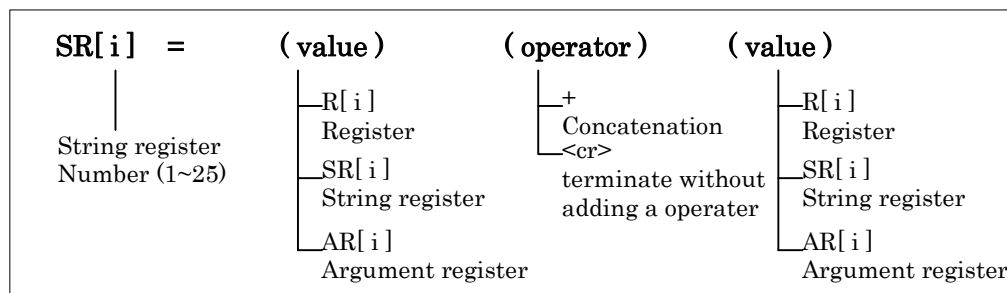


Fig. 4.5.5 (b) Instruction SR[i] = (value) (operator) (value)

Example: SR[i] = R[j] + SR[k]

R[j], SR[k] Value	SR[i] Result
R[j]=123.456+SR[k]='345.678'	SR[i]='456.134'
R[j]=456+SR[k]='1abc2'	SR[i]='457'

Example: SR[i] = SR[j] + R[k]

SR[j], R[k] Value	SR[i] Result
SR[j]='123.'+R[k]=456	SR[i]='123.456'
SR[j]='abc'+R[k]=81573	SR[i]='abc81573'

As the result of the concatenation of string, if the length of string stored in the string register is over 254 character, the alarm "INTP-323 Value overflow" occurs.

R[i] = STRLEN (value)

The instruction, R[i] = STRLEN (value), returns the length of string stored in the specified register.

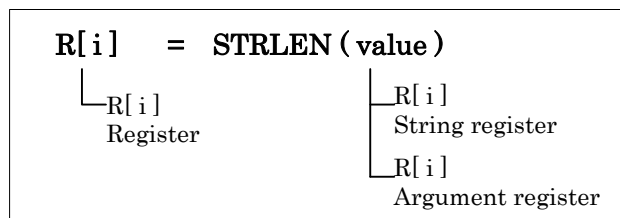


Fig. 4.5.5 (c) Instruction R[i] = STRLEN (value)

Example: R[i] = STRLEN SR[j]

SR[j] Value	R[i] Result
SR[j]='abcdefghij'	R[i]=10
SR[j]='abc1,2,3,4,5,6,de'	R[i]=17
SR[j]=''	R[i]=0

R[i] = FINDSTR (value) (value)

The first (value) means "the target string", and the second (value) means "the search string".

The instruction, R[i] = FINDSTR (value) (value), searches the target 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 "0" is returned.

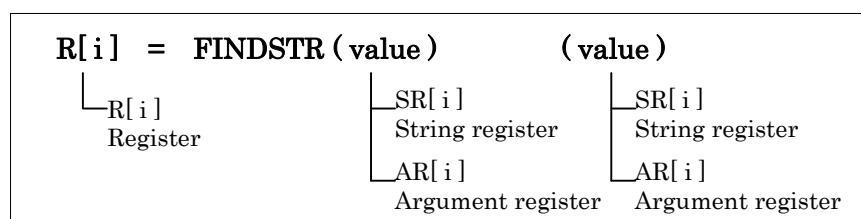


Fig. 4.5.5 (d) Instruction R[i] = FINDSTR (value) (value)

Example: $R[i] = \text{FINDSTR } SR[j], SR[k]$

$SR[k]$ Value, $SR[j] = \text{'find this character'}$	$R[i]$ Result
$SR[k] = \text{'find'}$	$R[i] = 1$
$SR[k] = \text{'character'}$	$R[i] = 11$
$SR[k] = \text{'nothing'}$	$R[i] = 0$
$SR[k] = \text{''}$	$R[i] = 0$

$SR[i] = \text{SUBSTR (value) (value) (value)}$

The first (value) means “the target string”, the second (value) means “the start point”, and the third (value) means “the length”.

The instruction, $SR[i] = \text{SUBSTR (value) (value) (value)}$, returns a substring from the target string based on the start point and length specified.

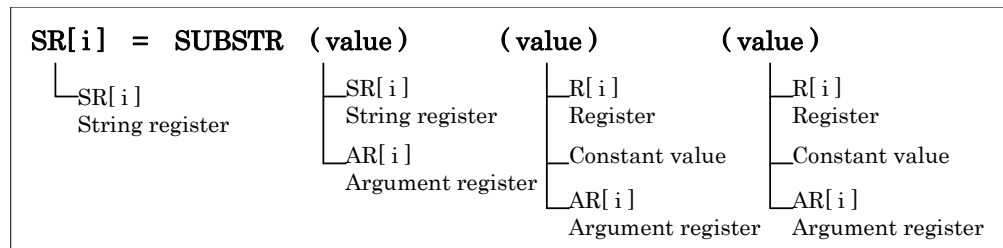


Fig. 4.5.5 (e) Instruction $SR[i] = \text{SUBSTR (value) (value) (value)}$

Example: $SR[i] = \text{SUBSTR } SR[j], R[k], R[l]$

$R[k], R[l]$ Value, $SR[j] = \text{'This string will be broken apart.'}$	$SR[i]$ Result
$R[k] = 1, R[l] = 4$	$SR[i] = \text{'This'}$
$R[k] = 28, R[l] = 5$	$SR[i] = \text{'apart'}$
$R[k] = 8, R[l] = 0$	$SR[i] = \text{''}$

NOTE

Starting point should be larger than “0”, and length should be larger than or equal to “0”. Starting point plus length should be smaller than the target string length.

4.6 I/O INSTRUCTIONS

The I/O (input/output signal) instructions are used to change the state of a signal output to peripheral devices and read the state of an input signal.

Instruction		JOINT 30 %
1 Registers	5 JMP/LBL	
2 I/O	6 CALL	
3 IF/SELECT	7 Palletizing	
4 WAIT	8 ---next page---	
PROGRAM		

- (System) digital I/O instruction
- Robot (digital) I/O instruction
- Analog I/O instruction
- Group I/O instruction

NOTE

As for the I/O signal, the logical number need to be allocated to the physical number before using it.(For configuring I/O, See Section 3.1.)

4.6.1 Digital I/O Instructions

The digital input signal (DI) and digital output signal (DO) are input/output signals that can be controlled by the user.

R[i] = DI[i]

The instruction, R[i] = DI[i] loads, the state of a digital input signal (on = 1/off = 0) into a specified register.

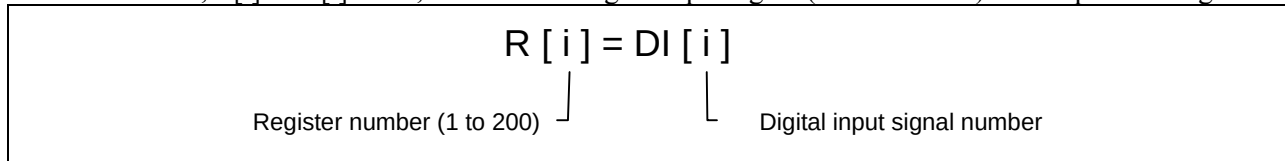


Fig. 4.6.1 (a) Instruction R[i] = DI[i]

Example 1: R[1] = DI[1]
2: R[R[3]] = DI[R[4]]

DO[i] = ON/OFF

The instruction, DO[i] = ON/OFF, turns on or off a specified digital output signal.

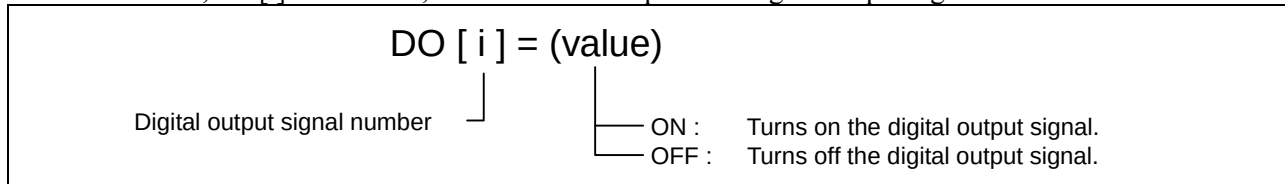


Fig. 4.6.1 (b) Instruction DO[i] = ON/OFF

Example 3: DO[1] = ON
4: DO[R[3]] = OFF

DO[i] = PULSE,[WIDTH]

The DO[i] = PULSE, [TIME] instruction inverts the current status of a specified digital output for a specified duration. When no duration is specified, pulse output is executed for the duration specified with \$DEFPULSE (0.1-second units).

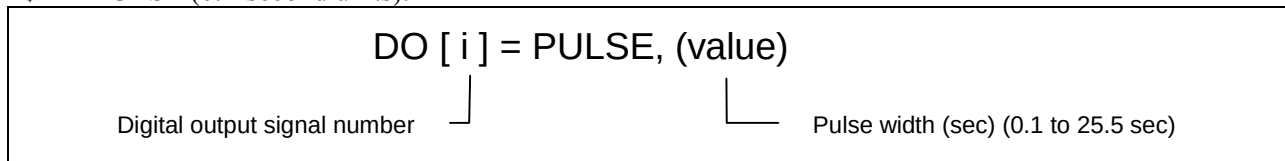


Fig. 4.6.1 (c) Instructions DO[i] = PULSE,(WIDTH)

Example 5: DO[1] = PULSE
6: DO[2] = PULSE, 0.2sec
7: DO[R[3]] = PULSE, 1.2sec

DO[i] = R[i]

The instruction, DO[i]=R[i], turns on or off a specified digital output signal according to the value of a specified register. When the value of the specified register is 0, the digital output signal is turned off. When the value of the specified register is other than 0, the digital output signal is turned on.

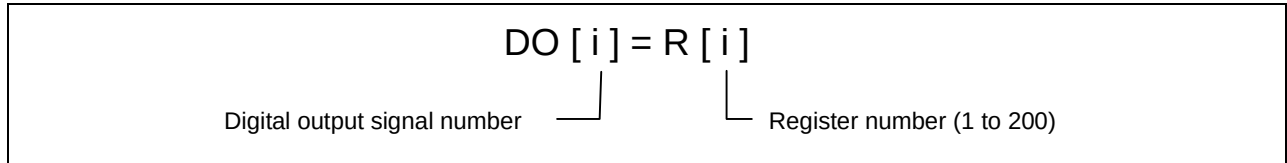


Fig. 4.6.1 (d) Instruction DO[i] = R[i]

Example 7: DO[1] = R[2]
 8: DO[R[5]] = R[R[1]]

4.6.2 Robot I/O Instructions

The robot input signal (RI) and robot output signal (RO) are input/output signals that can be controlled by the user.

R[i] = RI[i]

The instruction, R[i] = RI[i], loads the state of a robot input signal (on = 1/off = 0) into a specified register.

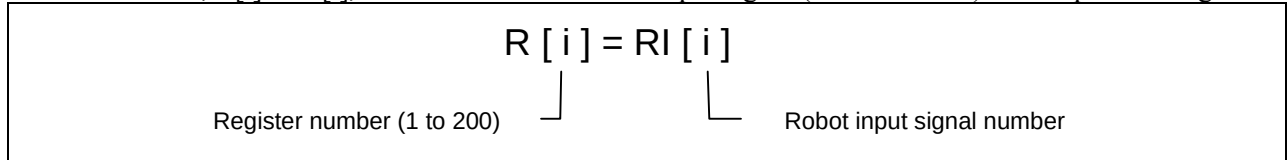


Fig. 4.6.2 (a) Instruction R[i] = RI[i]

Example 1: R[1] = RI[1]
 2: R[R[3]] = RI[R[4]]

RO[i] = ON/OFF

The instruction, RO[i] = ON/OFF, turns on or off a specified robot digital output signal.

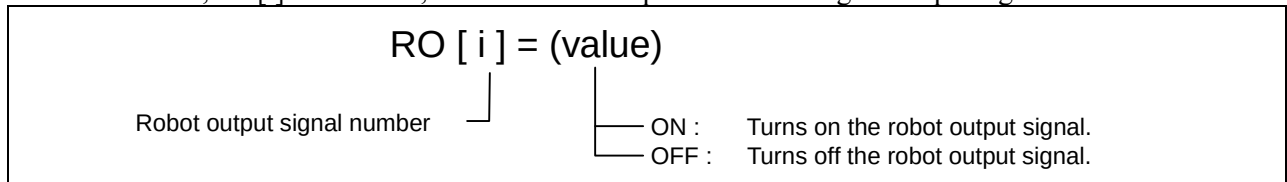


Fig. 4.6.2 (b) Instruction RO[i] = ON/OFF

Example 3: RDDO[1] = ON
 4: RDDO[R[3]] = OFF

RO[i] = PULSE,[WIDTH]

The RO[i] = PULSE,[TIME] instruction inverts the current status of a specified digital output for a specified duration. When no duration is specified, pulse output is executed for the duration specified with \$DEFPULSE (0.1-second units).

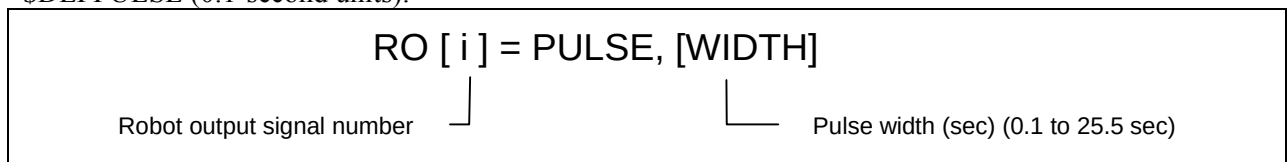


Fig. 4.6.2 (c) Instruction of RO[i] = PULSE,[WIDTH]

Example 5: RO[1] = PULSE
 6: RO[2] = PULSE, 0.2sec
 7: RO[R[3]] = PULSE, 1.2sec

RO[i] = R[i]

The instruction, RO[i] = R[i], turns on or off a specified robot output signal according to the value of a specified register. When the value of the specified register is 0, the robot output signal is turned off. When the value of the specified register is other than 0, the robot output signal is turned on.

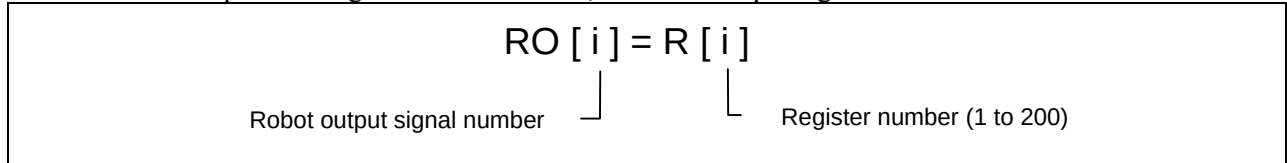


Fig. 4.6.2 (d) Instruction RO[i] = R[i]

Example 7: RO[1] = R[2]
8: RO[R[5]] = R[R[1]]

4.6.3 Analog I/O Instructions

Analog input (AI) and analog output (AO) signals indicate levels as a value on a continuum. Thus, the magnitude of a signal represents a temperature, voltage, or other data.

R[i] = AI[i]

The R[i] = AI[i] instruction stores the value of an analog input signal in a register.

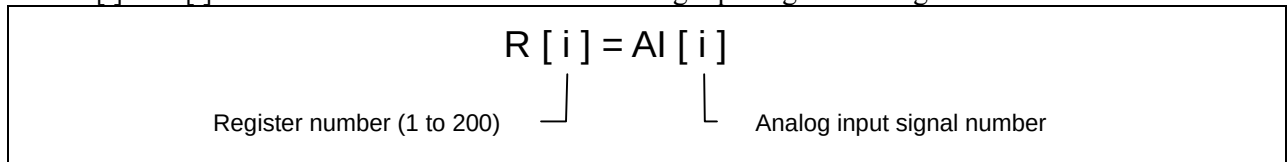


Fig. 4.6.3 (a) R[i] = AI[i] Instruction

Example 1: R[1] = AI[2]
2: R[R[3]] = AI[R[4]]

AO[i] = (value)

The AO[i] = (value) instruction outputs a value as a specified analog output signal.

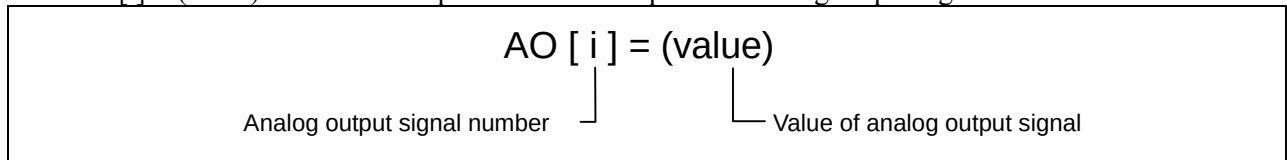


Fig. 4.6.3 (b) AO[i] = (value) Instruction

Example 3: AO[1] = 0
4: AO[R[3]] = 3276

AO[i] = R[i]

The AO[i] = R[i] instruction outputs a register value as an analog output signal.

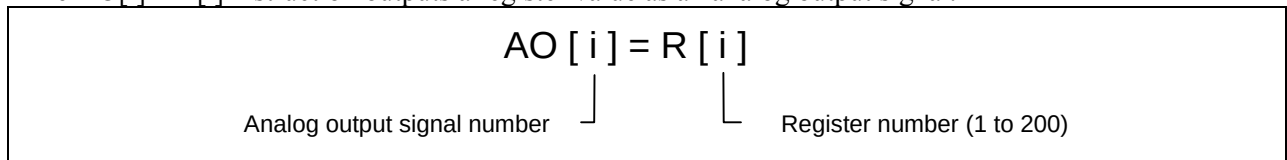


Fig. 4.6.3 (c) AO[i] = R[i] Instruction

Example 5: AO[1] = R[2]
6: AO[R[5]] = R[R[1]]

4.6.4 Group I/O Instruction

$R[i] = GI[i]$

The signal of the group input(GI) and the group output(GO) is that some digital input/output signals are grouped and this is controlled by one instruction.

The instruction, $R[i]=GI[i]$, converts the binary value of the specified group input signal to the decimal value and inputs it to the specified register.

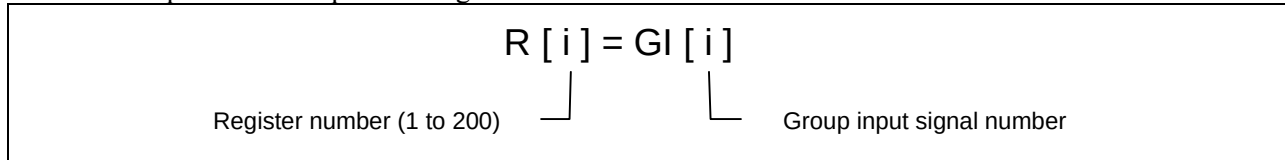


Fig. 4.6.4 (a) Instruction $R[i] = GI[i]$

Example 7: $R[1] = GI[1]$
 8: $R[R[3]] = GI[R[4]]$

$GO[i] = (value)$

The $GO[i]=(VALUE)$ instruction sends the binary equivalent of a value on the specified group output lines.

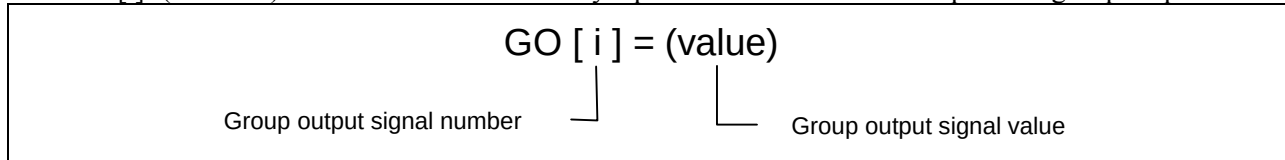


Fig. 4.6.4 (b) Instruction $GO[i] = (value)$

Example 3: $GO[1] = 0$
 4: $GO[R[3]] = 32767$

$GO[i] = R[i]$

The $GO[i]=R[i]$ instruction sends the binary equivalent of the contents of specified register on the specified group output lines.

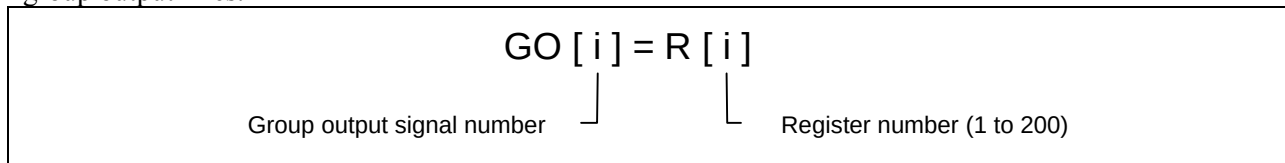


Fig. 4.6.4 (c) Instruction $GO[i] = R[i]$

Example 5: $GO[1] = R[2]$
 6: $GO[R[5]] = R[R[1]]$

4.7 BRANCH INSTRUCTIONS

A branch instruction causes a branch from one line of a program to another. Four types of branch instructions are supported.

- Label instruction
- Program end instruction
- Unconditional branch instruction
- Conditional branch instruction

4.7.1 Label Instruction

LABEL[i]

The label instruction (LBL[i]) is used to specify a program execution branch destination. A label is defined with a label definition instruction.

Instruction		JOINT 30 %
1 Registers	5 JMP/LBL	
2 I/O	6 CALL	
3 IF/SELECT	7 Palletizing	
4 WAIT	8---next page---	
PROGRAM		

A comment can be added to explain a label. Once a label is defined, it can be used for either an unconditional branch or conditional branch. It is not possible to specify the label number as the indirect addressing. To add a comment, move the cursor to the label number and press the ENTER key.

LBL [i : Comment]

Label (1 to 32766) └─┬─ A comment can consist of up to 16 characters including alphanumeric characters, asterisks(*), underlines(_), and at marks (@ etc...).

Fig. 4.7.1 LBL[i] Instruction

Example 1: LBL[1]
2: LBL[R[3]]

4.7.2 Program End Instruction

END

The program end instruction indicates the end of a program. The execution of a program is terminated by this instruction. If a program is called from another main program, control is returned to the main program.

Instruction		JOINT 30 %
1 Registers	5 JMP/LBL	
2 I/O	6 CALL	
3 IF/SELECT	7 Palletizing	
4 WAIT	8---next page---	
PROGRAM		

END

Fig. 4.7.2 Program End Instruction

4.7.3 Unconditional Branch Instructions

An unconditional branch instruction invariably causes a branch from one line to another in the same program. Two types of unconditional branch instructions are supported.

- Jump instruction: Causes a branch to a specified label or program.
- Program call instruction: Causes a branch to another program.

Jump instruction JMP LBL[i]

The JMP LBL[i] instruction transfers program control to a specified label.

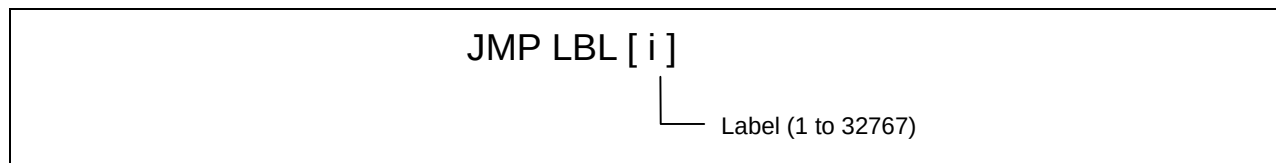


Fig. 4.7.3 (a) JMP LBL[i] Instruction

Example 3: **JMP LBL[2:hand open]**
 4: **JMP LBL[R[4]]**

Program call instruction CALL (program)

The CALL (program) instruction transfers program control to the first line of another program (subprogram) in order to execute it. When a program end instruction (END) in a called program is executed, control is returned to the instruction immediately after the program call instruction in the calling program (main program). To enter the calling program name, select it with the sub-menu automatically displayed or press F5,STRINGS to enter characters directly.

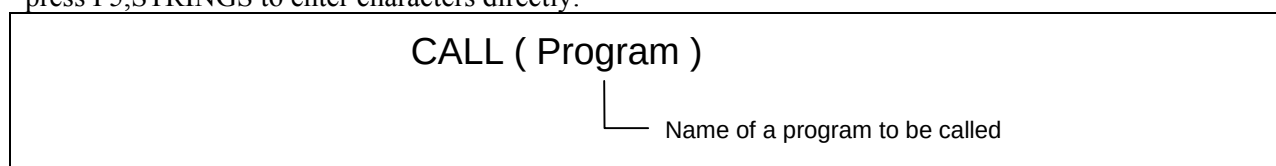


Fig. 4.7.3 (b) CALL (program) Instruction

Example 5: **CALL SUB1**
 6: **CALL PROGRAM2**

- *) It is possible to set an argument for the program call instruction and use its value in a subprogram. See Subsection 4.7.5, "Arguments" for Details.

4.7.4 Conditional Branch Instructions

A conditional branch instruction causes a branch from one location in a program to another when some condition is satisfied. Two types of conditional branch instructions are available.

Instruction		JOINT 30 %
1 Registers	5 JMP/LBL	
2 I/O	6 CALL	
3 IF/SELECT	7 Palletizing	
4 WAIT	8---next page---	
PROGRAM		

- Conditional compare instruction: Causes a branch to a specified label or program when some condition is satisfied. The register conditional compare instruction and I/O conditional compare instruction are available.
- Conditional select instruction: Causes a branch to a specified jump instruction or subprogram call instruction according to the value of a register.

Register conditional compare instruction**IF R[i] (operator) (value) (processing)**

A register conditional compare instruction compares the value stored in a register with another value and, when the compare condition is satisfied, executes processing.

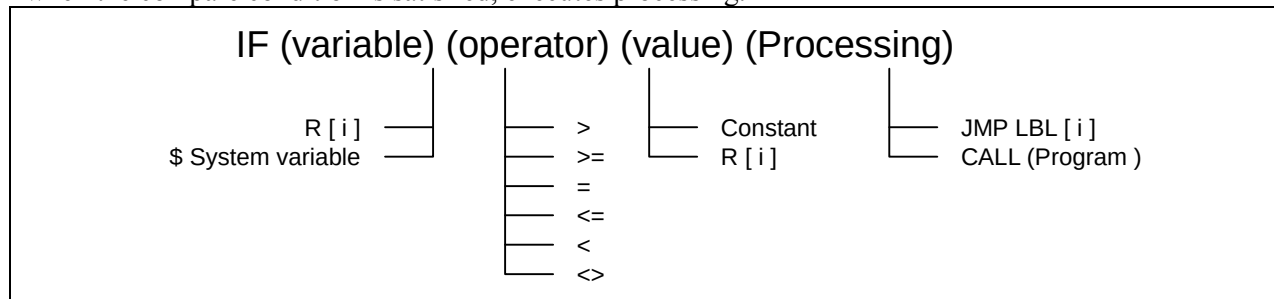


Fig. 4.7.4 (a) Register Conditional Compare Instruction

⚠ CAUTION

When the contents of a register is compared with the real value using the operator "=", the contents does not always correspond to the real value because of the rounding-off error of the contents. To compare with the real value, use the operator without an equal sign.

I/O conditional compare instruction**IF (I/O) (operator) (value) (processing)**

The I/O conditional compare instruction compares the value of an input/output signal with another value. When the comparison condition is satisfied, specified processing is executed.

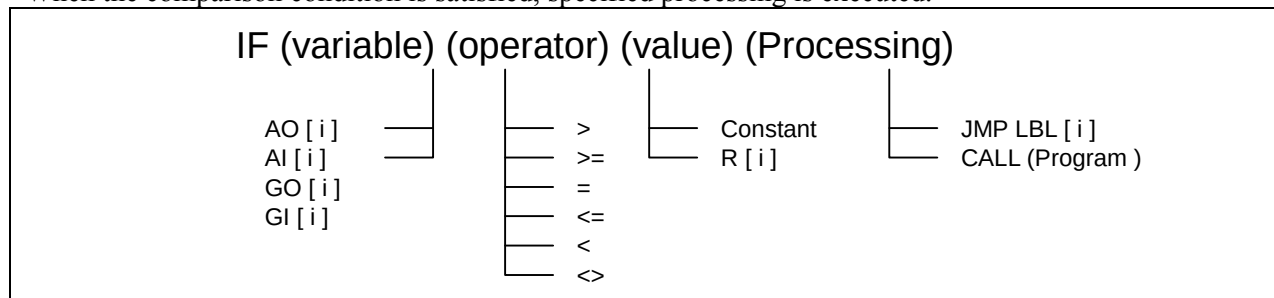


Fig. 4.7.4 (b) I/O Conditional Compare Instruction 1

Example 7: IF R[1] = R[2], JMP LBL[1]
 8: IF AO[2] >= 3000, CALL SUBPR01
 9: IF GI[R[2]] = 100, CALL SUBPR02

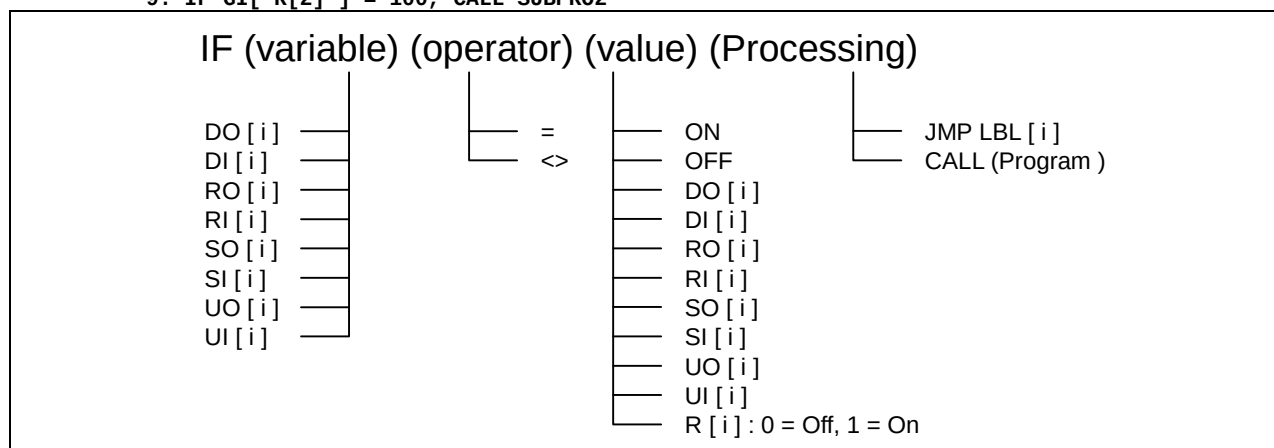


Fig. 4.7.4 (c) I/O Conditional Compare Instruction 2

Example 10: IF R0[2] <> OFF, JMP LBL[1]
 11: IF DI[3] = ON, CALL SUB1

In a conditional branch instruction, multiple conditions can be specified on a single line in the condition statement, using the logical operators ("and" and "or"). This simplifies the program structure, allowing the conditions to be evaluated efficiently.

Instruction format

- Logical product (and)
 IF <condition 1> and <condition 2> and <condition 3>, JMP LBL [3]
- Logical sum (or)
 F <condition 1> or <condition 2>, JMP LBL [3]

If the "and" (logical product) and "or" (logical sum) operators are used together, the logic becomes complex, impairing the readability of the program and ease of editing. For this reason, this function prohibits the use of the logical operators "and" and "or" in combination.

If multiple "and" (logical product) or "or" (logical sum) operators are specified for an instruction on a single line, and one of the operators is changed from "and" to "or" or from "or" to "and," all other "and" or "or" operators are changed accordingly, and the following message appears:

TPIF-062 AND operator was replaced to OR

TPIF-063 OR operator was replaced to AND

Up to five conditions can be combined with "and" or "or" operators on a single line.

Example IF <condition 1> and <condition 2> and <condition 3> and <condition 4> and <condition 5>, JMP LBL [3]

Pallet register conditional compare instruction

IF PR [i] (operator) (value) (processing)

The pallet register conditional compare instruction compares the value of the pallet register with the value of another pallet register element. When the comparison is true, specified processing is executed. If 0 is entered in each element, "*" appears. Only a numeric value or remainder specification can be used for each element to be compared.

For the pallet register element, specify the element whose value is to be compared with the value of the pallet register. There are four methods for specification.

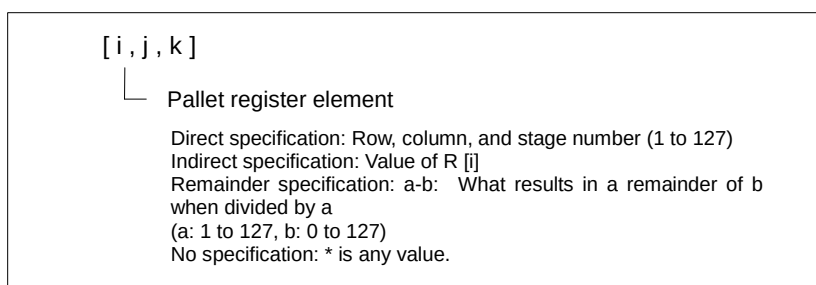


Fig. 4.7.4 (d) Pallet Register Element Format

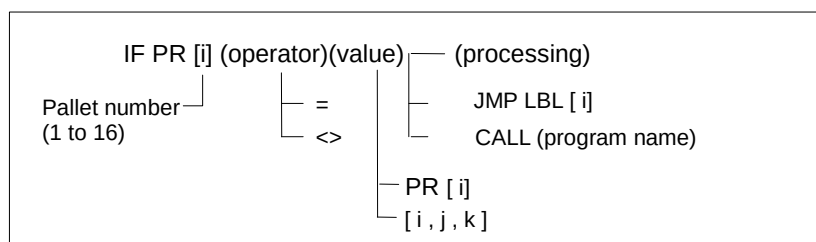


Fig. 4.7.4 (e) Pallet Register Conditional Compare Instruction

Example

```
12: IF PR[1] = R[2], JMP LBL [1]
13: IF PR[2] <> [1, 1, 2] , CALL SUB1
14: IF PR[R[3]] <> [*, *, 2-0] , CALL SUB1
```

Conditional select instruction

SELECT R[i] = (value) (processing) = (value) (processing) = (value) (processing) ELSE (processing)

The conditional select instruction consists of several register compare instructions. The conditional select instruction compares the value of a register with one or more values, then selects a statement that satisfies the comparison condition.

- If the value of a specified register matches one value, the jump instruction or subprogram call instruction corresponding to the value is executed.
- If the value of a specified register does not match any of the values, the jump instruction or subprogram call instruction corresponding to ELSE is executed.

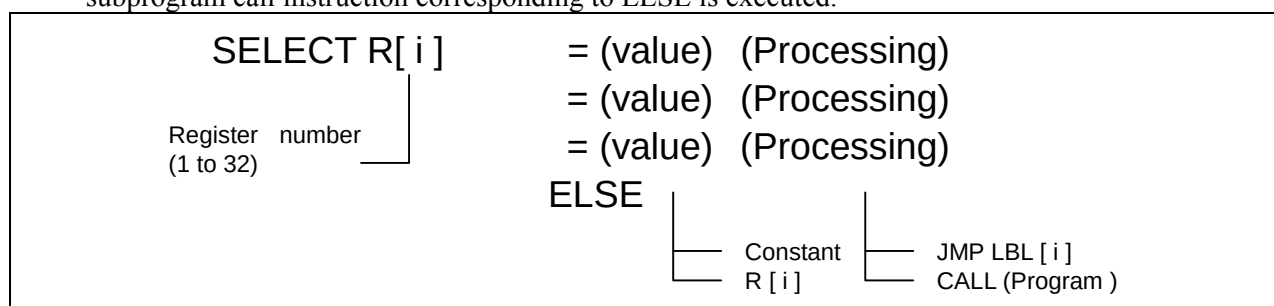


Fig. 4.7.4 (f) Conditional Select Instruction

Example

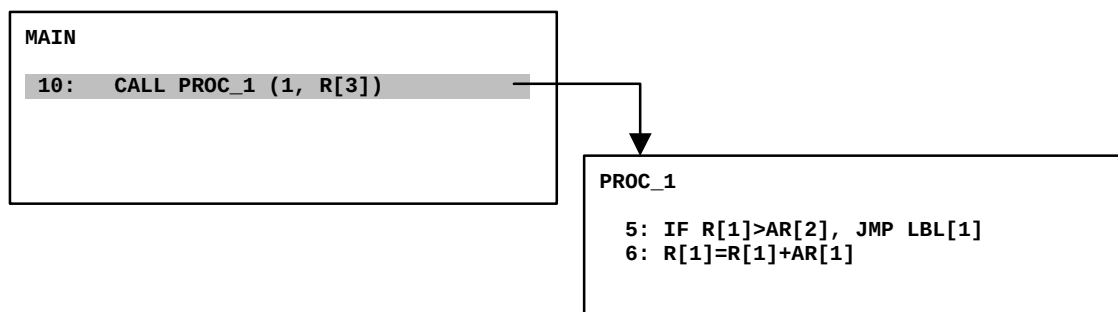
```
11: SELECT R[1] = 1, JMP LBL[1]
12:      = 2, JMP LBL[2]
13:      = 3, JMP LBL[2]
14:      = 4, JMP LBL[2]
15:      ELSE, CALL SUB2
```

4.7.5 Arguments

By using "arguments" and "argument registers," it is possible to transfer data between two programs only.

Example)

In this example, the main program MAIN calls the subprogram PROC_1 with two arguments. PROC_1 can use the values of the arguments with the argument registers. The first argument corresponds to AR[1] while the second argument corresponds to AR[2].



Arguments can be used in macro instructions in the same way.

Argument types

The following arguments are supported.

Table 4.7.5 (a) Argument types

Argument type	Example
Constant	1, 3.5
Character string	'Perch'
Argument register *1	AR[3]
Register	R [6]
String Register	SR [4]

*1 Used as argument registers in subprograms.

```

CALL PROC_1      (1,  AR[1], R[6])
                  ↓   ↓   ↓
PROC_1 :         AR[1], AR[2], AR[3]

```

Instructions for which arguments can be set

Table 4.7.5 (b) Instructions for Which Arguments Can be Set

Instruction	Example
Program call instruction	CALL SUBPRG (1, R[3], AR[1])
Macro instruction	Vacuum hand open (2.5)

⚠ CAUTION

A program call used for branching with an instruction such as a conditional branch instruction cannot use arguments. This problem can be solved by programming as follows:

```

(Arguments cannot be set) (Arguments can be set)
IF R[1]=3, CALL PROC_5 → IF R[1]<>3, JMP LBL[1]
                        CALL PROC_5 (1, R[2])
                        LBL[1]

```

Instructions that can use argument registers

Table 4.7.5 (c) Instructions That Can Use Argument Registers

Instruction	Example
Right side of an instruction and conditional expression having a register on the left side	R[1]=AR+R[2]+AR[4] F R[1]=AR[1], JMP LBL[1]
Right side of the analog output (AO[]) and group output (GO[]) instructions	AO[1]=AR[2] GO[1]=AR[2]
Right side of a conditional expression having analog input/output (AI[]/AO[]) or group input/output (GI[]/GO[]) on the left side	IF AO[1]=AR[1], JMP LBL[1] WAIT GI[1]<>AR[2], TIMEOUT, LBL[1]
Right side of the user coordinate system selection instruction and the tool coordinate system selection instruction	UTOOL_NUM=AR[4]
Indirect index specification	R[AR[1]]=R[AR[2]] DO[AR[1]]=ON
Argument of a program call instruction	CALL SUBPRG1 (AR[5])
Argument of a macro instruction	hand 3 open (AR[1])

Restrictions on arguments

The following restrictions are imposed on arguments:

- Up to 10 arguments can be set.
- An argument of character string type can be 1 to 33 characters in length. (An argument with 0 characters is regarded as being uninitialized.)
- An indirect specification cannot be used for an already indirectly specified element of an index.
 - [AR[1]]
 - × [R[AR[1]]]
- The value stored in an argument register cannot be changed in a subprogram.

Specifying arguments

When a program call instruction or macro instruction is specified, the cursor stops at the end of the line. If no arguments need be specified, press the ENTER key or "→" or "↓" key to move the cursor to the next line.

To display the argument selection submenu, press function key [CHOICE].

JOINT 10%	
CALL PROC_1	
[CHOICE]	

Parameter select		JOINT 10 %
1 R[]	5 <None>	
2 Constant	6 <Insert>	
3 String	7 SR[]	
4 AR[]	8	

Specifying arguments of the constant type

To specify an argument of the constant type, press the [CHOICE] function key and select "1 Constant" from the submenu (see "Specifying arguments").

Parameter selec	1: CALL PROC_1(Constant)
1 R[]	
2 Constant	Enter value
3 String	
4 AR[]	

1:CALL PROC_1(1) Enter 1.

Specifying arguments of character type

To specify an argument of character type, press function key [CHOICE] and select String from the submenu (see "Specifying arguments"). The character string type selection menu appears.

Parameter select	
1 R[]	
2 Constant	
3 String	
4 AR[]	

String select		JOINT 10 %
1 PARTS	5 DEV	
2 TOOL	6 PALT	
3 WORK	7 GRIP	
4 POS	8 --- next page ---	
PROGRAM1		1/2
1: CALL PROC_1 (1, '...')		
[End]		
[CHOICE] [STRING]		

F4

F5

When a character string type is selected, the character string selection menu appears.

String select		JOINT 10 %
1 Parts ITEM1	5 Parts ITEM5	
2 Parts ITEM2	6 Parts ITEM6	
3 Parts ITEM3	7 Parts ITEM7	
4 Parts ITEM4	8 --- next page ---	

Select a character string from the menu. The character string is confirmed.

1: CALL PROC_1 ('Parts ITEM2') Select Parts ITEM2 from the menu

To enter a character string directly, press function key STRINGS from the character type selection menu or the character string selection menu.

		JOINT 10 %
1 Words		
2 Upper Case		
3 Lower Case		
4 Options		
MAIN		
1: CALL PROC_1 (1, 'Tool12		1/2
[End]		')
\$	[	]

F1
F2
F3
F4
F5

Press the Enter key to confirm the character string.

1: CALL PROC_1 ('Tool 12') 'Tool2' is fixed.

To change a character string, move the cursor to the character string and press the [CHANGE] function key. The character string type selection menu appears.

		JOINT 10 %
CALL PROC_1 (1, 'TOOL12')		
[CHOICE]	CHANGE	

F5

Specifying arguments of the argument register type

To set an argument of the argument register type, press the [CHOICE] function key and select AR[] from the submenu (see "Specifying arguments").

```
Parameter select
1 R[ ]
2 Constant
3 String
4 AR[ ]
```

1: CALL PROC_1 (AR[...])

Enter an index.

1: CALL PROC_1 (AR[1]) Enter 1 to the index of the argument register.

To toggle between direct and indirect index specifications, press the [INDIRECT] function key. The display changes as follows:

AR[R[...]] → AR[AR[...]] → AR[R[...]] → ...

Specifying arguments of the register type

To set an argument of register type, press the [CHOICE] function key and select "4 R[]" from the submenu (see "Specifying arguments").

```
Parameter select
1 R[ ]
2 Constant
3 String
4 AR[ ]
```

1: CALL PROC_1 (R[...])

Enter an index.

1: CALL PROC_1 (R[1])

To toggle between direct and indirect index specifications, press the [INDIRECT] function key. The display changes as follows:

R[R[...]] → R[AR[...]] → R[R[...]] → ...

Adding arguments

Move the cursor to "*)" at the end of the line.

1: CALL PROC_1 (1)

Press function key CHOICE and select an argument type from the submenu (see "Specifying arguments"). A new argument can be added to the cursor position.

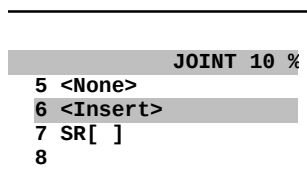
1: CALL PROC_1 (1, Constant) In case that Constant is selected

Select an argument type and set a value.

1: CALL PROC_1 (1, **Constant**) Select the constant type
 1: CALL PROC_1 (1, **2**) Set a value of "2"

Inserting arguments

Move the cursor to the argument for which an argument is to be inserted.



1: CALL PROC_1 (1, **2**)

Press function key [CHOICE] and select <Insert> from the submenu (see "Specifying arguments"). A new argument can be inserted at the cursor position.

1: CALL PROC_1 (1, **...**, 2)

Select an argument type and set a value, index, and so on.

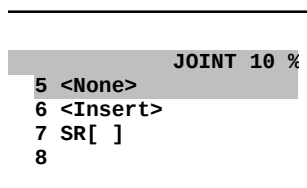
1: CALL PROC_1 (1, R[**...**], 2) Select the constant type
 1: CALL PROC_1 (1, R[3], **2**) Set a value of "3"

NOTE

An argument cannot be inserted when no argument has been set, and at ")" at the end of a line. The same submenu reappears; select the argument type.

Deleting arguments

Position the cursor to the argument to be deleted.



1: CALL PROC_1 (1, **2** , 3)

Press function key [CHOICE] and select <None> from the submenu (see "Specifying arguments"). The argument is deleted from the cursor position.

1: CALL PROC_1 (1, **3**)

NOTE

Selecting <None> when no argument has been set, and at ")" at the end of a line, simply closes the submenu; no argument is deleted.

Specifying argument registers

The following explanation uses a register instruction as an example.

The selections for the right side of a register instruction are as follows:

JOINT 10 %

R[1]=

[CHOICE]

F4

REGISTER StatementJOINT 10 %

1 R[]5 AI[]

2 AR[]6 GO[]

3 Constant7 GI[]

4 AO[]8 --- next page ---

To use an argument with the instruction, select AR[] from the menu.

1: R[1]=AR[]

Specify the index.

1: R[1]=AR[1]

DIRECT INDIRECT [CHOICE]

F3

If function key F3 "Indirect specification" is pressed twice at an element having an index, an argument register can be used for an indirect index specification.

1: WAIT R[]

1: WAIT R[AR]

When F3 is pressed once

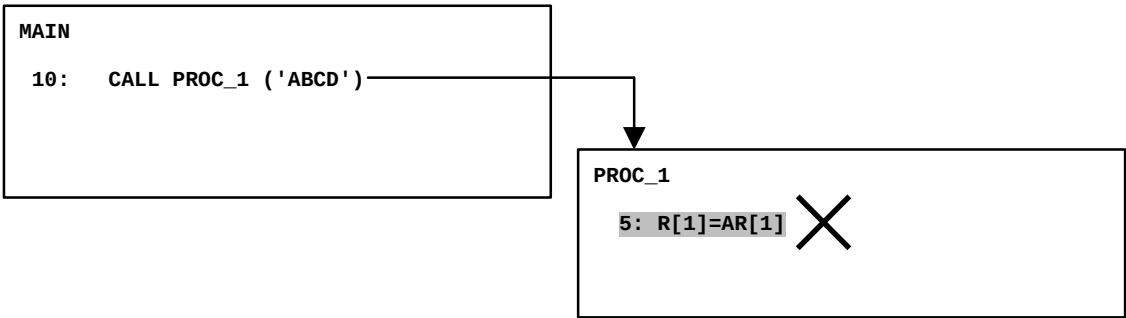
When F3 is pressed again

Notes on using arguments

Note the following when specifying arguments:

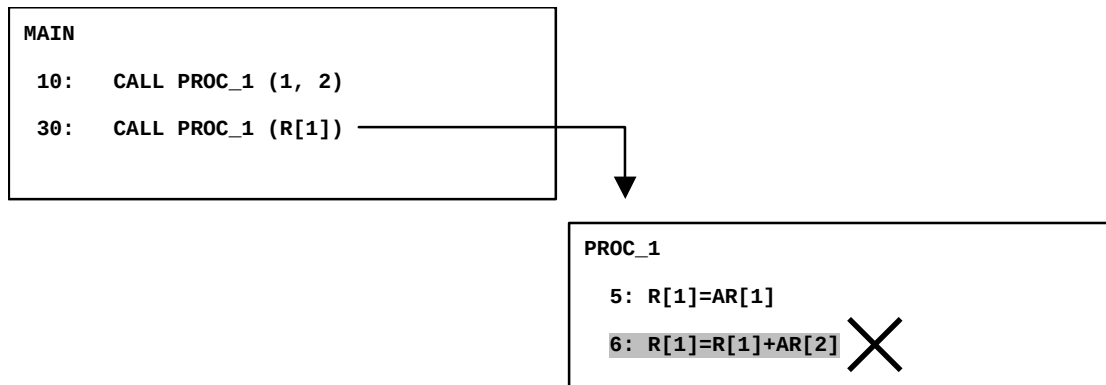
- The contents of an argument are not checked when the argument is specified. If the type of an argument does not match the type of the corresponding one in the subprogram, an error occurs during execution.

Example In this example, although a value of AR[1] is assigned to the register in subprogram PROC_1, an argument of character string type is specified in the main program. An error occurs when line 5 of the subprogram is executed.



- The number of arguments is not checked when arguments are specified. Even if the number of arguments is not correct, no errors occur if the arguments specified in the main program are not used in a subprogram.

Example In this example, only one argument is specified in the main program, but two arguments are used in subprogram PROC_1. An error occurs when line 6 of PROC_1 is executed.



Notes on specifying arguments for a program call instruction

- When the program name is changed, the arguments that have been set are kept intact.
- When the program call instruction itself is re-specified, not only the program name but all the arguments are deleted.

Notes on specifying arguments for a macro instruction

- When the macro name is changed, those arguments that have been set are kept intact.

Notes on execution

As described in "Notes on using arguments," the contents and number of arguments to be passed between the calling program and the called program are not checked when they are specified. If an argument is set or used incorrectly, an error occurs on a line where a conflict is detected during program execution.

- Check that the number of arguments specified in the main program is equal to that of the arguments used in the subprogram.
- If the arguments specified in the main program are not used in the subprogram, an error does not occur.
- Check that the contents of the arguments specified in the main program match the types of instructions in the subprogram that use those arguments.
- Check that the indexes and values of the specified arguments are set correctly.

1: CALL PROC_1 (Constant)	An error occurs because the value is uninitialized
2: CALL PROC_1 (R[...])	The index is uninitialized

When lines containing these are executed, the error "INTP-201 Unspecified Statement" occurs.

System variables relating to arguments

The argument-attached program call/macro instruction function displays, as selections, the character strings set as system variables when an argument of the character string type is to be selected. These system variables are given below.

Table 4.7.5 (d) System Variables Relating to Arguments

Item	System variable	Remarks
Use of string data	\$STRING_PRM=TRUE/FALSE Standard value=TRUE	
String data type	\$ARG_STRING[i].\$TITLE (i = 1 to 10)	More than 1 and up to 16 characters
String data	\$ARG_STRING[i].\$ITEMj (i = 1 to 10, j=1 to 20)	Up to 16 characters
Word to enter string data	\$ARG_WORD [i](i = 1 to 5)	Up to 7 characters

4.8 WAIT INSTRUCTIONS

A wait instruction is used to stop program execution for a specified period of time or until a condition is satisfied. Two types of wait instructions are available.

Instruction		JOINT 30 %
1 Registers	5 JMP/LBL	
2 I/O	6 CALL	
3 IF/SELECT	7 Palletizing	
4 WAIT	8 ---next page---	
PROGRAM		

- Time-specified wait instruction: Waits program execution for a specified period of time.
- Conditional wait instruction: Waits program execution until a specified condition is satisfied or a specified period of time has elapsed.

4.8.1 Time-specified Wait Instruction

WAIT (TIME)

The time-specified wait instruction waits program execution for a specified period of time (in seconds).

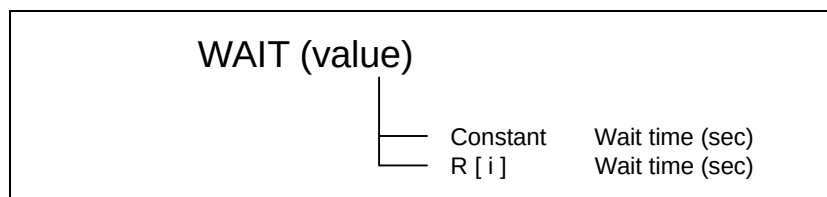


Fig. 4.8.1 Time-Specified Wait Instruction

Example 1: WAIT
 2: WAIT 10.5sec
 3: WAIT R[1]

4.8.2 Conditional Wait Instructions

WAIT (condition) (processing)

A conditional wait instruction waits program execution until a specified condition is satisfied or a specified period of time has elapsed. Two methods of specifying time-out processing are available:

- If no processing is specified, program execution is waits until a specified condition is satisfied.
- Timeout, LBL[i] is transferred to a specified label if the specified condition is not satisfied until the time specified in 14 WAIT timeout on the system configuration screen.

Register conditional wait instruction

The register conditional wait instruction compares the value of a register with another value, and waits until the comparison condition is satisfied,

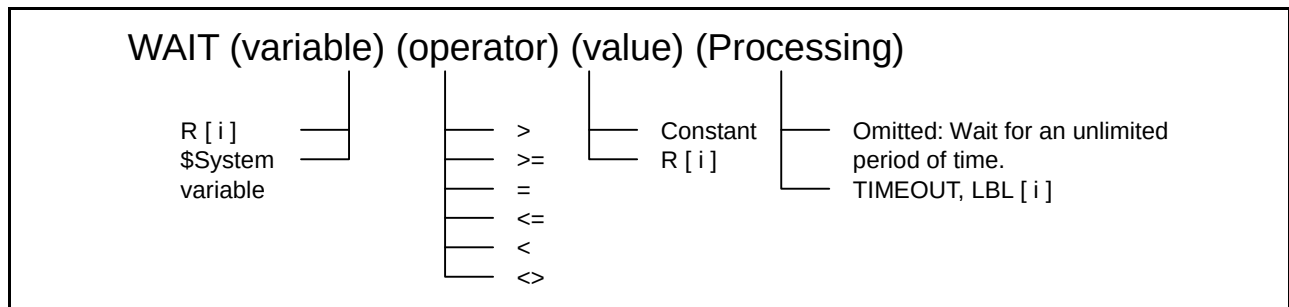


Fig. 4.8.2 (a) Register Conditional Wait Instruction

Example 3: WAIT R[2] <> 1, TIMEOUT LBL[1]
 4: WAIT R[R[1]] > = 200

I/O conditional wait instruction

The I/O conditional wait instruction compares the value of an input/output signal with another value, and waits until the comparison condition is satisfied.

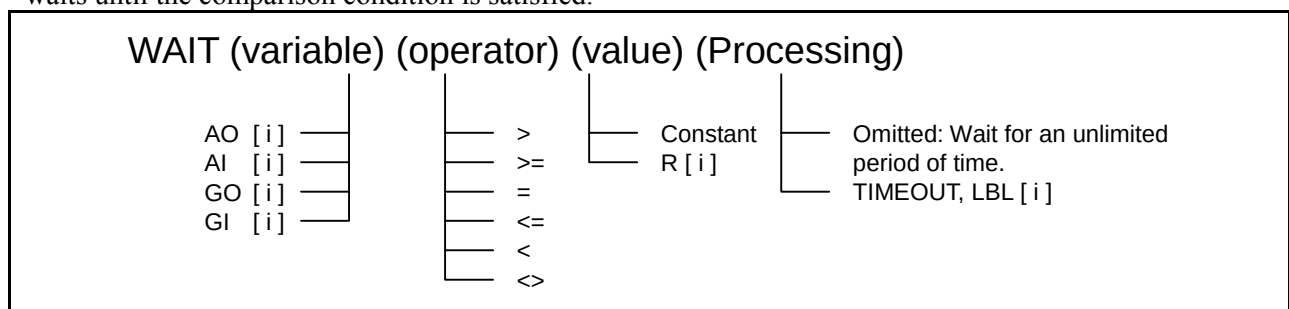


Fig. 4.8.2 (b) I/O Conditional Wait Instruction 1

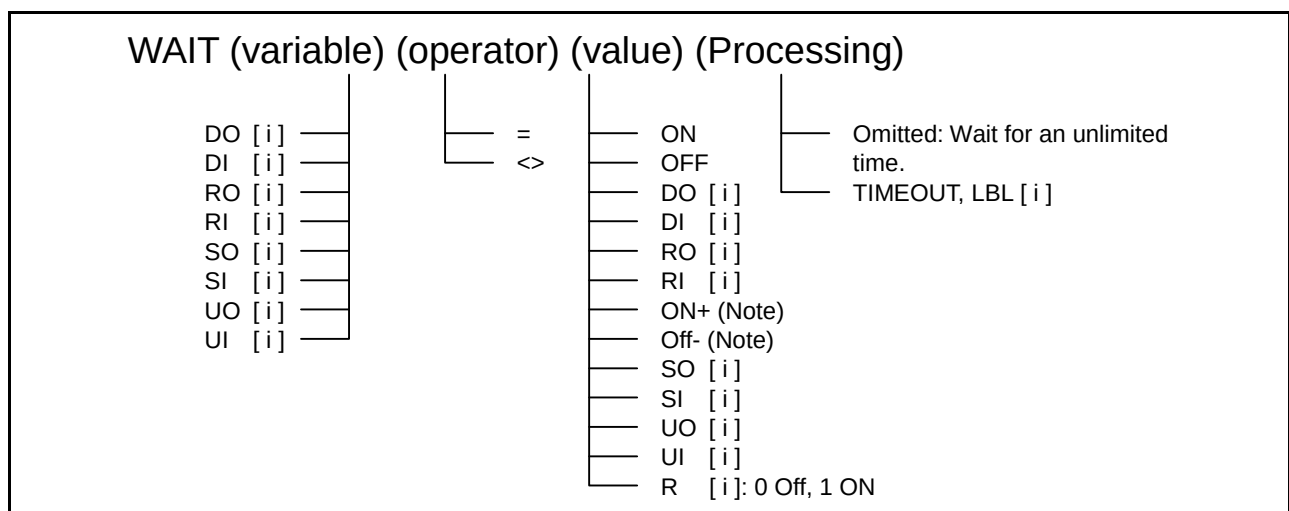


Fig. 4.8.2 (c) I/O Conditional Wait Instruction 2

Example 5: WAIT DI[2] <> OFF, TIMEOUT LBL[1]
 6: WAIT RI[R[1]] = R[1]

NOTE

Off-:

The falling edge of a signal is regarded as being a detection condition. The condition is not satisfied while the signal remains off. The detection condition is satisfied when the signal changes from the on state to the off state.

On+:

The rising edge of a signal is regarded as being a detection condition. The condition is not satisfied while the signal remains on. The detection condition is satisfied when the signal changes from the off state to the on state.

Error condition wait instruction

The error condition wait instruction waits for the occurrence of an alarm having a specified error number.

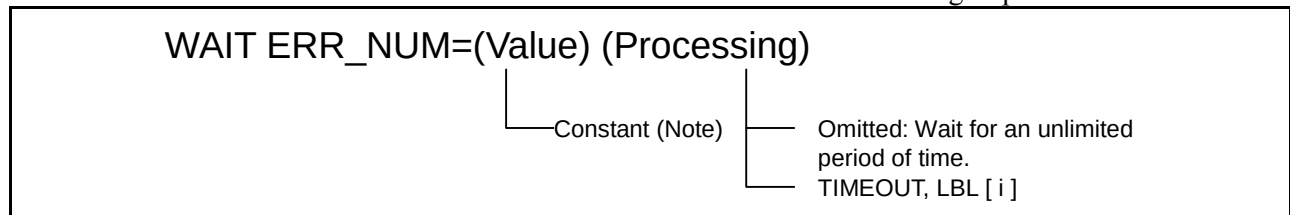


Fig. 4.8.2 (d) Error condition wait instruction

NOTE

An error number is specified with an alarm ID followed by an alarm number.

Error number = aabbbb

where aa = alarm ID

bbb = alarm number

For an explanation of alarm IDs and numbers, refer to the Alarm Code List Operator's Manual.

Example

For SRVO-006 HAND broken, the servo alarm ID is 11, and the alarm number is 006. Thus,

Error number = 11006

In the condition wait instruction, multiple conditions can be specified on a single line in the condition statement, using the logical operators ("and" and "or"). This simplifies the program structure, allowing the conditions to be evaluated efficiently.

Instruction format

- Logical product (and)
WAIT <condition 1> and <condition 2> and <condition 3>
- Logical sum (or)
WAIT <condition 1> or <condition 2> or <condition 3>

If the "and" (logical product) and "or" (logical sum) operators are used in combination, the logic becomes complex, impairing the readability of the program and the ease of editing. For this reason, this function prohibits the use of the logical operators "and" and "or" in combination.

If multiple "and" (logical product) or "or" (logical sum) operators are specified for an instruction on a single line, and one of the operators is changed from "and" to "or" or from "or" to "and," all other "and" or "or" operators are changed accordingly, and the following message appears:

TRIF-062 AND operator was replaced to OR

TRIF-063 OR operator was replaced to AND

Up to five conditions can be combined with "and" or "or" operators on a single line.

Example)

WAIT <condition 1> and <condition 2> and <condition 3> and <condition 4> and <condition 5>

4.9 SKIP CONDITION INSTRUCTION

The skip condition instruction specifies, in advance, a skip condition (condition for executing a skip instruction) used with a skip instruction. Before a skip instruction can be executed, a skip condition instruction must be executed. A skip condition once specified is valid until the execution of the program is completed, or the next skip condition instruction is executed.

Instruction	
1 Miscellaneous	5 MACRO
2 Program control	6 SENSOR
3 Skip	7 Multiple control
4 Offset/Frames	8 ---next page---

A skip instruction causes a jump to a branch destination label if the skip condition is not satisfied. If the skip condition is satisfied, a skip instruction causes the robot to suspend the current motion toward a target point, instead executing the program instruction on the next line.

If the skip condition is currently not satisfied, a skip instruction causes a jump to a destination label upon the completion of the current motion.

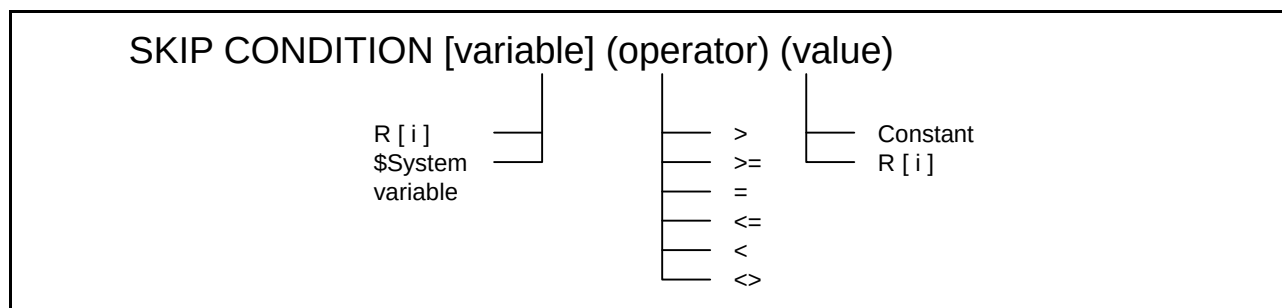


Fig. 4.9 (a) Skip Condition Instruction (Register Condition)

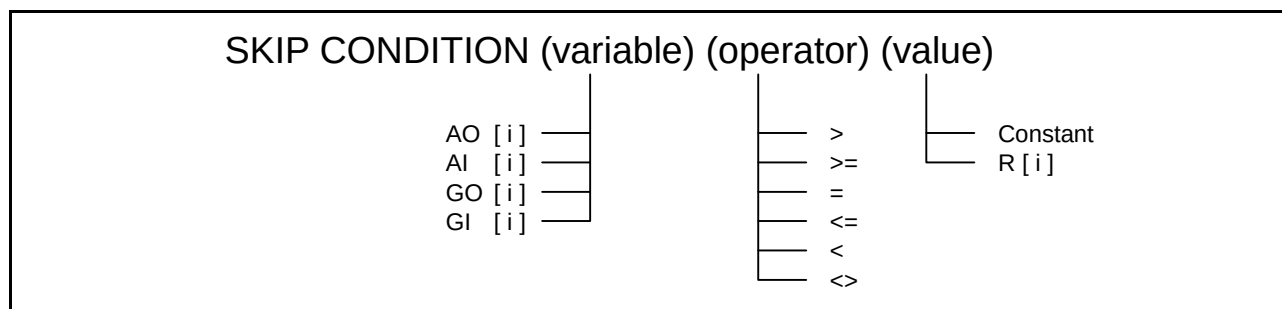


Fig. 4.9 (b) Skip Condition Instruction (I/O condition 1)

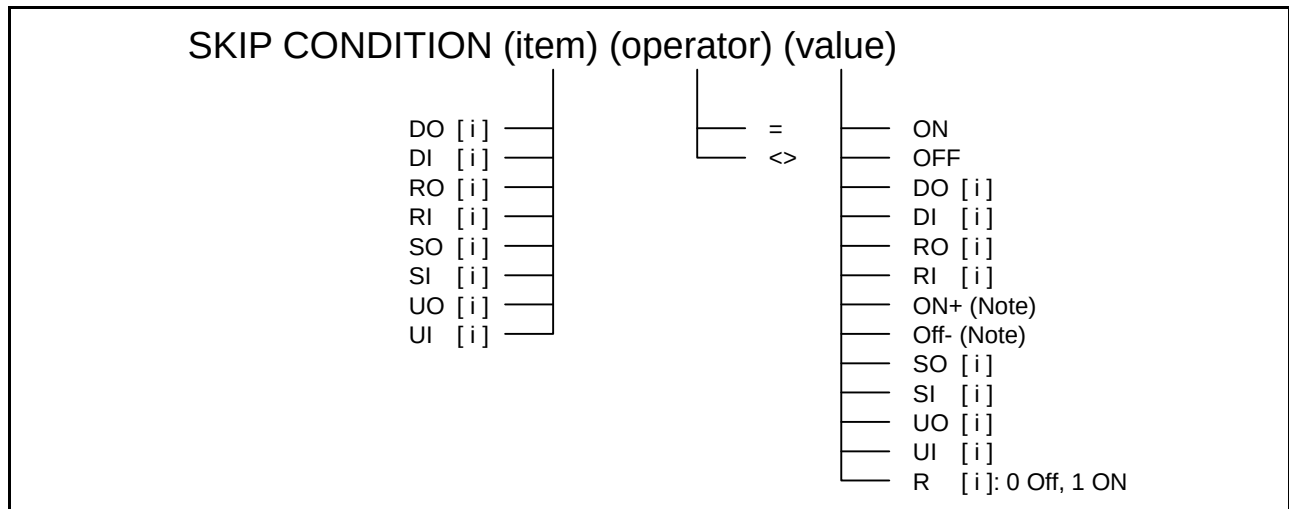


Fig. 4.9 (c) Skip Condition Instruction (I/O condition 2)

Example 1: **SKIP CONDITION DI[R[1]] <> ON**
 2: **J P[1] 100% FINE**
 3: **L P[2] 1000mm/sec FINE Skip, LBL[1]**
 4: **J P[3] 50% FINE**
 5: **LBL[1]**
 6: **J P[4] 50% FINE**

NOTE

Off-:

The falling edge of a signal is regarded as being a detection condition. The condition is not satisfied while the signal remains off. The detection condition is satisfied when the signal changes from the on state to the off state.

On+:

The rising edge of a signal is regarded to be a detection condition. The condition is not satisfied while the signal remains on. The detection condition is satisfied when the signal changes from the off state to the on state.

SKIP CONDITION ERR_NUM=(Value)

Constant (Note)

Fig. 4.9 (d) Skip Condition Instruction (Error condition)

NOTE

An error number is specified with an alarm ID followed by an alarm number.

Error number = aabbb

where aa = alarm ID

bbb = alarm number

For an explanation of alarm IDs and numbers, refer to the Alarm Code List Operator's Manual.

Example

For SRVO-006 Hand broken, the servo alarm ID is 11, and the alarm number is 006. Thus,

Error number = 11006

In the skip condition instruction, multiple conditions can be specified on a single line in the condition statement, using the logical operators ("and" and "or"). This simplifies the program structure, allowing the conditions to be evaluated efficiently.

Instruction format

- Logical product (and)
SKIP CONDITION <condition 1> and <condition 2> and <condition 3>
- Logical sum (or)
SKIP CONDITION <condition 1> or <condition 2> or <condition 3>

If the "and" (logical product) and "or" (logical sum) operators are used in combination, the logic becomes complex, impairing the readability of the program and ease of editing. For this reason, this function prohibits the use of the logical operators "and" and "or" in combination.

If multiple "and" (logical product) or "or" (logical sum) operators are specified for an instruction on a single line, and one of the operators is changed from "and" to "or" or from "or" to "and," all other "and" or "or" operators are changed accordingly, and the following message appears:

TRIF-062 AND operator was replaced to OR

TRIF-063 OR operator was replaced to AND

Up to five conditions can be combined with "and" or "or" operators on a single line.

Example)

SKIP CONDITION <condition 1> and <condition 2> and <condition 3> and <condition 4> and <condition 5>

4.10 OFFSET CONDITION INSTRUCTION

The OFFSET CONDITION instruction specifies the offset condition used in the OFFSET instruction, in advance. The OFFSET CONDITION is needed to be executed before the OFFSET instruction is executed. The specified offset condition is effective until the program execution finishes or the next OFFSET CONDITION instruction is executed. (For the offset instruction, see Subsection 4.3.5.)

Instruction		JOINT 30 %
1 Miscellaneous	5 MACRO	
2 Program control	6 SENSOR	
3 Skip	7 Multiple control	
4 Offset/Frames	8 ---next page---	
PROGRAM		

- The position register specifies the shifting direction and the shift amount.
- When the positional information is expressed in the joint frame, the shift amount of each axis is applied.
- When the positional information is expressed in the Cartesian coordinate system, the number of user frame by which the offset condition is decided should be specified. When it is not specified, the user frame being selected now is used.



CAUTION

If teaching is made by joint coordinates, changing the user coordinate system does not affect the position variables and position registers. However, note that both position variables and registers are affected by the user coordinate systems when the robot is taught in the Cartesian format.

The OFFSET instruction shifts positional information programmed at the destination position by the offset amount specified by position register, and moves the robot to the shifted position. The shifting condition is specified by the OFFSET CONDITION instruction.

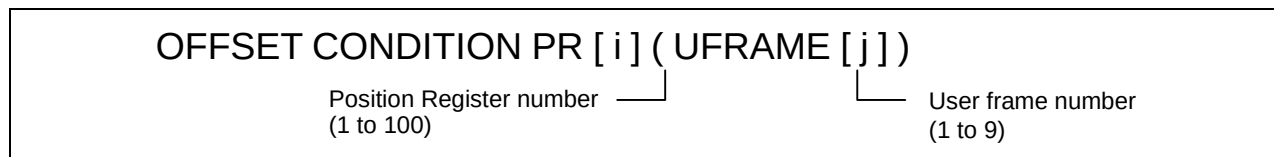


Fig. 4.10 Offset Conditional Instruction

Example 1: **OFFSET CONDITION PR[R[1]]**
 2: **J P[1] 100% FINE**
 3: **L P[2] 500mm/sec FINE Offset**

4.11 TOOL OFFSET CONDITION INSTRUCTIONS

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. Once the tool offset conditions have been specified, they remain effective until the program terminates or the next tool offset condition instruction is executed. (For the tool offset instruction, see Subsection 4.3.5 "Additional motion instructions")

Instruction		JOINT 30 %
1	Miscellaneous	5 Tool_Offset
2	Program control	6 MACRO
3	SKIP	7
4	Offset/Coordinate	8 ---next page---
PROGRAM		

- 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 omitted, the currently selected tool coordinate system is used.
- When the position data is expressed in the joint coordinates, an alarm is issued and the program stops temporarily.

A tool offset instruction moves the robot to a position shifted from the target position, recorded in the position data, by the offset specified in the tool offset conditions. The condition when the offset is applied is specified by a tool offset condition instruction.

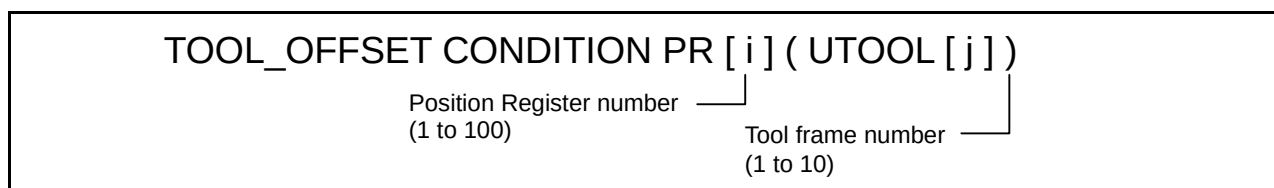


Fig. 4.11 Tool Offset Condition Instruction

Example 1: **TOOL_OFFSET PR[1]**
 2: **J P[1] 100% FINE**
 3: **L P[2] 500mm/sec FINE Tool_Offset**

4.12 FRAME INSTRUCTIONS

The FRAME instruction is used to change the setting of the Cartesian coordinate system by which the robot works. There are two kinds in the FRAME instruction.

Instruction		JOINT 30 %
1	Miscellaneous	5 MACRO
2	Program control	6 SENSOR
3	SKIP	7 Multiple control
4	Offset/Coordinate	8 ---next page---
PROGRAM		

- Frame setup instruction - The definition of the specified frame is changed.
- Frame select instruction - The frame number being selected now is changed.

The frame setup instruction

The tool frame setup instruction changes the setting of the tool frame specified by the tool frame number in this instruction. The user frame setup instruction changes the setting of the user frame specified by the user frame number in this instruction.

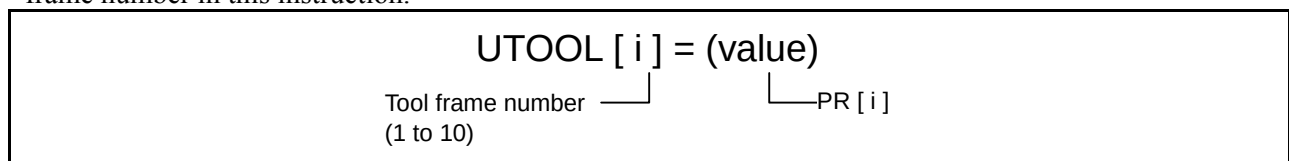


Fig. 4.12 (a) Tool Frame Setup Instruction

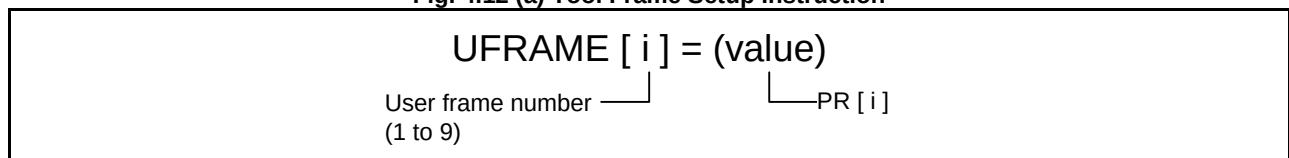


Fig. 4.12 (b) User Frame Setup Instruction

Example 1: **TOOL[1] = PR[1]**
 2: **UFRAME[3] = PR[2]**

Frame select instruction

The tool frame select instruction changes the current tool frame number.

The user frame select instruction changes the current user frame number.

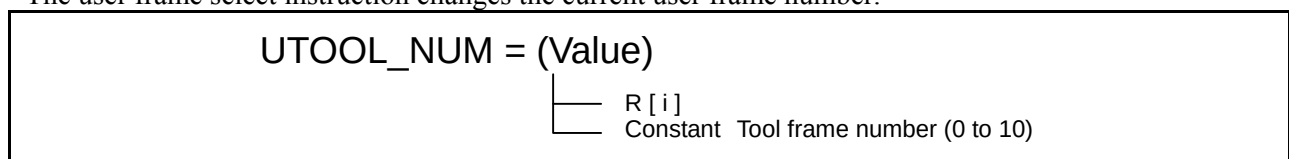


Fig. 4.12 (c) Tool Frame Select Instruction

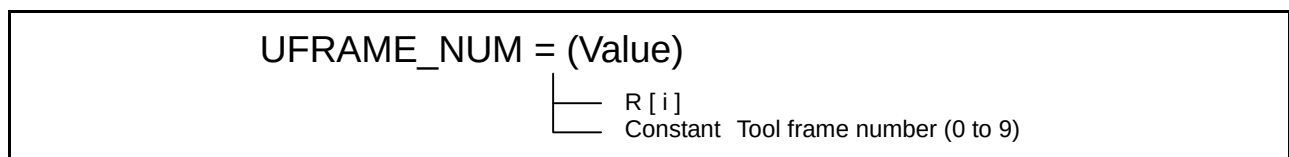


Fig. 4.12 (d) User Frame Select Instruction

Example 1: **UFRAME_NUM = 1**
 2: **J P[1] 100% FINE**
 3: **L P[2] 500mm/sec FINE**
 4: **UFRAME_NUM = 2**
 5: **L P[3] 500mm/sec FINE**
 6: **L P[4] 500mm/sec FINE**

4.13 PROGRAM CONTROL INSTRUCTIONS

The program control instructions control program execution.

Instruction		JOINT 30 %
1 Miscellaneous	5 MACRO	
2 Program control	6 SENSOR	
3 SKIP	7 Multiple control	
4 Offset/Coordinate	8 ---next page---	
PROGRAM		

- Halt instruction
- Abort instruction

4.13.1 Halt Instruction

PAUSE

The halt instruction stops program execution, causing the robot in motion to decelerate and stop:

- If an operation instruction is being executed, the program stops before the operation is completed.
- The cursor moves to the next line. When restarted, the program is executed from this line.
- If the local program timer is active, it is stopped. When the program is resumed, the local program timer is activated. The global program timer is not stopped.
- If a pulse output instruction is being executed, the program stops after that instruction has been executed.
- If an instruction other than a program call instruction is being executed, the program stops after that instruction has been executed. A program call instruction is executed when the program is restarted.

PAUSE

Fig. 4.13.1 Halt Instruction

4.13.2 Abort Instruction

ABORT

The abort instruction aborts program execution in the following way, causing the robot in motion to decelerate and stop:

- If an operation instruction is being executed, the program stops before the operation is completed.
- The cursor stops on the current line.
- When the abort instruction is executed, the execution of the program cannot be continued. Information held by a program call instruction about the main program is lost.

ABORT

Fig. 4.13.2 Abort Instruction

4.14 OTHER INSTRUCTIONS

The following miscellaneous instructions are available:

Instruction		JOINT 30 %	
1	Miscellaneous	5	MACRO
2	Program control	6	SENSOR
3	SKIP	7	Multiple control
4	Offset	8	---next page---
PROGRAM			

- RSR instruction
- User alarm instruction
- Timer instruction
- Override instruction
- Comment instruction
- Message instruction
- Parameter instruction
- Maximum speed instruction

4.14.1 RSR Instruction

RSR [i] = (value)

The RSR instruction alternately enables and disables the RSR function having a specified RSR number.

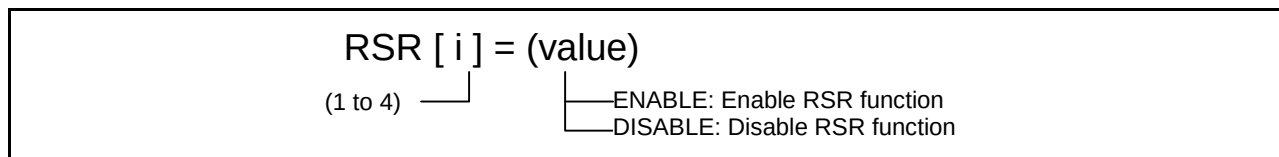


Fig. 4.14.1 RSR instruction

Example RSR[2:Workproc.2.]=ENABLE

4.14.2 User Alarm Instruction

UALM[i]

The user alarm instruction displays the alarm message corresponding to an already set user alarm number on the alarm display line. The user alarm instruction pauses the program which is on progress. A user alarm is specified on the user alarm setting screen(See Section 3.12) and this setting is registered in the system variable \$UALM_MSG . The total number of user alarms can be changed at a controlled start (See Section B.1, "START MODE").

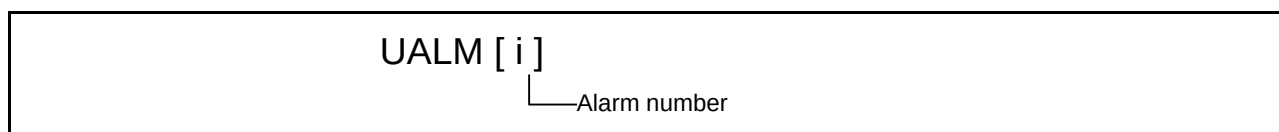


Fig. 4.14.2 User Alarm Instruction

Example 1: UALM[1] (\$UALM_MSG[1] = WORK NOT FOUND

4.14.3 Timer Instruction

timer [i] = (state)

The timer instruction starts/stops the program timer. The operating state of the program timer can be viewed on program timer screen STATUS PRGTIMER.

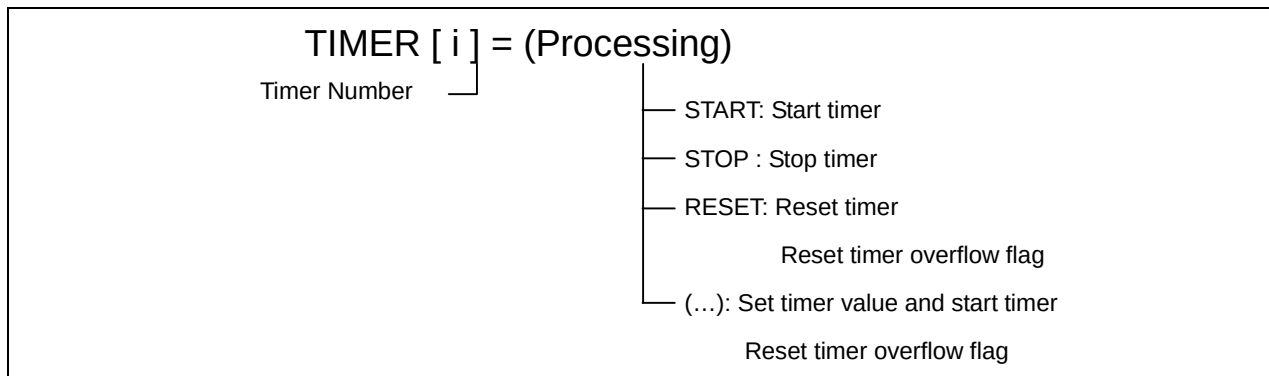


Fig. 4.14.3 Timer instruction

Example 1: `TIMER [1]=START`
`TIMER [1]=STOP`
`TIMER [1]=RESET`
`TIMER [1]=(R[1]+1)`

Mixed logic can be used for timer assignment statement (9.19 Mixed Logic).

NOTE

Timer assignment statement(`TIMER[i]=(...)`) is available only on 7DA6 series or later.

The value of the timer can be referenced in a program, using a register instruction. It is possible to determine whether the timer has overflowed by using a register instruction. The program timer overflows if it exceeds 2147483.647 seconds.

`R[1]=TIMER [1]`
`R[2]=TIMER_OVERFLOW[1]`
 0: Not over flow
 1: Over flow

4.14.4 Override Instruction

OVERWRITE = (value)%

The override instruction changes a feedrate override.

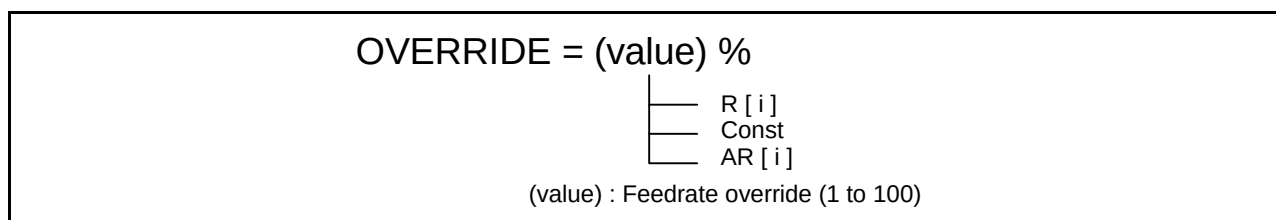


Fig. 4.14.4 Override Instruction

Example 1: `OVERWRITE = 100%`

4.14.5 Comment Instruction

!(Remark)

The comment instruction adds a comment in a program. A comment has no effect on program execution. A comment specified in a comment instruction can consist of up to 32 characters including alphanumeric characters, asterisks (*) underlines (_), and at marks (@). To add a comment, press the ENTER key.

!(Remark)

└ A comment can consist of up to 32 characters including alphanumeric characters, asterisks (*), underlines (_), and at marks (@ etc...).

Fig. 4.14.5 Comment Instruction

Example 1: !APPROACH POSITION

4.14.6 Multi-language Comment Instruction

--(Remark)

The multi-language comment instruction adds a comment in a program alike to the comment instruction. A comment has no effect on program execution. To add a comment, press the ENTER key.

It is different from 4.14.5 Comment Instruction. It is possible to input comment to each language independently. (refer 3.16 SETTING THE GENERAL ITEMS about change current language.) This function is available in 7DA4 or later.

--(Remark)

└ A comment can consist of characters including alphanumeric characters, asterisks(*),underlines(_),and atmarks(@ etc...)

Fig. 4.14.6 Multi-language comment Instruction

Example

1: --APPROACH POSITION (Current language : ENGLISH)

↓ switch to language

1: -- (Current language : JAPANESE)

In this example, comment in English is not reflected to Japanese comment. Because comment is independent to each language.

It is necessary to input comment to each language.

4.14.7 Message Instruction

MESSAGE[message statement]

The message instruction displays a specified message on the user screen. (For the user screen, see Section 7.2.) A message can consist of up to 24 characters including alphanumeric characters, asterisks (*), underlines (_), and at marks (@). To add a comment, press the ENTER key.

Message [message statement]

└ A message statement can consist of up to 24 characters including alphanumeric characters, asterisks (*), underlines (_), and at marks (@).

Fig. 4.14.7 Message Instruction

Example 1: MESSAGE[DI[1] NOT INPUT]

4.14.8 Parameter Instruction

\$(SYSTEM VARIABLE NAME) = (value)

The parameter instruction changes the value of a system variable. This instruction can be used only for a system variable containing a numeric value (constant). You can enter the parameter name after pressing the ENTER key. It is possible to enter the parameter name up to 30 characters or less without the first character, "\$".

There are two types of system variables, variable type and position type. A system variable of variable type can be assigned to a register. A system variable of position type can be assigned to a position register.

System variables of position data type are divided into three data types, cartesian (XYZWPR type), joint type (J1-J6 type), and matrix type (AONL type).

When a system variable of position data type is assigned to a position register, the data type of the position register is converted to the data type of the system variable.

If a system variable of position type is assigned to a register, or if a system variable of variable type is assigned to a position register, the following alarm is generated during execution.

INTP-240 Incomputible datatype

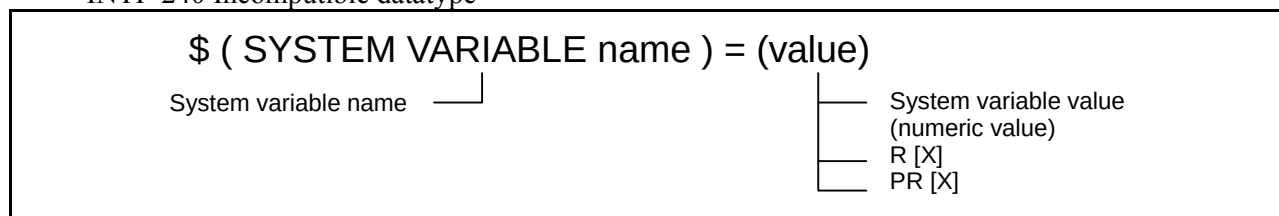


Fig. 4.14.8 (a) Parameter Instruction (Writing)

Example 1: \$SHELL_CONFIG.\$JOB_BASE = 100

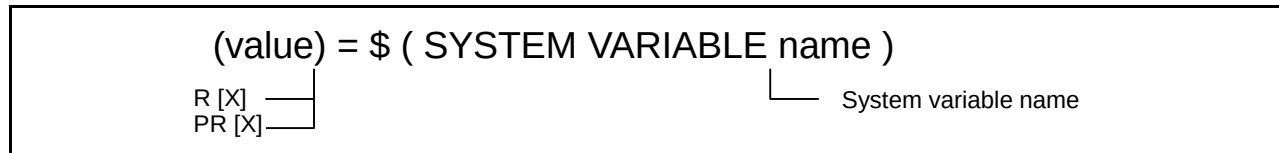


Fig. 4.14.8 (b) Parameter Instruction (Reading)

Example 1: R[1] = \$SHELL_CONFIG.\$JOB_BASE



WARNING

The operation of the robot and controller is controlled with system variables. Only a person who is aware of how changes to the system variables will affect the system should set system variables. If a person without detailed knowledge attempts to set the system variables, the robot and controller would malfunction, causing injury to personnel or damage to equipment.

Procedure 4-3 Specifying parameter instructions

Step

- 1 On the program edit screen, press function key [INST]. Select item Miscellaneous from the menu. Then, select item Parameter name from the menu.

Miscellaneous stat		JOINT 10 %
1	\$...=...	5
2	...=\$...	6
3		7
4		8
PROGRAM1		
		1/1
[End]		
Select item		
[CHOICE]		

- 2 Select item 2 "...=\$..."

Miscellaneous stat		JOINT 10 %
1	R[]	5
2	PR[]	6
3		7
4		8
PNS0001		
		1/1
1: ...=\$...		
[End]		
Select item		

- 3 Select item 1 "R[]" and enter the desired register number.

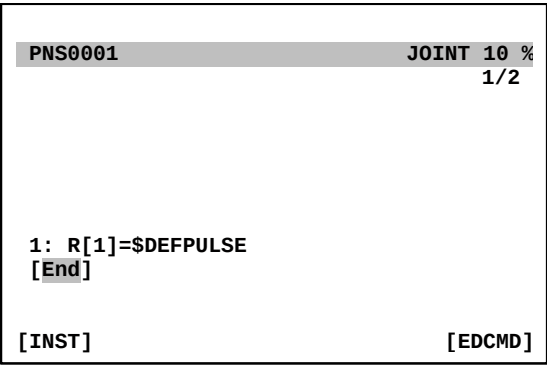
PNS0001		JOINT 10 %
		1/2
1: R[1]=\$...		
[End]		
Press ENTER		
[CHOICE]		

- 4 To display the system variable menu, press the [CHOICE] function key.
To enter a character string, press the Enter key.

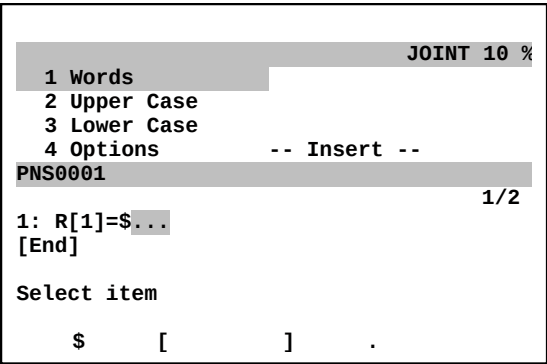
When function key CHOICE is pressed

Parameter menu		JOINT 10 %
1	DEFPULSE	5
2	WAITTMOUT	6
3	RCVTMOUT	7
4		8 --- next page ---
PROGRAM1		
		1/2
1: R[1]=\$...		
[End]		
Select item		
[CHOICE]		

5 Select item 1 "DEFPULSE."



6 When ENTER is pressed



7 Enter the desired system variable name.

4.14.9 Maximum Speed Instructions

A maximum speed instruction specifies the maximum operating speed of a program. There are two maximum speed instructions, the instruction for specifying the joint operation speed and that for specifying the path control operating speed. If a speed exceeding the speed specified with a maximum speed instruction is specified, the speed specified with the maximum speed instruction is assumed.

JOINT_MAX_SPEED[i]=(value)

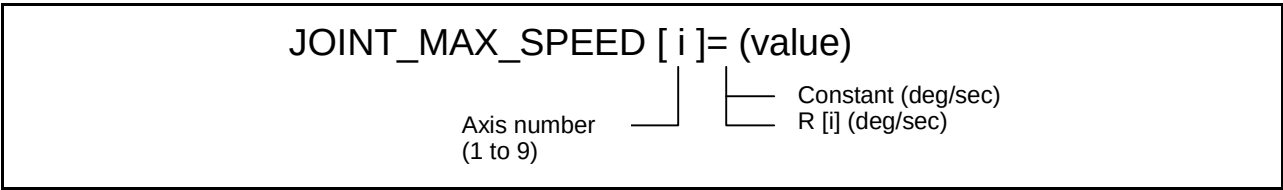


Fig. 4.14.9 (a)

Example JOINT_MAX_SPEED[3] = R[3]

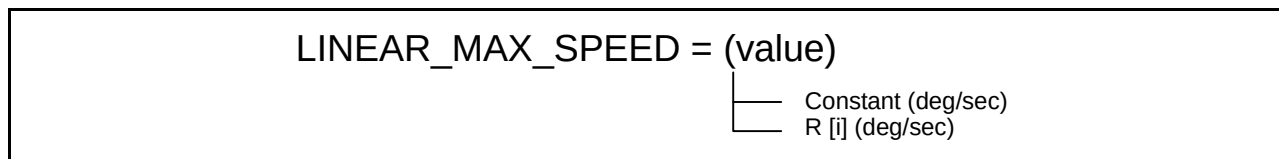
LINEAR_MAX_SPEED= (value)

Fig. 4.14.9 (b)

Example `LINEAR_MAX_SPEED = 100`

4.15 MULTIAxis CONTROL INSTRUCTIONS

Multiaxis control instructions control the execution of a multitask program.

Instruction		JOINT 30 %	
1	Miscellaneous	5	MACRO
2	Program control	6	SENSOR
3	SKIP	7	Multiple control
4	Offset/Coordinate	8	---next page---

- Program execution instruction

4.15.1 Program Execution Instruction

During the execution of a program, the program execution instruction starts the execution of another program.

- The difference from the program call instruction is that, with the program call instruction, those lines following the call instruction are executed after the called program has been executed, whereas with the program execution instruction, the program that starts the execution of another program continues concurrently.
- To synchronize programs that are being executed simultaneously, use the register instruction and the register condition wait instruction.
- If an attempt is made to execute a program for which the same motion group is specified, an alarm is generated. If this occurs, specify a different motion group.

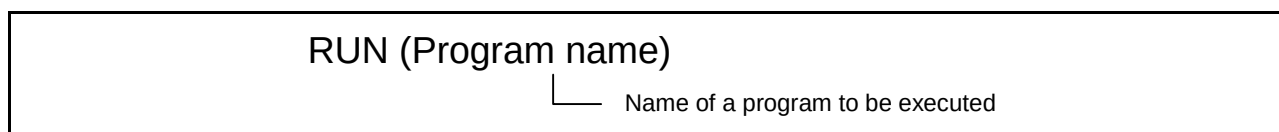


Fig. 4.15.1 Program execution instruction

Example PROG1

```
1: R[1]=0
2: RUN PROG1
3: J P[1] 100% FINE
4: J P[2] 100% FINE
5: WAIT R1[1]
```

MOTION GROUP[1,*,*,*]

PROG2

```
1: J P[3] 100% FINE
2: J P[4] 100% FINE
3: J P[5] 100% FINE
4: J P[6] 100% FINE
5: R[1]=1
```

MOTION GROUP[* ,1,*,*,*]

4.16 OPERATION GROUP INSTRUCTIONS

The operation group instructions enable the following in single-line operation instructions in a program having multiple operation groups:

- Specification of the operation format for each operation group (excluding the arc)
- Specification of the feedrate for each operation group
- Specification of the positioning format for each operation group

This allows each operation group to operate asynchronously.

These instructions can be specified and executed only when the multitask option is supported.

Instruction		JOINT 30 %
1 Miscellaneous	5 MACRO	
2 Program control	6 Independent GP	
3 Skip	7 Simultaneous GP	
4 Offset/Frames		

- Asynchronous operation group instruction
- Synchronous operation group instruction

With ordinary operation instructions for which these operation group instructions are not specified, all operation groups are executed with the same operation format, feedrate, and positioning format, and are synchronized with the additional motion instructions. The operation group having the longest travel time is that with which the other operation groups are synchronized.

4.16.1 Asynchronous Operation Group Instruction

The asynchronous operation group instruction controls operation groups asynchronously, with the operation formats, feedrates, and positioning formats specified separately for the individual operation groups.

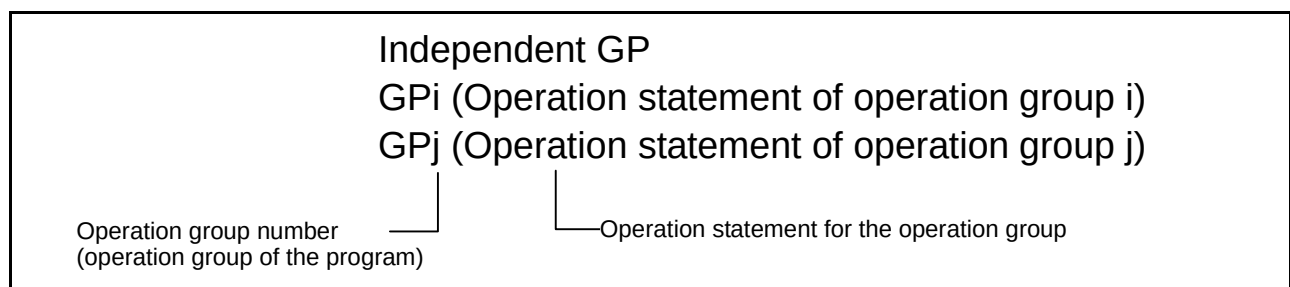


Fig. 4.16.1 Asynchronous operation group instruction

4.16.2 Synchronous Operation Group Instruction

The synchronous operation group instruction controls operation groups synchronously, with the operation formats specified separately for the individual operation groups.

- As with ordinary operation instructions, the operation group having the longest travel time is that with which the other operation groups are synchronized. Thus, the feedrate is not always the same as that specified in the program.
- The positioning format for an operation group with the smallest CNT value (closest to FINE) is also applied to the other operation groups.

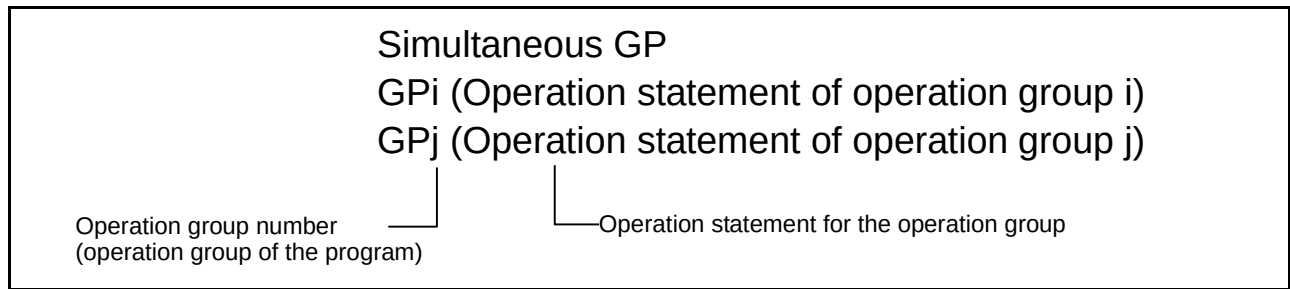


Fig. 4.16.2 Synchronous operation group instruction

5 PROGRAMMING

This chapter describes how to create and change a program for moving the robot.

Contents of this chapter

- 5.1 TIPS ON EFFECTIVE PROGRAMMING
- 5.2 TURNING ON THE POWER AND JOG FEED
- 5.3 CREATING A PROGRAM
- 5.4 CHANGING A PROGRAM
- 5.5 PROGRAM OPERATION
- 5.6 BACKGROUND EDITING
- 5.7 SINGULAR POINT CHECK FUNCTION
- 5.8 OTHER EDITING FUNCTION

Various program instructions are issued with the robot and peripherals to specify robot and hand motions. When these instructions are combined together, they create what is called a hand application program. A hand application program for instance, can:

- Move the robot to desired positions in the operating area along the specified path
- Handle workpiece
- Send output signals to the peripherals
- Receive input signals from the peripherals

Before programming, design the outline of a program. In the design, incorporate the most effective method for the robot to do the target work. This enables efficient programming and ensures that only the instructions appropriate for the purpose are used.

Instructions must be selected from menus displayed on the teach pendant during programming. To teach a target position to the robot, the robot must be moved to the target position by jog feed.

After you have finished creating the program, change the program if necessary. To change, add, delete, copy, find, or replace an instruction, select the desired item from the menu displayed on the teach pendant. This chapter describes the following:

- Tips on effective programming
- Turning on the power and jog feed
- Creating a program
- Changing a program

See Chapter 4 for the configuration of a program and the program instructions.

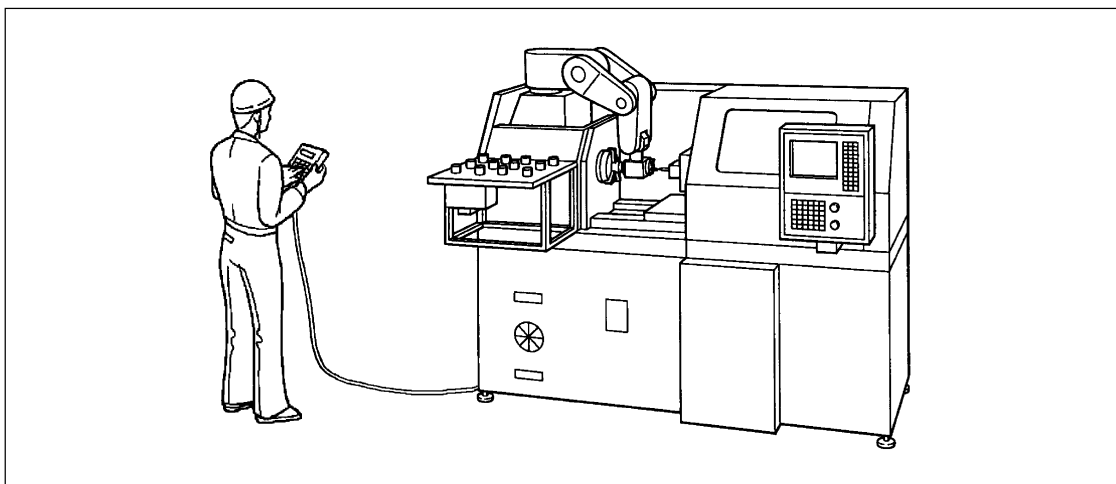


Fig. 5 Programming by Teaching

5.1 TIPS ON EFFECTIVE PROGRAMMING

This section describes tips on effective programming. The following items are explained:

- Motion instructions
- Fixed positions

NOTE

This section describes tips on programming, but does not describe tips on jog feed.

5.1.1 Motion Instructions

Refer to the following instructions when teaching motions to the robot.

Workpiece hold position = FINE positioning

Use FINE positioning for all workpiece hold positions. The robot stops exactly at the workpiece hold position. When CNT positioning is used (explained next), the robot does not stop at taught points.

Moving around workpieces = CNT positioning

Use CNT positioning for moving around workpieces. The robot continuously moves to the next target point without stopping at taught points. If the robot moves near the workpieces, adjust the path of CNT positioning.

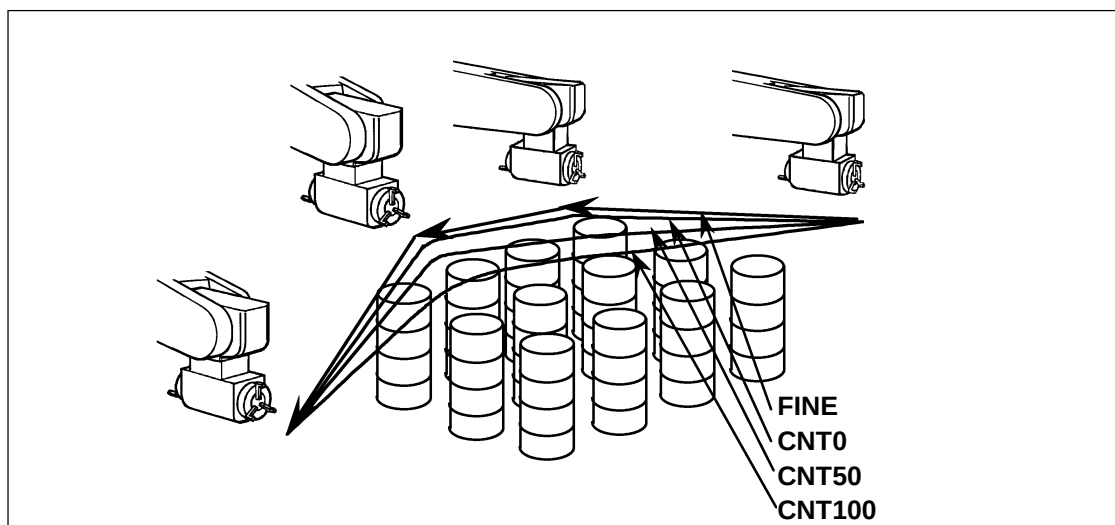


Fig. 5.1.1 (a) Adjusting the path of CNT Positioning

Fixing the attitude of the tool

Cycle time is wasted when the robot motion considerably changes the attitude of the tool. The robot moves much faster when the attitude of the tool is changed smoothly and gradually. Teach positions so that the attitude of the tool changes as gradually as possible with respect to the robot.

When the attitude of the tool must be changed considerably, teach one large motion by dividing it into several small motions. Namely, teach positions so that the attitude of the tool changes gradually.

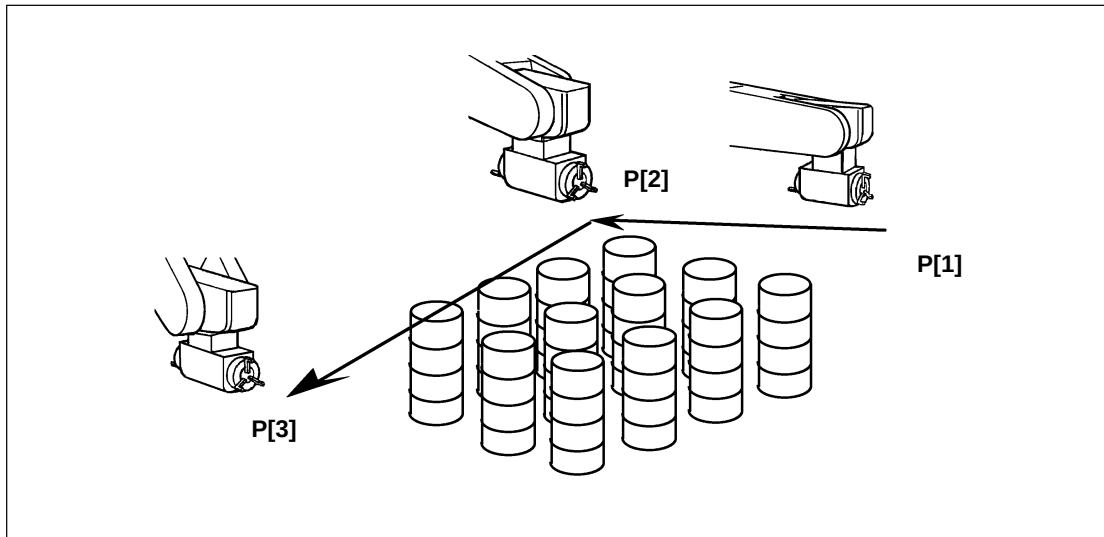


Fig. 5.1.1 (b) Teaching Positions According to the Tool Attitude

To change the attitude of the hand as smoothly as possible:

- 1 Teach the first position of the work so that the robot has a normal attitude.
- 2 Move the robot to the last position of the work by jog feed. Then check that the robot has a normal attitude.
- 3 Teach the last position.
- 4 In accordance with the work, teach a position between the first and last positions.
- 5 Select a Cartesian coordinate system (World, user or jog coordinate system) and move the robot to the first position by jog feed.
- 6 Select the Cartesian coordinate system, move the robot toward the last position by jog feed, then stop the robot at the next position to be taught.
- 7 Correct the taught position so that the robot has a normal attitude.



WARNING

If the J5 axis passes singular points (near 0 degrees) when the robot is operated by setting the move type to linear, the wrist joint motion of the additional motion instruction must be used for these points, or the move type must be changed from linear to joint.

- 8 Repeat steps 6 and 7 for all the remaining positions to be taught between the first and last positions.

5.1.2 Predefined Position

The predefined position is the position that is referenced many times in a program. The predefined positions that are used often are the pounce position and the home(perch) position. You should define these positions to program efficiently or delete cycle time.

Pounce position

The pounce position is the reference position for all work. This is the safe position away from the motion area of the machine tool and peripheral device.

Home(perch) position

The home position, or perch position, is a safety position away from the machine tool and the workpiece transfer area. The reference position digital output signal is turned on when the robot is at this position. (See Section 3.10, "SETTING A REFERENCE POSITION")

NOTE

HOME is a peripheral device I/O input signal, and does not represent a home position. A reference position is one of the home positions, but there is no utility used to move the robot to the reference position.

Other predefined position

The pounce position, reference position, or any other position can be defined as a predetermined position. Specify those positions that are frequently used in a program as predetermined positions.

When using the fixed position, use position registers (See Section 7.4) and macro instructions (See Section 9.1).



CAUTION

If teaching is made by joint coordinates, changing the user coordinate system does not affect the position variables and position registers. However, note that both position variables and registers are affected by the user coordinate systems when the robot is taught in the Cartesian format.

NOTE

To move the robot to the same spatial position when the position register is shared by two programs, the two programs must have the same tool and user coordinate system.

5.2 TURNING ON THE POWER AND JOG FEED

5.2.1 Turning On the Power and Turning Off the Power

Turning on the power starts up the robot system. Turning on the power normally executes internal processing called a cold start or hot start, then the system is started up. The special operation is necessary to perform processing with a control or initial start. (See Section B.1, "START MODE")



CAUTION

Some systems require inspection before the robot is turned on. For the sake of safety, the system should be checked before the robot is turned on.

Hot start

You can select hot start if the hot start is setup when you start the robot system. The hot start is the function that save the condition of the system just before power off and revives it after the next power on. (See Section 3.15, "SYSTEM CONFIG MENU")

- If the hot start is set to disable (\$SEMIPOWERFL=FALSE), the system starts up with the cold start. In cold start mode, the system software of the controller is initialized during starting. When you change the setting of the system such as I/O configuration, you should start up in cold start mode.

- If the hot start is set to be effective(\$SEMIPOWERFL=TRUE), the system starts up in hot start mode. In hot start mode, the system software of the controller is not initialized during starting up.

HOT START done signal

You can set that the digital output signal(DO) is turned on when the hot start is finished. This function is set with the system configuration screen [6 SYSTEM.Config].(See Section 3.15, "SYSTEM CONFIG MENU")

Automatic start program

An automatic start program can be specified. The program is automatically started when the power is turned on. If override and parameter instructions are specified in the program to be started, the system can be customized when the power is turned on.

- Set a program to be automatically started at power on when hot start is disabled to the item "Autoexec program for Cold start" in the system configuration menu. If this item is not set, the program is not started.
- Set a program to be automatically started at power on when hot start is enabled to the item "Autoexec program for Hot start" in the system configuration menu. If this item is not set, the program is not started.

The automatic start program cannot operate the robot. The automatic start program is used to set up the system or initialize the state of I/O, etc. (See Section 3.15, "SYSTEM CONFIG MENU")

Program selection after power on

The condition of the program selection after the power on is the following:

- When hot start is disable, it depends on the setting of the item "Restore selected program" in the system configuration menu.
 - TRUE : The program which had been selected at the power off is selected as it is.
 - FALSE : No program is selected.
- When hot start is effective, a program which had been selected at power off is selected as it is.

System condition

The table below lists settings in different start modes.

Table 5.2.1 System Statuses in Different Start Modes

	Hot start	
	Effective	Disable (default setting)
Contents of register	A	A
Override	A	C [10%]
Selection program	A	B (NOTE 3)
Execution line	A	C [First line]
I/O status	A (NOTE 1)	C [All off]
TP screen	B (NOTE 2)	C [Hint screen]

A: All values that are current at power-down are saved and restored at power-up.

B: Only some of the values that are current at power-down are saved.

C: The values that are current at power-down are not saved. At power-up, the default values are set.

NOTE

- 1 Generally, the status existing at power-down is restored, but digital output (DO), being performed by a pulse instruction at power-down, is turned off. To restore the I/O status, specify the desired restoration status in [6 SYSTEM Config] (see Section 3.15, "SYSTEM CONFIG MENU").
Even if hot start is enabled, none of the output signals are resumed, but all output signals are turned off in the following cases:
 - The I/O allocation was changed before power-off.
 - The fuse of the I/O device blew, or the power to the I/O device was turned off.
 - The I/O device configuration was changed.
- 2 The screen type selected at power-down is restored, but the page, cursor, and other screen statuses are not restored. Instead, the screen is restored using the same page, cursor, and other screen statuses assumed immediately after a cold start.
- 3 The name of the main program that calls the subprogram is stored.

**CAUTION**

Before the power is turned on, system statuses in the corresponding start mode described above should be checked.

5.2.2 Three-Mode Switch

The three-mode switch is a key operation switch installed on the operator's panel or operation box. This switch is used to select an optimum robot operation mode according to the robot operation conditions and use status. There are operation modes AUTO, T1, and T2. See Fig. 5.2.2.

The R-30iA Mate single chain spec. (Standard specification) does not have a 3-mode switch. When teach pendant enable switch is OFF, the AUTO mode is selected. When teach pendant enable switch is ON, the T2 mode is selected.

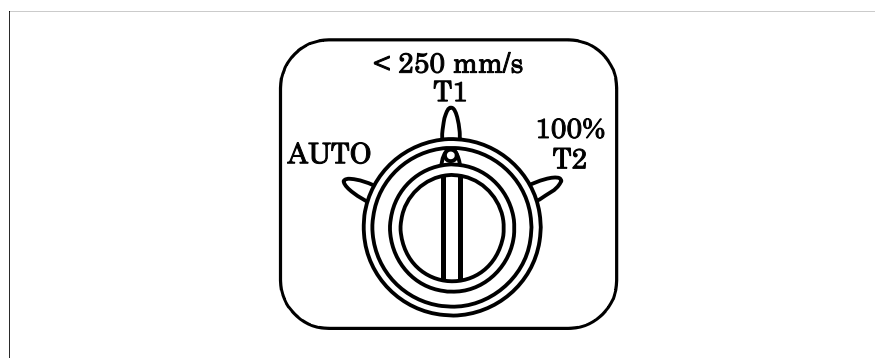


Fig. 5.2.2 Three-mode switch

When the three-mode switch is used to switch between operation modes, a message appears on the screen of the teach pendant, and the robot halts. When the key is removed from the switch, the switch setting position can be fixed.

**CAUTION**

In the case of DCS specification, if switching between T1 or T2 mode and AUTO mode is made with the deadman's switch kept held, a system error occurs. In this case, selected mode is not set until the deadman's switch is released. Release the deadman's switch, then hold the deadman's switch again.

- **Connection**

Connect the *FENCE signal to the protective fence. Make a connection in such a way that, when the protective fence is open, the signal entered to the robot is off and, when it is closed, the signal is on. The following explains the operation modes that can be selected using the three-mode switch:

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

This mode is intended for use to teach the position of operation to the robot. It can also be used to check the robot path at low speed and the program sequence.

- **Program execution**

A program can be executed only from the teach pendant.

- **Robot speed at jogging**

- The speeds at the tool tip and flange are both limited not to exceed 250 mm/sec.

- **Robot speed at executing program**

- The override value can be increased to up to 100%, but the speeds at the tool tip and flange surface are limited to 250 mm/sec or slower. For example, if the taught speed is 300 mm/sec, the speeds at the tool tip and flange surface are limited to 250 mm/sec. If the taught speed is 200 mm/sec, they are not limited.

Even when the taught speed is 250 mm/sec or below, the speed on the flange surface may exceed 250 mm/sec in a portion (for example, a corner) where the posture of the tool changes. In this case, the actual operation speed is limited.

The warning message MOTN - 231 T1 speed limit (G:i) appears only if the operation speed is limited and the taught speed is 250 mm/sec or below.

- Speed limitation is performed based on the taught speed with an override value of 100%. Therefore, if the taught speed is, for example, 2000 mm/sec, the operation speed is limited to 250 mm/sec for an override value of 100%. However, the operation speed can be decreased further, for example, to 125 mm/sec by lowering the override value to 50%.

- **Protective fence**

If you want to work with the protective fence kept open, it is necessary to set the three-mode switch to T1 or T2 before starting operating the robot.

- It is possible to operate the robot only when the teach pendant is enabled and the deadman switch is pressed (gripped).
- Disabling the teach pendant puts the robot in alarm condition, so the robot cannot run.
- When the teach pendant is enabled, but the deadman switch is not pressed, the robot is in alarm condition, so it cannot run.



CAUTION

When checking the program you created, be sure to follow the safety manual.

- **Fixing operation mode**

When the switch is set in the T1 mode position, the operation mode can be fixed to T1 mode by removing the key.

- **Troubleshooting**

- When the switch is set in the T1 mode position, turning off the teach pendant enable switch stops the robot and causes an error message to appear. To release the error, set the teach pendant enable switch to on, then press the RESET key.

T2 (100%): Test mode 2

The T2 mode is intended for use to make a final check of the program you created. In the T1 mode, it is impossible to verify the robot's actual tool path and cycle time because the operation speed is limited. In the T2 mode, it is possible to verify them by running the robot at the production speed because there is basically no speed limitation.

- Program execution

A program can be executed only from the teach pendant.

- Robot speed at jogging

- The speeds at the tool tip and flange are both limited not to exceed 250 mm/sec.

- Robot speed at executing program

- The override value can be increased to up to 100%. There is no special speed limitation.
- When the switch is switched to T2 mode, the default override is limited to such a value that the speeds at the tool tip and flange are both equal to or less than 250mm/sec. The override can be changed with appropriate means such as the override key.

- Protective fence

If you want to work with the protective fence kept open, it is necessary to set the three-mode switch to T1 or T2 before starting operating the robot.

- It is possible to operate the robot only when the teach pendant is enabled and the deadman switch is pressed (gripped).
- Disabling the teach pendant puts the robot in alarm condition, so the robot cannot run.
- When the teach pendant is enabled, but the deadman switch is not pressed, the robot is in alarm condition, so it cannot run.



CAUTION

When checking the program you created, be sure to follow the safety manual.

- Fixing operation mode

When the switch is set in the T2 mode position, the operation mode can be fixed to T2 mode by removing the key.

- Troubleshooting

- When the switch is set in the T2 mode position, turning off the teach pendant enable switch stops the robot and causes an error message to appear. To release the error, set the teach pendant enable switch to on, then press the RESET key.

AUTO: Auto mode

The AUTO mode is intended for use at production.

- Program execution

A program can be executed from external devices and operator's panel. Program execution from the teach pendant is impossible if the switch is set in the AUTO mode position.

- Robot speed

The robot can be operated at a maximum speed.

- Robot speed at jogging

Jogging operation is not possible.

- **Robot speed at executing program**
 - The robot can be operated at a maximum speed.
- **Protective fence**

Close the protective fence. When the protective fence is opened during program execution, the robot stops immediately (Please refer to "STOP TYPE OF ROBOT" in SAFETY for detail of stop type).
- **Fixing operation mode**

When the switch is set in the AUTO mode position, the operation mode can be fixed to AUTO mode by removing the key.
- **Troubleshooting**
 - When the switch is set in the AUTO mode position, turning on the teach pendant enable switch stops the robot and causes an error message to appear. To release the error, set the teach pendant enable switch to off, then press the RESET key.
- **Three-mode switch and program operation**

The following table lists the relationships among the three-mode switch setting, protective fence status (*FENCE signal), teach pendant (TP) enabled/disabled, deadman switch setting and program-specified robot operation speed.

Relationships between three-mode switch settings and program operations [DCS spec.(RIA, CE spec.)]

Three-Mode switch	Protective fence(*1)	TP enabled/disabled	TP deadman	Robot status	Units that can be started	Program-specified operation speed
AUTO	Open	Enabled	Gripped	Servo off (fence open)		
			Released	Servo off (deadman, fence open)		
		Disabled	Gripped	Servo off (fence open)		
			Released	Servo off (fence open)		
	Closed	Enabled	Gripped	Alarm and stop (AUTO and TP enable)		
			Released	Alarm and stop (deadman)		
		Disabled	Gripped	Operable	External start(*2)	Programmed speed
			Released	Operable	External start(*2)	Programmed speed
T1	Open	Enabled	Gripped	Operable	TP only	T1 speed
			Released	Servo off (deadman)		
		Disabled	Gripped	Servo off (T1/T2 and TP disabled)		
			Released	Servo off (T1/T2 and TP disabled)		
	Closed	Enabled	Gripped	Operable	TP only	T1 speed
			Released	Servo off (deadman)		
		Disabled	Gripped	Servo off (T1/T2 and TP disabled)		
			Released	Servo off (T1/T2 and TP disabled)		
T2	Open	Enabled	Gripped	Operable	TP only	Programmed speed
			Released	Servo off (deadman)		
		Disabled	Gripped	Servo off (T1/T2 and TP disabled)		
			Released	Servo off (T1/T2 and TP disabled)		
	Closed	Enabled	Gripped	Operable	TP only	Programmed speed
			Released	Servo off (deadman)		

Three-Mode switch	Protective fence(*1)	TP enabled/disabled	TP deadman	Robot status	Units that can be started	Program-specified operation speed
		Disabled	Gripped	Servo off (T1/T2 and TP disabled)		
			Released	Servo off (T1/T2 and TP disabled)		

*1 Protective fence status

Open: *FENCE is off.

Closed: *FENCE is on.

*2 External speed

Remote mode: Program start on the line control panel

Local mode: Start button on the robot operation panel

Note: Please refer to "STOP TYPE OF ROBOT" in SAFETY for detail of stop type.

Program operation relationship [single chain spec.(Standard specification)]

Protective fence(*1)	TP enabled/disabled	TP deadman	Robot status	Unit that can be started	Program-specified operation speed
Open	Enabled	Gripped	Operable	TP only	Programmed speed
		Released	Servo off (deadman)		
	Disabled	Gripped	Servo off (fence open)		
		Released	Servo off (fence open)		
Closed	Enabled	Gripped	Operable	TP only	Programmed speed
		Released	Servo off (deadman)		
	Disabled	Gripped	Operable	External start(*2)	Programmed speed
		Released	Operable	External start(*2)	Programmed speed

*1 Protective fence status

Open: *FENCE is off.

Closed: *FENCE is on.

*2 External speed

Remote mode: Program start on the line control panel

Local mode: Start button on the robot operation panel

Note: Please refer to "STOP TYPE OF ROBOT" in SAFETY for detail of stop type.

5.2.3 Moving the Robot by Jog Feed

The robot moves by jog feed when the jog keys on the teach pendant are pressed. The robot must be moved to a target position when motion instructions are specified in the program.

Jog feed depends on the following two factors:

- Feedrate override: Robot motion speed (jog feedrate)
- Manual-feed coordinate system: Coordinate system for robot motion (jog feed type)

Feedrate override

A feedrate override is one of the two factors on which jog feed depends. The feedrate override is represented in percentage (%).

The current feedrate override is displayed at the upper right corner of the screen of the teach pendant.

Pressing the feedrate override key displays a pop up window in reverse video at the upper right of the screen to call the user's attention. The popup window in reverse video automatically disappears after a few seconds or when another key is pressed.

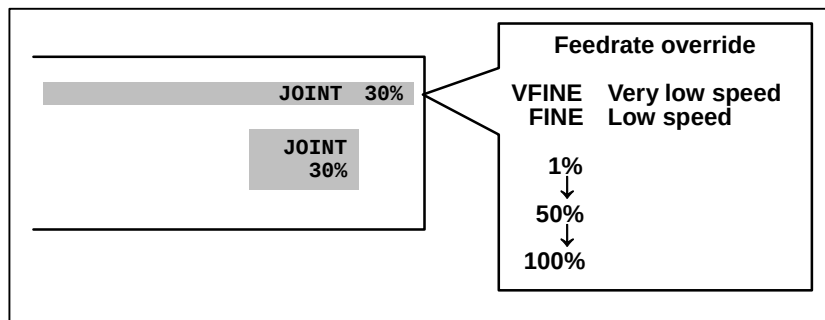


Fig. 5.2.3 (a) Screen Display for Feedrate Override

Feedrate override 100% means that the robot moves at the maximum feedrate. The step feed-rate of FINE is specified by a system variable, \$JOG_GROUP.\$FINE_DIST in linear jog.(Standard : 0.1mm). In standard setting, each axis rotates at 0.001deg per step. The step width of VFINE is one-tenth of that of FINE.

NOTE

If VFINE or FINE is used as the current speed override, the robot makes a motion of a single step at a time. To resume the robot motion, release and press the jog key.

Table 5.2.3 (a) shows the change in feedrate override when the override key is pressed.

Table 5.2.3 (a) Feedrate Override

When the override key is pressed	VFINE → FINE → 1% → 5% → 50% → 100% In 1% In 5% increments increments
When the override key is pressed while pressing the SHIFT key (*1)	VFINE → FINE → 5% → 50% → 100%

*1 Enabled only when \$\$SHFTOV_ENB is 1

To change the feedrate override, press the override key. Whenever the override key is pressed while the shift key is pressed, the feedrate changes sequentially in the order: FINE, VFINE, 5%, 50%, and 100%. However, the feedrate is changed in this way only when system variable \$\$SHFTOV_ENB = 1.

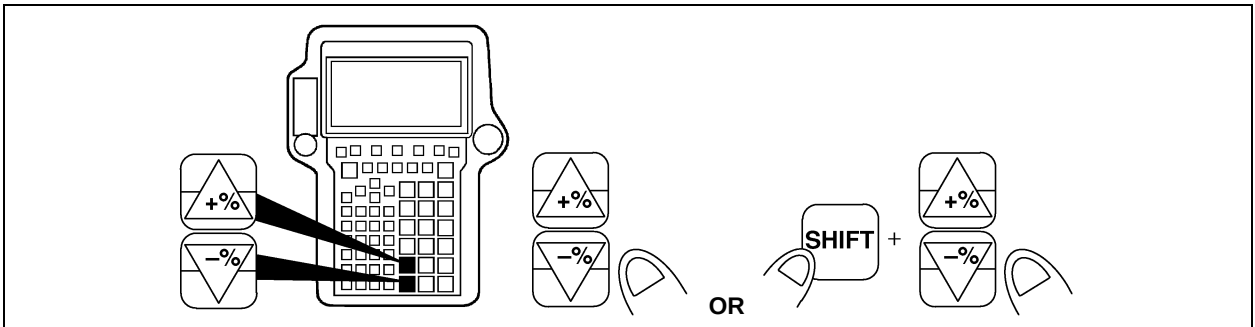


Fig. 5.2.3 (b) Override Keys

A feedrate override must be determined according to the condition of the machining cell, type of the robot motion, or the skill of the operator. Therefore, an inexperienced robot operator should use a low feedrate override.

NOTE

When the override key is pressed, a window indicating the manual feed coordinate system and speed override appears on the screen in reverse video. Pressing the override key again enables you to change the override value. This window is automatically closed if the override keys are not pressed for a while.

Jog feedrate

A jog feedrate is a speed at which the robot moves during jog feed. The jog feedrate is obtained by the following expression: If the following value exceeds the speed limit 250 mm/sec for the T1 or T2 mode described above, the operation speed is clamped at the one described earlier.

Jog feedrate(joint feed) (deg/sec, mm/sec) =	
$\text{Maximum joint feedrate} \times \frac{\text{Each axis jog override}}{100} \times \frac{\text{Feedrate override}}{100}$	
Jog feedrate(linear feed) (mm/sec) =	
$\text{Maximum linear feedrate} \times \frac{\text{Jog override}}{100} \times \frac{\text{Feedrate override}}{100}$	
Jog feedrate(Circular feed) (deg/sec) =	
$\text{Maximum circular feedrate} \times \frac{\text{Orientation Jog override}}{100} \times \frac{\text{Feedrate override}}{100}$	
Each axis jog override	\$SCR_GRP[g]. \$JOGLIM_JNT[i] (%)
Jog override	\$SCR. \$JOGLIM (%)
Orientation jog override	\$SCR. \$JOGLIMROT (%)
Maximum joint feedrate	\$PARAM_GROUP[g]. \$JNTVELLIM[i] (deg/sec, mm/sec)
Maximum linear feedrate	\$PARAM_GROUP[g]. \$SPEEDLIM (mm/sec)
Maximum circular feedrate	\$PARAM_GROUP[g]. \$ROTSPEEDLIM (deg/sec)
Note: g is group number. i is axis number.	

Fig. 5.2.3 (c) Jog Feedrate

- **Manual-feed coordinate systems (Jog type)**

Manual-feed coordinate systems determine how the robot moves during jog feed. The manual-feed coordinate systems are classified into three types:

- **Joint jog (JOINT)**

During joint jog, the robot moves independently around each axis according to each joint coordinate system. See Section 3.9 for the joint coordinate systems.

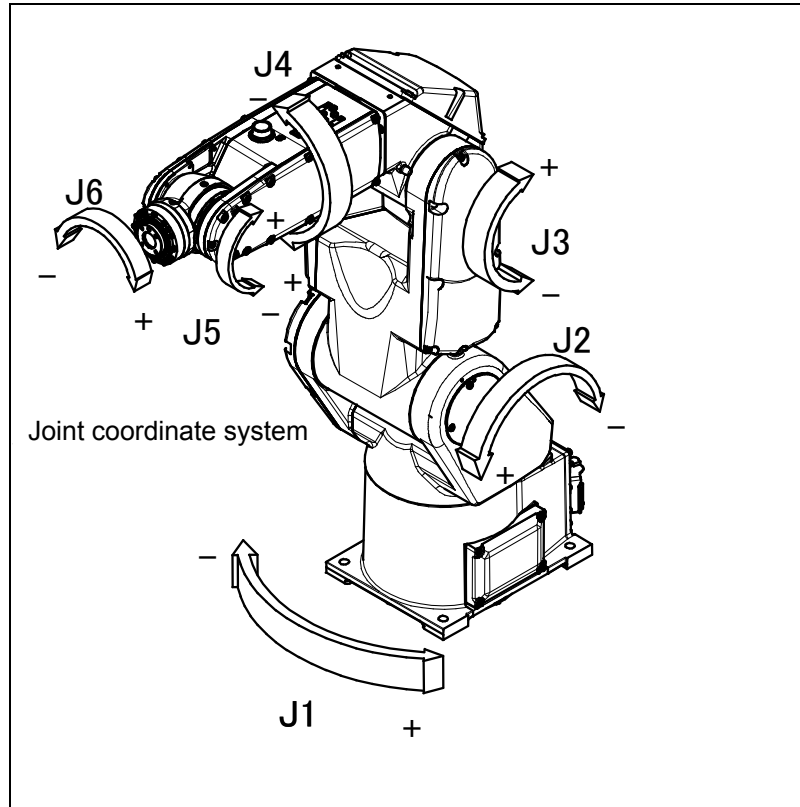


Fig. 5.2.3 (d) Joint Jog

- Cartesian jog (XYZ)

During Cartesian jog, the tool center point of the robot moves along the X-, Y-, and Z-axes of the user or jog coordinate systems. You can cause the robot to rotate the tool around x-,y-,and z-axis of the user frame or jog frame.(See Subsection 3.9.2, "Setting a User Coordinate System", and Subsection 3.9.3, "Setting a Jog Coordinate System")

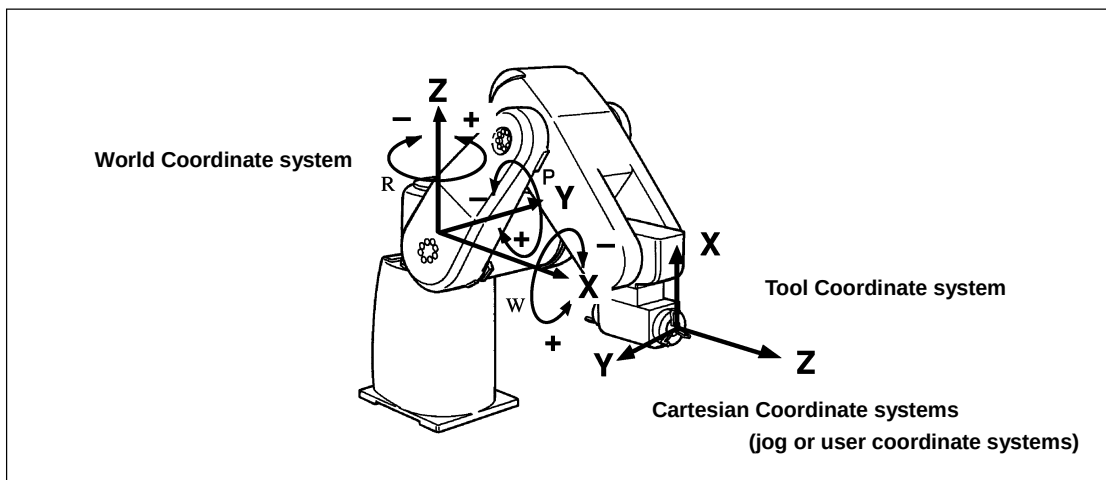


Fig. 5.2.3 (e) Cartesian Jog

Tool jog (TOOL)

During tool jog, the tool center point (TCP) moves along the X-, Y-, and Z-axes of the tool coordinate system defined for the wrist of the robot. You can cause the robot to rotate the tool around x-,y-,and z-axis of the tool frame.(See Subsection 3.9.1, "Setting a Tool Coordinate")

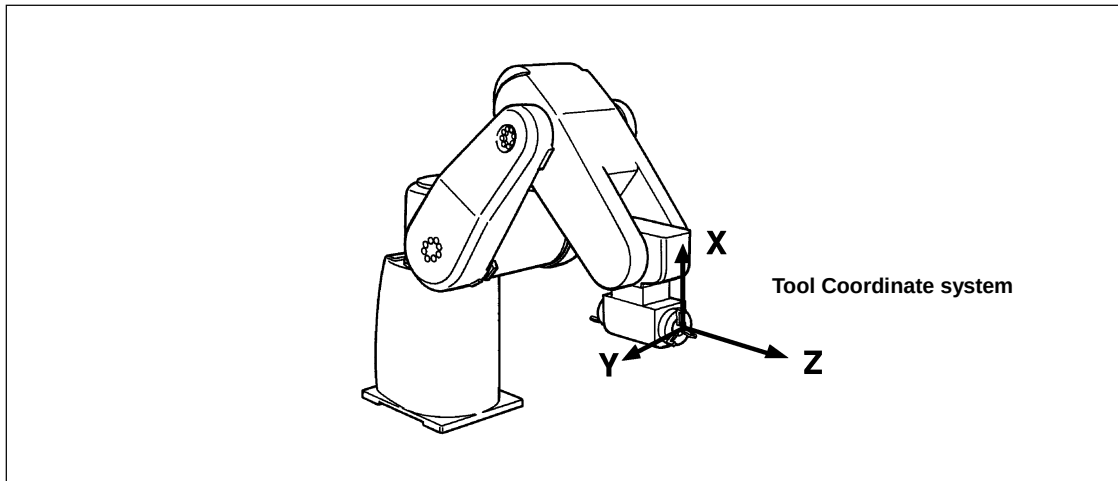


Fig. 5.2.3 (f) Tool Jog

Selecting a manual-feed coordinate system

The current manual-feed coordinate system is displayed at the upper right corner of the screen of the teach pendant.

Pressing the COORD key displays a popup menu in reverse video at the upper right of the screen to call the user's attention. The popup menu in reverse video automatically disappears after a few seconds or when another key is pressed.

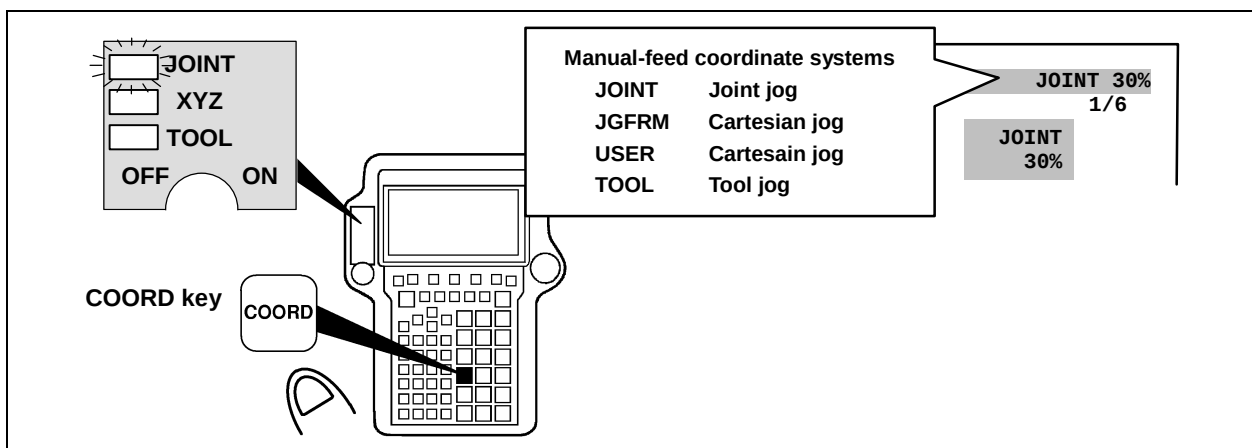


Fig. 5.2.3 (g) Screen Display for Manual-Feed Coordinate Systems

Whenever the COORD key on the teach pendant is pressed, the selected manual-feed coordinate system change cyclically. When a manual-feed coordinate system changes sequentially in the order shown in Table 5.2.3(b), LED correspondig to the selected coordinate system lights.

Table 5.2.3 (b) Jog type Selection Sequence

Screen display	JOINT → JGFRM → TOOL → USER → JOINT
LED state	JOINT LED on → XYZ LED on → TOOL LED on → XYZ LED on → JOINT LED on

Enabling a wrist joint feed

In wrist joint feed, the attitude of the tool is not held during linear feed (Cartesian jog feed) or circular feed (tool jog feed).

- When wrist joint feed is disabled, the attitude of the tool is held during jog feed. (Standard setting)
- When wrist joint feed is enabled, the attitude of the tool is not held during jog feed. In this case, [W/] is displayed on the screen.

- In linear feed (linear motion along the axes of the Cartesian coordinate system), the tool center point moves linearly while the wrist joint is fixed.
- In rotational feed (attitude rotation about the wrist axis), the wrist axis is moved by joint motion while the tool center position is held.

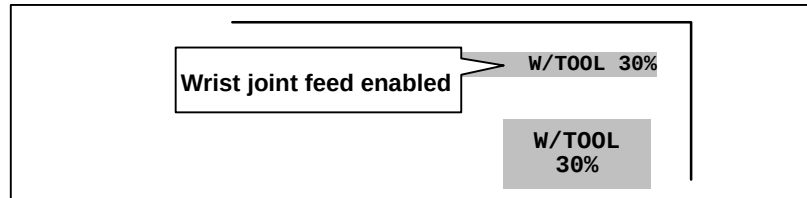


Fig. 5.2.3 (h) Indication that Wrist Joint Feed Is Enabled

NOTE

When the motion instruction for linear or circular motion under path control is executed, wrist joint feed has the same function as the wrist joint motion additional instruction (WRIST JOINT).

Jog feed of 5-axis robot

Since 5-axis robot cannot move under a full control of its Tool attitude, there are many attitudes which the robot cannot reach. For this reason, the jog feed of 5-axis robot has some unique characteristics.

- 1 During linear jog feed (along X-, Y-, Z-axis), 5-axis robot can exactly control its Tool attitude ONLY IF the flange surface faces in the vertical direction. If not, 5-axis robot moves in an alternative Tool attitude which is reachable. On the other hand, 5-axis robot can always exactly control its TCP position.
- 2 Rotational jog feed (around X-, Y-, Z-axis) is automatically translated into the wrist joint feed.
- 3 During Joint jog feed, you will see a message displayed and the robot temporarily stops. A message “Vertical fixture position” means that the flange surface now faces in the vertical direction. And a message “Horizontal fixture position” means that the flange surface now faces in the horizontal direction.

Switching to additional axes

In addition to the standard robot axes (usually 4 to 6 axes) in one operation group, up to three additional axes can be controlled as a subgroup.

NOTE

The user can switch to a subgroup by using the function menu or jog menu described below.

Jog menu

With the jog menu function, the following data related to jog operation can be displayed or updated easily:

- Tool, jog, or user coordinate system number currently selected
- Group number currently selected
- Subgroup selection state (robot or additional axes)

To display the jog menu, press the manual feed coordinate system key while holding down SHIFT key.

TEST	
UTILITIES Hints	TOOL 100%
	Tool (.=10) 2
	Jog 3
	User 1
	Group 2
	Robot/Ext

Table 5.2.3 (c) Operation Procedure Using the Jog Menu

Operation	Procedure
Opening the menu	Press the manual feed COORD key while holding down SHIFT key.
Closing the menu	• Press the manual feed COORD key while holding down SHIFT key. • Press PREV key. • Value modification using numerical key (See the descriptions of coordinate system number change and group switching.)
Moving the cursor	cursor key
Changing the coordinate system number	• Tool coordinate system, jog coordinate system 1 to 10 (Put the "." key to select 10.) • User coordinate system 0 to 9
Group switching (for a multi-group system only)	Numeric key (valid for existing group numbers only)
Subgroup switching (for a system with a subgroup)	After moving the cursor to the line containing Robot/Ext, switch between Robot and Ext by using the left/right cursor key. (The position of reverse video switches.)

⚠ WARNING

- 1 Be sure to remember the current coordinate system number/group number. Otherwise, in such a case, a robot may move in an unexpected direction at jog time, or a robot of an unexpected group may move, thus leading to a fatal accident.
- 2 After coordinate system number/group number switching, be sure to close the jog menu. If the jog menu is left open, the operator may change the coordinate system number or group number by touching a numeric key of the teach pendant unconsciously. In such a case, a robot may move in an unexpected direction at jog time, or a robot of an unexpected group may move, thus leading to a fatal accident.

Procedure 5-1 Moving the robot by jog feed**Condition**

- Do not enter the operating area. Do not put any obstacles within the work area.

⚠ CAUTION

Before you jog the robot be sure that all safety requirements for the work area are satisfied. Otherwise, injury or property damage could occur.

Step

- 1 Press the COORD key to display a desired manual-feed coordinate system on the teach pendant.

NOTE

The feedrate override is automatically set to 10%.

- 2 Press the override key to adjust the jog feedrate displayed on the teach pendant.
- 3 Hold the teach pendant and press the deadman switch on the back of the teach pendant. Continue pressing the deadman switch during jog feed.
- 4 Turn on the teach pendant enable switch.

NOTE

- 1 If the deadman switch is released when the teach pendant enable switch is on, an alarm occurs. To reset the alarm, press and hold down the deadman switch again, then press the RESET key on the teach pendant.
- 2 If the operator is not accustomed to the operation of the robot or is not sure about the robot motions, low feedrate overrides should be set.

**WARNING**

The robot starts its motion in the next step. If the jog feed of the robot needs to be stopped in an emergency in order to avoid danger, the operator should release the deadman's switch or press the emergency stop button.

- 5 To move the robot by jog feed, press the jog key corresponding to the desired robot motion direction while pressing the SHIFT key. When the jog key is released, the robot stops.

NOTE

When the override is FINE or VFINE, press the jog key and release it every time for each motion.

Switch to wrist joint feed

- 6 Press the FCTN key. The function menu is displayed.
- 7 Select 5, TOGGLE WRIST JOG. The mark, [W/], is displayed to show the wrist joint jog mode. To release this mode, select 5, TOGGLE WRIST JOG again.

4 TOGGLE SUB GROUP
5 TOGGLE WRIST JOG
6

FCTN

SAMPLE1 W/T00L 30%
1/6

Switch to a extended axis

- 8 Press the FCTN key. The function menu is displayed.
- 9 Select 4, TOGGLE SUB GROUP. The jog control is switched from the robot standard axes to an extended axis. The control will be returned when it is done.

3 CHANGE GROUP
4 TOGGLE SUB GROUP
5 TOGGLE WRIST JOG

FCTN

SAMPLE1 S W/T00L 30%
1/6

- 10 To terminate jog feed, turn off the teach pendant enable switch and release the deadman switch.

5.3 CREATING A PROGRAM

To create a program, use the following procedure:

- Register a program and specify program information
- Modify standard instruction (standard motion instructions)
- Teach motion instructions
- Teach various control instructions including a palletizing instruction

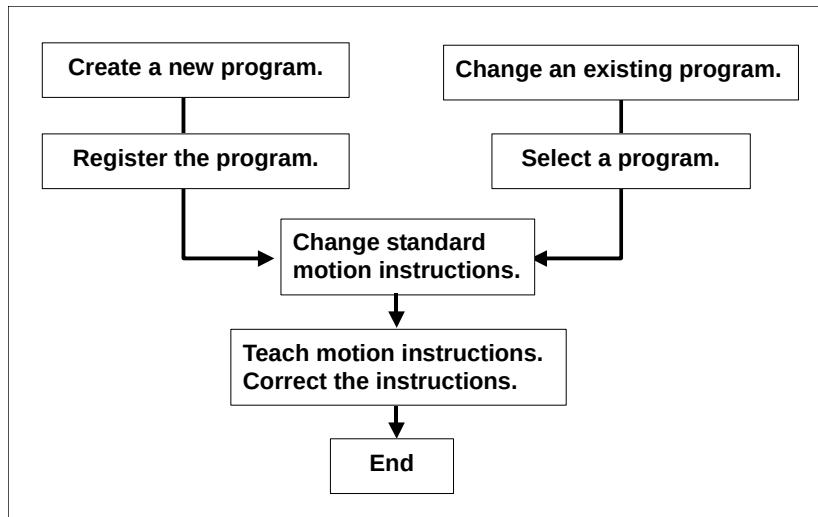


Fig. 5.3 Creating and Changing a Program

Registering a program

Create a null program with a new name.

Specifying program information

Specify the attributes of the program.

Changing standard motion instructions

Respecify the standard instructions to be used when teaching motion instructions.

Teaching motion instructions

Teach a motion instruction and a supplementary motion instruction.

Teaching control instructions

Teach control instructions including a palletizing instruction.

Use the teach pendant to create a new program and correct an existing program. To do this, the teach pendant must be enabled beforehand.

To enable the teach pendant, satisfy the following condition:

- The teach pendant enable switch must be turned on.

To prevent the program from being started by mistake, prohibit starting a program with a teach pendant while teaching. (See Fig. 2.3.1 (f), "Function menu")



5.3.1 Registering a Program

Enter a program name and register the program. A program name consists of up to eight alphanumeric characters (36 alphanumeric characters in 7DA4 or later) including symbols to discriminate program names from one another. For the program name, see Subsection 4.1.1.

Register a program on the program registration screen.



CAUTION

When a new program is made, the current program is halted.

Entering a program name

There are three methods for entering a program name:

- Words: Up to five words consisting of up to seven characters can be used as program names. Enter these reserved words, such as RSR, PNS, STYLE, JOB, and TEST, in \$PGINP_WORD[1 TO 5] in advance (See Section 3.15, "SYSTEM CONFIG MENU")
- Uppercase or lowercase alphabetic characters: Any letter of the alphabet can be specified for a program name. The alphabetic characters combined with any numeric characters and/or any symbols are used as the characters of a program name.



CAUTION

Asterisks (*) and at marks (@) should not be used in a program name.

Options

During optional settings, an overwrite or insert mode can be specified for character entry, or character string deletion.

- In the overwrite mode, entered characters are written over existing characters.
- In the insert mode, entered characters are inserted before the character pointed to by the cursor. In this case, all the characters to the right of the entered character(s) are shifted to the right. INSERT or OVRWPT is displayed on the screen.
- All the characters in the field where the cursor is positioned are deleted.

NOTE

The program name should not begin with a numeral.

Setting program information

Set the following program information items on the program information screen. See Section 4.1.

- Program name
- Subtype
 - Comments : Comments can be written in a program. Up to 16 alphanumeric characters and symbols, which can be used for a program name. In some cases, comments may not be entered.
 - Group mask : Specifies a motion group to be controlled in a program.

- Write protection : Prevents a program from being changed.
- Interruption disable : Causes the program having no motion not to be paused by an alarm with a severity of WARN, PAUSE, STOP and SERVO, the emergency stop, and HOLD. However, this setting is not applied to the alarm that is generated by the program. In this case the program is stopped.

Procedure 5-2 Registering a program

Condition

- The teach pendant must be enabled.

Step

- 1 Press the MENUS key to display the screen menu.
- 2 Select SELECT.

Alternatively, the following program list screen can also be displayed by pressing the SELECT key.

Select		JOINT 30%
61276 bytes free		1/4
No.	Program name	Comment
1	SAMPLE1	[SAMPLE PROGRAM1]
2	SAMPLE2	[SAMPLE PROGRAM2]
3	PROG001	[PROGRAM001]
4	PROG002	[PROGRAM002]
[TYPE] CREATE DELETE MONITOR [ATTR] >		
COPY DETAIL LOAD SAVE AS PRINT >		

- 3 Press the F2 [CREATE] key. The program registration screen is displayed.

1 Words		JOINT 30%
2 Upper Case		
3 Lower Case		
4 Options		--Insert--
Select		
--- Create Teach Pendant Program ---		
Program Name		
█		
-- End --		
Enter program name		
RSR	PNS	STYLE JOB TEST

- 4 Select a method for entering a program name (words or alphabetic characters) using the cursor keys.

```

                                JOINT 30%
1 Words
2 Upper Case
3 Lower Case
4 Options      --Insert--
Select
--- Create Teach Pendant Program ---
Program Name:
|
-- End --

Enter program name
ABCDEF GHIJKL MNOPQR STUVWX YZ_@* . >

```

- 5 Enter a program name by pressing the function keys corresponding to the characters in the program name. The function key menu displayed depends on the method selected in step 4. With alphabetic character entry, for instance, press the function key corresponding to a desired character repeatedly until the character is displayed in the program name field; that is, if you want to enter S, press the F4 function key one times. Press the NEXT key to move the cursor to the right one character. Repeat this procedure until the program name is completely entered.

MNOPQR STUVWX YZ_@* . >

F4

```

Select                                JOINT 30%
                                1/3
--- Create Teach Pendant Program ---
Program name:
S
Enter program name
ABCDEF GHIJKL MNOPQR STUVWX YZ_@* . >

```

NOTE

When creating a program using RSR or PNS for automatic operation, follow the rule below. Otherwise, the program does not run.

- A RSR program must be written as RSRnnnn, where nnnn represents a four-digit number. An example is RSR0001.
- A PNS program must be written as PNSnnnn, where nnnn represents a four-digit number. An example is PNS0001.

- 6 After entering a program name, press the ENTER key.

```

Select
---Create Teach Pendant
Program name:
SAMPLE3|

```

ENTER

```

Select                                JOINT 30%
--- Create Teach Pendant Program ---
Program name:
SAMPLE3
-- End --

Select function
DETAIL      EDIT

```

- 7 To edit the registered program, press the F3 (EDIT) or ENTER key. The program edit screen for the registered program is displayed.

Select function DETAIL EDIT F3	<table border="1"> <tr> <td>SAMPLE3</td> <td>JOINT 30%</td> </tr> <tr> <td></td> <td>1/1</td> </tr> <tr> <td>[END]</td> <td></td> </tr> <tr> <td>POINT</td> <td>TOUCHUP></td> </tr> </table>	SAMPLE3	JOINT 30%		1/1	[END]		POINT	TOUCHUP>
SAMPLE3	JOINT 30%								
	1/1								
[END]									
POINT	TOUCHUP>								

- 8 To enter program information, press the F2 (DETAIL) key (or the ENTER key). The program information screen is displayed.

Select function DETAIL EDIT F2	<table border="1"> <tr> <td colspan="2">Program detail</td> <td>JOINT 30%</td> </tr> <tr> <td colspan="2"></td> <td>1/7</td> </tr> <tr> <td>Creation Date:</td> <td colspan="2">10-MAR-1998</td> </tr> <tr> <td>Modification Data:</td> <td colspan="2">11-MAR-1998</td> </tr> <tr> <td>Copy Source:</td> <td colspan="2"></td> </tr> <tr> <td>Positions: FALSE</td> <td>Size:</td> <td>312 Bytes</td> </tr> <tr> <td colspan="3">Program name:</td> </tr> <tr> <td>1</td> <td colspan="2">SAMPLE3</td> </tr> <tr> <td>2</td> <td>Sub Type:</td> <td>[None]</td> </tr> <tr> <td>3</td> <td>Comment:</td> <td>[SAMPLE PROGRAM 3]</td> </tr> <tr> <td>4</td> <td>Group Mask:</td> <td>[1, *, *, *, *, *, *]</td> </tr> <tr> <td>5</td> <td>Write protect:</td> <td>[OFF]</td> </tr> <tr> <td>6</td> <td>Ignore pause:</td> <td>[OFF]</td> </tr> <tr> <td>7</td> <td>Stack size:</td> <td>[300]</td> </tr> <tr> <td>END</td> <td>PREV</td> <td>NEXT</td> </tr> </table>	Program detail		JOINT 30%			1/7	Creation Date:	10-MAR-1998		Modification Data:	11-MAR-1998		Copy Source:			Positions: FALSE	Size:	312 Bytes	Program name:			1	SAMPLE3		2	Sub Type:	[None]	3	Comment:	[SAMPLE PROGRAM 3]	4	Group Mask:	[1, *, *, *, *, *, *]	5	Write protect:	[OFF]	6	Ignore pause:	[OFF]	7	Stack size:	[300]	END	PREV	NEXT
Program detail		JOINT 30%																																												
		1/7																																												
Creation Date:	10-MAR-1998																																													
Modification Data:	11-MAR-1998																																													
Copy Source:																																														
Positions: FALSE	Size:	312 Bytes																																												
Program name:																																														
1	SAMPLE3																																													
2	Sub Type:	[None]																																												
3	Comment:	[SAMPLE PROGRAM 3]																																												
4	Group Mask:	[1, *, *, *, *, *, *]																																												
5	Write protect:	[OFF]																																												
6	Ignore pause:	[OFF]																																												
7	Stack size:	[300]																																												
END	PREV	NEXT																																												

- 9 Specify the following program information items:
- To change a program name, move the cursor to the setting field, change the program name, then press the ENTER key.
 - To change a subtype (see Subsection 4.1.3), press the F4 [CHOICE] key to display a subtype menu. Then, select None, Job, Process, or Macro. JOB or PROCESS can be selected only when system variable \$JOBPROC_ENB is set to 1.
 - To enter comments, move the cursor to the setting field, enter the comments, then press the ENTER key (see Subsection 4.1.2).
 - To specify a group mask, move the cursor to the setting field and select 1, *. The specified motion group is controlled (see Subsection 4.1.4). For safety, specify (*, *, *, *, *, *, *) for programs which do not contain any motion instructions.

**CAUTION**

You cannot change the motion group of a program that contain operation instructions cannot be changed.

NOTE

If the system used does not have the multi-group setting, only either of the following settings is allowed: The first group is set as 1; An asterisk (*) indicating no group is set.

- To specify write protection, move the cursor to the setting field and select ON or OFF (see Subsection 4.1.5).
- To specify interruption disable, move the cursor to the setting field and press the function key (ON or OFF) (see Subsection 4.1.6). Select ON for programs not to be halted when an alarm occurs such as macro instructions or automatic start programs.

NOTE

To return to the list screen, press the PREV key repeatedly until the list screen is displayed.

Program detail

JOINT 30%

1/7

Creation Date: 10-MAR-1998

Modification Data: 11-MAR-1998

Copy Source:

Positions: FALSE Size: 312 Bytes

Program name:

1 SAMPLE3

2 Sub Type: [None]

3 Comment: [SAMPLE PROGRAM 3]

4 Group Mask: [1,*,*,*,*,*,*]

5 Write protect: [OFF]

6 Ignore pause: [OFF]

7 Stack size: [300]

END PREV NEXT

- 10 After entering the program information items, press the F1 (END) key. The program edit screen for the registered program is displayed.

END PREV NEXT

F1

SAMPLE3

JOINT 30%

1/1

[END]

POINT

TOUCHUP>

5.3.2 Changing a Standard Motion Instruction

For specification of a move statement, many items including move type, move speed, and positioning type need to be set. For convenience, the user can register frequently used move instructions as standard move statements.

To modify a standard operation statement, first press the F1 key. A list of standard operation statements appears. Press the F key again. The screen for editing the standard operation statements appears.

POINT

TOUCHUP>

- Press F1 POINT to list the standard operation statements.

Procedure 5-3 Changing a standard motion instruction

Condition

- The program edit screen must be selected.
- The teach pendant must be enabled.

SAMPLE3	JOINT 30%
	1/1
[END]	
POINT	TOUCHUP>

Step

- 1 Press the F1 POINT key. The standard motion instruction menu is displayed.

POINT	
F1	

Default Motion	JOINT 30%
1 J P[] 100% FINE	
2 J P[] 100% FINE	
3 L P[] 1000cm/min CNT50	
4 L P[] 1000cm/min CNT50	
SAMPLE3	1/1
[End]	
ED_DEF	TOUCHUP>

NOTE

If the instructions listed on the submenu are necessary, they need not be changed.

- 2 To change a standard motion instruction, press the F1 ED_DEF.

ED_DEF	
F1	

Default Motion	JOINT 30%
	1/4
1 J P[] 100% FINE	
2 J P[] 100% FINE	
3 L P[] 1000cm/min CNT50	
4 L P[] 1000cm/min CNT50	
	DONE

- 3 Move the cursor to the instruction item to be changed (motion type, feedrate, positioning type, or supplementary motion instruction) using the cursor keys.

Default Motion	JOINT 30%
	2/4
1 J P[] 100% FINE	
2 J P[] 100% FINE	
3 L P[] 1000cm/min CNT50	
4 L P[] 1000cm/min CNT50	
Enter value	
REGISTER	[CHOICE] DONE

- 4 Select numeric keys and function keys to correct the instruction item.
To change the feedrate, for instance, move the cursor to feedrate. Enter a new value with numeric keys, then press the ENTER key.

7

0

Old Value: 100

ENTER

Default Motion
JOINT 30%
1/4

1 J P[] 100% FINE
2 J P[] 70% FINE
3 L P[] 1000cm/min CNT50
4 L P[] 1000cm/min CNT50

Enter value
REGISTER
[CHOICE] DONE

- 5 When [CHOICE] is displayed in the F4 key name field, press the F4 key. Then, an option of another instruction item can be selected from the submenu.

[CHOICE]
DONE

F4

Motion Modify
JOINT 30%

1 Fine
2 Cnt
3
4

Default Motion
2/4

1 J P[] 100% FINE
2 J P[] 70% FINE
3 L P[] 1000cm/min CNT50
4 L P[] 1000cm/min CNT50

Select item
[CHOICE] DONE

Motion Modify

1 Fine
2 Cnt
3
4

Default Motion

1 J P[] 100% FINE
2 J P[] 70% CNT50
3 L P[] 1000cm/min CNT50
4 L P[] 1000cm/min CNT50

Enter value
REGISTER
[CHOICE] DONE

5

0

ENTER

- 6 Repeat steps 3 to 5 for each instruction to be changed.
7 After teaching is completed, press the F5 (DONE) key.

[CHOICE]
DONE

F5

5.3.3 Teaching a Motion Instruction

A motion instruction moves the robot to the specified position in the work area at the specified feedrate using the specified movement method. When the motion instruction is taught, the instruction items of the motion instruction and position data are simultaneously taught.

The instruction items of a motion instruction are as follows (see Section 4.3 for the motion instruction):

- Motion type: Controls a path to the specified position. (joint, linear, circular)
- Position variable: Stores data on positions to which the robot moves.
- Feedrate: Specifies the speed of the robot when it moves.
- Positioning type: Specifies whether positioning is performed at the specified position.
- Supplementary motion instruction: Specifies the instruction which executes with the loader robot.

Teaching a motion instruction is selected after a standard motion instruction is created. In this case, the current position (position data) is stored in the position variable.

- Press the F1 key to list the stored standard statements. Choose a desired statement from the list, and then program that statement.
- To program a single standard statement repeatedly, hold down the shift key and press the F1 key.

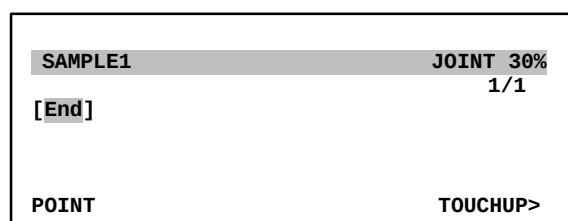


- Press F1 POINT to list the standard operation statements.
- Check whether the position to be programmed is one of the robot's singular points (for singular points, see Position data in 4.3.2). The user can program the position by the joint coordinate, if so desired. (see Singular point check functions in 5.7)

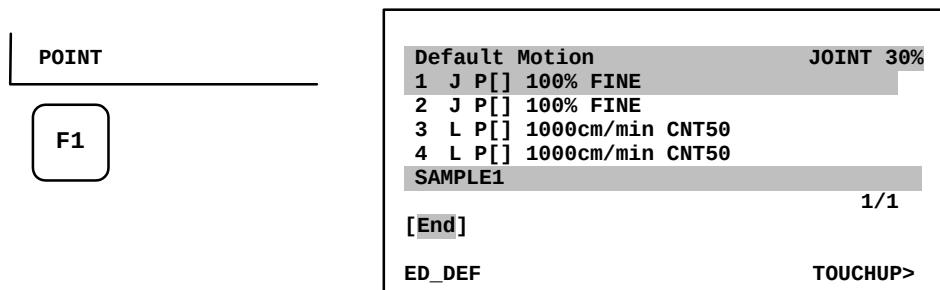
Procedure 5-4 Teaching a motion instruction

Step

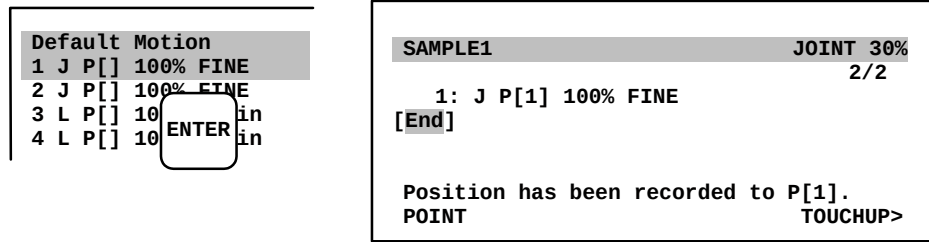
- 1 Move the robot to the desired position in the work area by jog feed.
- 2 Move the cursor to END.



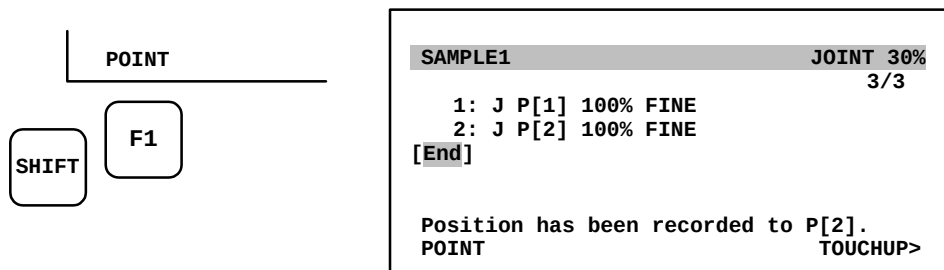
- 3 Press the F1 [POINT] key to display the standard motion instruction menu.



- 4 Select the standard motion instruction to be taught, press the ENTER key, and specify the desired motion instruction. At the same time the position is taught.



- 5 Repeat steps 2 to 4 for each motion instruction to be specified in the program.
 6 To specify the same standard motion instruction repeatedly, press the F1 [POINT] key while pressing the SHIFT key. This adds the previously specified motion instruction to the currently selected standard motion instruction.

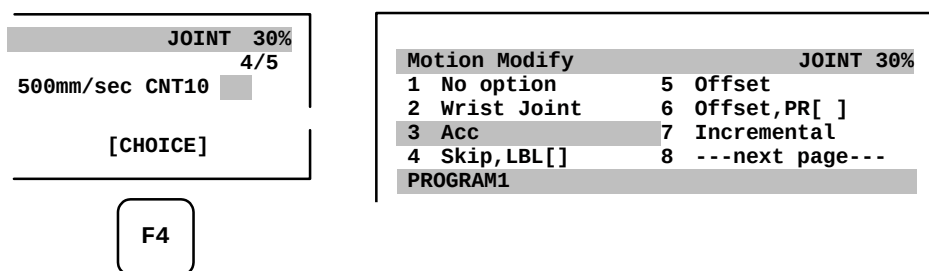


5.3.4 Teaching an Supplementary Motion Instruction

The supplementary motion instruction makes the robot do special work while it is moving according to the motion instruction. Some of the following supplementary motion instructions are provided (see Subsection 4.3.5 for the supplementary motion instructions):

- Wrist joint motion instruction
- Acceleration/deceleration override instruction
- Skip instruction
- Position compensation instruction
- Direct position compensation instruction
- Tool offset instruction
- Direct tool offset instruction
- Incremental instruction
- Path instruction
- Soft float
- Asynchronous additional speed
- Synchronous additional speed
- Pre-execution
- Post-execution

To teach a supplementary motion instruction, place the cursor behind the motion instruction and press the F4 [CHOICE] key to display the supplementary motion instruction menu. Select a supplementary motion instruction from the menu. (See Appendix A.3 for the program instruction menu.)

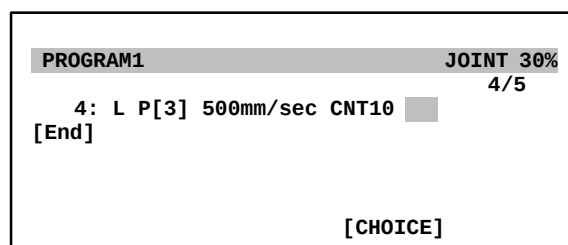
**NOTE**

The available supplementary motion instructions vary according to your software configuration.

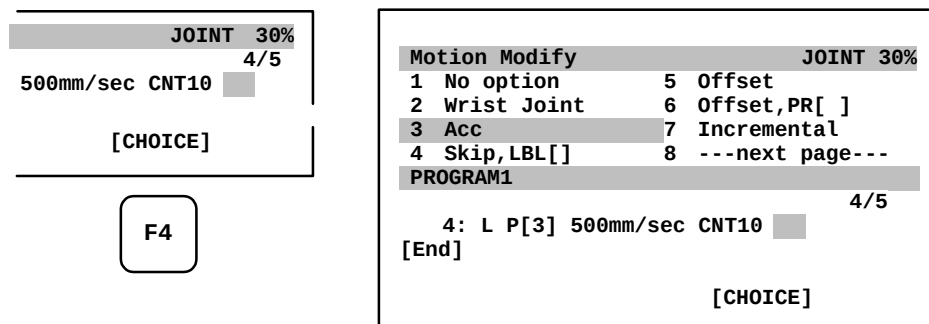
Procedure 5-5 Teaching the supplementary motion instruction

Step

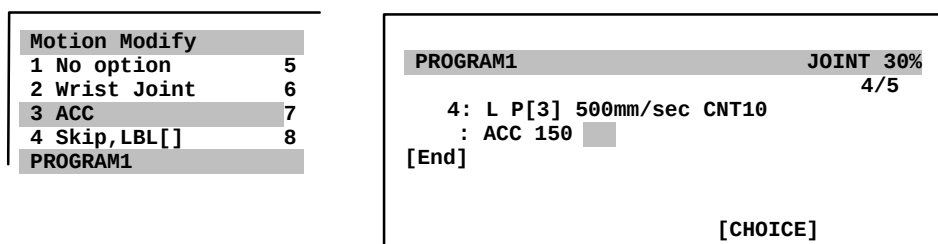
- 1 Place the cursor immediately behind the motion instruction.



- 2 Press the F4 [CHOICE] key. The supplementary motion instruction menu is displayed.



- 3 Select a desired item.
For example, the following screen teaches a acceleration override instruction.



For details of the instructions, see Chapter 4.

Procedure 5-6 Teaching the incremental instruction

Step

- 1 Move the cursor to the space at the end of the motion instruction. The teaching incremental instruction is shown as follow.

JOINT 30%
4/5

100% FINE

[CHOICE]

F4

PROGRAM1 JOINT 30%
4/5

4: J P[3] 100% FINE

[End]

[CHOICE]

Motion Modify JOINT 30%

1 No option	5 Offset
2 Wrist Joint	6 Offset,PR[]
3 ACC	7 Incremental
4 Skip,LBL[]	8 ---next page---

PROGRAM1
4/5

4: J P[3] 100% FINE

PROGRAM1 JOINT 30%
4/5

4: J P[3] 100% FINE INC

[End]

[CHOICE]



CAUTION

Teaching the incremental instruction makes the position data have no position information.

Enter the incremental amount to the position data manually.

- 2 Enter the incremental amount directly to the position data.

[CHOICE] POSITION

F5

PROGRAM1 JOINT 30%
4/5

4: J P[3] 100% FINE INC

[End]

[CHOICE] POSITION

Position Detail JOINT 30%

P[3] GP:1 UF:0 UT:1 CONF:NUT 000

X: ***.*** mm	W: ***.*** deg
Y: ***.*** mm	P: ***.*** deg
Z: ***.*** mm	R: ***.*** deg

PROGRAM1
4/5

4: J P[3] 100% FINE INC

[End]

Enter value

PAGE	CONF	DONE	[PREP]

- 3 Enter the incremental amount directly.

JOINT 30%			
UT:1	CONF:NUT	000	
W:	0.000	deg	
P:	0.000	deg	
R:	***.***	deg	

0

ENTER

Position Detail		JOINT 30%	
P[3]	GP:1 UF:0 UT:1	CONF:NUT	000
X:	500.000 mm	W:	0.000 deg
Y:	100.000 mm	P:	0.000 deg
Z:	100.000 mm	R:	0.000 deg
PROGRAM1			
		4/5	
4: J P[3] 100% FINE INC			
[End]			
Enter value			
PAGE	CONF	DONE	[PREP]

- 4 When you are fished to enter the position data, press F4,DONE.

CONF	DONE	[PREP]

F4

PROGRAM1		JOINT 30%	
		4/5	
4: J P[3] 100% FINE INC			
[End]			
Enter value or press ENTER			
[CHOICE] POSITION			

5.3.5 Teaching a Control Instruction

A control instruction is a program instruction for the Robot controller that is not a motion instruction. The control instructions are as follows:

- Palletizing instruction
- Register instruction
- Position register instruction
- Soft float instruction
- I/O (input/output) instruction
- Branch instruction
- Wait instruction
- Macro instruction
- Program end instruction
- Comment instruction
- Supplementary motion instruction
- Other instructions

To teach a control instruction, first press the F1 (INST) key to display the submenu. Then, select a desired control instruction item from the menu (see Appendix A.3 for the program instructions menu).

Instruction		JOINT 30%	
1 Registers	5 JMP/LBL		
2 I/O	6 CALL		
3 IF/SELECT	7 Palletizing		
4 WAIT	8 ---next page---		
PROGRAM1			

Instruction		JOINT 30%	
1 Miscellaneous	5 MACRO		
2 Program control	6		
3 Skip	7		
4 Offset	8 ---next page---		
PROGRAM1			

NOTE

The program instructions vary according to you software configuration.

Procedure 5-7 Teaching a register instruction

Condition

- The teach pendant must be enabled.
- The program edit screen must be selected.

```

PROGRAM1                                JOINT 30%
                                     2/2
1: J P[1] 100% FINE
[End]

[ INST ]                                [EDCMD]>

```

Step

- 1 Move the cursor to END.
- 2 Press the F1 (INST) key. Then, the control instruction menu is displayed.

[INST]

Instruction		JOINT 30%
1 Registers	5 JMP/LBL	
2 I/O	6 CALL	
3 IF/SELECT	7 Palletizing	
4 WAIT	8 ---next page---	
PROGRAM1		

- 3 To teach a register instruction, select REGISTERS.
The following screens indicate that the value of register [1] is increased by one.

Instruction	
1 Registers	5
2 I/O	6
3 IF/SELECT	7
4 WAIT	8
PROGRAM1	

ENTER

REGISTER statement		JOINT 30%
1 ...=...	5 ...=.../...	
2 ...=...+...	6 ...=...DIV...	
3 ...=...-...	7 ...=...MOD...	
4 ...=...*...	8	
PROGRAM1		

REGISTER statement		JOINT 30%
1 R[]	5	
2 PL[]	6	
3 PR[]	7	
4 PR[i,j]	8	
PROGRAM1		

2/3

```

2:  ...=...+...
[End]

```

REGISTER statement		JOINT 30%
1 R[]	5 RO[]	
2 Constant	6 RI[]	
3 DO[]	7 GO[]	
4 DI[]	8 ---next page---	
PROGRAM1		

2/2

```

2:  R[1]=...+...
[End]

```

REGISTER statement		JOINT 30%
1 R[]	5 RO[]	
2 Constant	6 RI[]	
3 DO[]	7 GO[]	
4 DI[]	8 ---next page---	
PROGRAM1		2/3
2: R[1]=R[1]+...		
[End]		

PROGRAM1		JOINT 30%
		3/3
1:J P[1] 100% FINE		
2: R[1]=R[1]+1		
[End]		
[INST]		[EDCMD]>

For the details of the register instruction, see Chapter 4.

Procedure 5-8 Teaching the position register instruction

Step

- 1 Move the cursor to END.
- 2 Press the F1, [INST] key. Then, the control instruction menu is displayed.

Instruction		JOINT 30%
1 Registers	5 JMP/LBL	
2 I/O	6 CALL	
3 IF/SELECT	7 Palletizing	
4 WAIT	8 ---next page---	
PROGRAM1		

- 3 Select REGISTERS.

REGISTER statement		JOINT 30%
1 ...=...	5 ...=.../...	
2 ...=...+...	6 ...=...DIV...	
3 ...=...-...	7 ...=...MOD...	
4 ...=...*...	8	
PROGRAM1		

- 4 Select PR[].
Teach the instruction assigning the Cartesian coordinates of the current position to the position register on the following screens.

REGISTER statement		JOINT 30%
1 R[]	5	
2 PR[]	6	
3 PR[i, j]	7	
4	8	
PROGRAM1		

REGISTER statement		JOINT 30%
1 Lpos	5	UTOOL[]
2 Jpos	6	PR[]
3 P[]	7	
4 UFRAME[]	8	
PROGRAM1		
		2/3
2: PR[1]=...		
[End]		
Select item		
[CHOICE]		

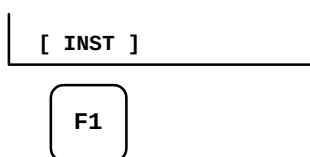
PROGRAM1		JOINT 30%
		3/3
2: PR[1]=LPOS		
[End]		
[INST]		
[EDCMD]>		

For details of the instruction, see Chapter 4.

Procedure 5-9 Teaching an I/O instruction

Step

- 1 Move the cursor to END.
- 2 Press the F1, [INST] key. Then, the control instruction menu is displayed.



Instruction		JOINT 30%
1 Registers	5	JMP/LBL
2 I/O	6	CALL
3 IF/SELECT	7	Palletizing
4 WAIT	8	---next page---
PROGRAM11		

- 3 Select I/O.
Teach the instruction that turns on RO[1] on the following screens.

Instruction	
1 Registers	5
2 I/O	6
3 IF/SELECT	7
4 WAIT	8
PROGRAM1	

I/O statement		JOINT 30%
1 DO[]=...	5	GO[]=...
2 R[]=DI[]	6	R[]=GI[]
3 RO[]=...	7	AO[]=...
4 R[]=RI[]	8	---next page---
PROGRAM1		

I/O statement		加減 30%
1 On	5	
2 Off	6	
3 Pulse(,width)	7	
4 R[]	8	
PROGRAM1		2/3
2:	R0[1]=...	
[End]		

PROGRAM1		JOINT 30%
		3/3
2:	R0[1]=ON	
[End]		
[INST]		[EDCMD]>

For details of the instruction, see Chapter 4.

Procedure 5-10 Teaching move group instructions

Step

- 1 Move the cursor to the line number of a desired move statement (other than for circular movement).

PROGRAM1		JOINT 30%
1:	L P[1] 1000mm/sec CNT100	
[End]		
POINT		TOUCHUP>

- 2 Press F1, [INST]. Then, a list of control instructions is displayed.

Instruction		JOINT 30%
1 Registers	5 JMP/LBL	
2 I/O	6 Independent GP	
3 IF/SELECT	7 Simultaneous GP	
4 WAIT	8 ---next page---	
PROGRAM1		

- 3 Select Independent GP or Simultaneous GP.
The contents of group 1 are moved to another group.
Note that in this case, position data remains unchanged.

```

PROGRAM1                                JOINT 30%

1: Independent GP
: GP1 L P[1] 1000mm/sec CNT100
: GP2 L P[1] 1000mm/sec CNT100
[End]

[ INST ]                                [EDCMD]>

```

- 4 For a move statement within the move group instructions, edit the move type, move speed, and positioning type in the same way as for an ordinary move statement.

Note that the following operations cannot be performed:

- Changing the move type to circular
- Specification of position data type (R[], PR[])
- Position number change
- Teaching of additional move instructions (Deletion is allowed.)
- Deletion/creation of move groups
- Position modification by SHIFT + TOUCHUP

For details of instructions, see Chapter 4.

5.3.6 TP Start Prohibition

The Robot controller can execute the program immediately while editing it. To prevent the program from being executed by mistake, you can prohibit starting the program while teaching with this function.

When you select Disable FWD/BWD in the function menu, starting a program with a teach pendant is prohibited. At this time, "FBD" is reversely displayed in the upper right hand corner of the teach pendant screen to inform that TP FWD/BWD key is disabled.

This "FBD" means "Forward, Backward Disabled". To release the prohibition mode, press Disable FWD/BWD in the function menu again. At this time, the indicator of "FBD" disappears and the override is decreased to the setting value specified in the system variable, \$SCR.\$FWDENBLOVRD, if it is larger than the setting value. (Standard value : 10%)

Though the indicator, "FBD", displayed in upper right hand corner of the screen disappears when the teach pendant is disabled, "FBD" is displayed again when the teach pendant is enabled again.

Press and hold the SHIFT key, and press FWD or BWD in prohibition mode. At this time, a warning message, "FWD/BWD is disabled", is displayed at the first line of the screen.

Jog feed during TP start prohibition

A system variable can be set to enable jog feed only in the TP start prohibition state.

To make this setting, system variable \$SCR.\$TPMOTNENABL is used.

To enable this function (to enable jog feed only in the TP start prohibition state), change the value of system variable \$SCR.\$TPMOTNENABL from 0 to 1 (or from 2 to 3) on the system variable screen.

The table below indicates the relationship between the value of system variable \$SCR.\$TPMOTNENABL and whether TP start and jog feed are enabled.

Table 5.3.6 Setting for Jog feed during TP start prohibition

\$SCR.\$TPMOTNENABL	TP start	Jog feed
0	Enabled	Enabled
1	Enabled	Disabled
2	Disabled	Enabled
3	Disabled	Enabled

With the standard setting, this function is disabled (jog feed is enabled irrespective of whether the teach pendant can start a program).

Procedure 5-11 Prohibiting Starting with Teach Pendant

Step

- 1 Press the FCTN key. The function menu is displayed.
- 2 Select 2 Disable FWD/BWD. "FBD" is reversely displayed in the uppermost right hand line of the screen.

```

1 ABORT (ALL)
2 Disable FWD/BWD
3 CHANGE GROUP

```

FCTN

	LINE 0	FBD
SAMPLE		
SAMPLE		JOINT 30%
		1/1
[End]		
[INST]		[EDCMD]>

- 3 To release the prohibition mode, select "2 Disable FWD/BWD" in the function menu again. "FBD" disappears and the override is reduced to a setting of \$SCR.\$FWDENBLOVRD.

```

1 ABORT (ALL)
2 Disable FWD/BWD
3 CHANGE GROUP

```

FCTN

	LINE 0	
SAMPLE		
SAMPLE		JOINT 30%
		1/1
[End]		
[INST]		[EDCMD]>

Procedure 5-12 When effective/disable of teach pendant is switched

Condition

- TP is in prohibition mode.
- The teach pendant is disabled.

Step

- 1 The following program edit screen is displayed.
"FBD" is not displayed in TP prohibition state because a teach pendant is disabled.

	LINE 0	
SAMPLE		
SAMPLE		JOINT 30%
		1/1
[End]		
[INST]		[EDCMD]>

- 2 Enable the teach pendant.
"FBD" is displayed at uppermost right hand corner of the screen and the override is reduced to the setting of \$SCR.\$FWDENBLOVRD.

SAMPLE	LINE 0	FBD
SAMPLE		JOINT 10%
		1/1
[End]		
[INST]		[EDCMD]>

5.4 CHANGING A PROGRAM

The method of changing the contents of an existing program is described in this section.

- Selecting a program
- Modifying a standard motion instruction
- Changing a motion instruction
- Changing a control instruction
- Editing a program instruction
 - Inserting a blank line
 - Deleting a program instruction
 - Copying a program instruction
 - Finding a program instruction item
 - Replacing a program instruction item
 - Renumbering program lines

Selecting a program

Select a program from the menu of existing programs.

Changing a motion instruction

Change a motion instruction item. An example is position data, which is an instruction item that must be frequently changed.

Changing other instructions

Change other instructions.

5.4.1 Selecting a Program

When selecting a program, call the registered program to display the program edit screen for editing, changing and executing a program.

Once a program is selected, the program is effective until another program is selected. While another screen is displayed such as the current position screen, the currently selected program is started by the start switch.

- In case that the teach pendant is enabled, the current or halted program is forcibly terminated when a program is selected.
- In case that the teach pendant is disabled, another program cannot be selected while a program is being executed or halted.

Select a program on the program list screen.

Procedure 5-13 Selecting a program

Step

- 1 Press the MENUS key.
- 2 Select SELECT.

Alternatively, press the SELECT key to enable a program to be selected. In this case, the program list screen is displayed.

Select		JOINT 30%
61092 bytes free		3/5
No.	Program name	Comment
1	SAMPLE1	JB[SAMPLE PROGRAM1]
2	SAMPLE2	JB[SAMPLE PROGRAM2]
3	SAMPLE3	JB[SAMPLE PROGRAM3]
4	PROG001	PR[PROGRAM001]
5	PROG002	PR[PROGRAM002]
[TYPE] CREATE DELETE MONITOR [ATTR] >		

- 3 Move the cursor to the name of a program to be corrected using the cursor keys (↑ and ↓) press the ENTER key. The selected program edit screen is displayed.

SAMPLE3		JOINT 30%
		1/6
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]		
POINT		TOUCHUP>

5.4.2 Changing a Motion Instruction

When changing a motion instruction, change the instruction items of the motion instruction or change taught position data. For the motion instructions, see Section 4.3.

Changing position data

To change position data, assign new position data to the position variable by pressing the F5 (TOUCHUP) key while pressing the SHIFT key.

Position data information

The coordinates and configuration for position data can be directly changed on the position data information screen.

PAGE	CONF	DONE	[REPRE]

- F2 (PAGE): Toggles between the standard axes and the extended axes
- F3 (CONFIG): Edits the configuration value.
- F4 (DONE): Terminates changing the position data information.
- F5 (REPRE): Toggles between Cartesian coordinates and joint coordinates.

Changing an instruction item

To change an instruction item, press the F4, [CHOICE] key to display the motion instruction item menu, then select an instruction item from the menu.

- Motion type: Controls a path to the end position (joint, linear, circular). When the motion type is changed, the feedrate unit is also automatically changed.
- Position variable: The variable storing position data and the variable number are changed.
- Feedrate: The speed of the robot when it moves (robot motion speed) and the feedrate unit are changed.
- Positioning type: Positioning at the specified position is changed.
- Supplementary motion instruction: An additional instruction to be executed when the robot is moving is changed.

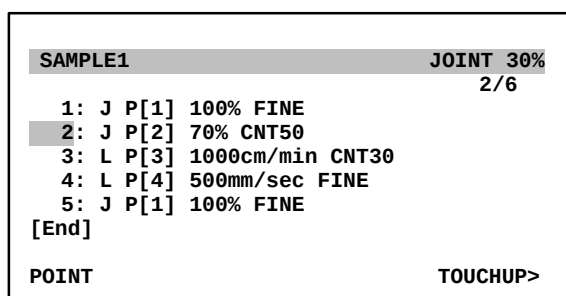
Procedure 5-14 Changing position data

Condition

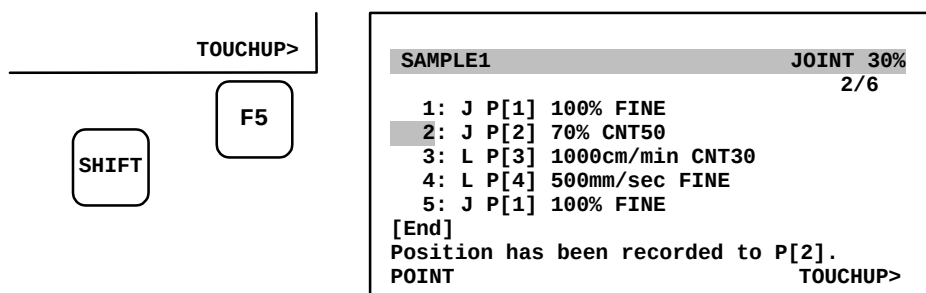
- The program to be changed must be selected.
- The teach pendant must be enabled.

Step

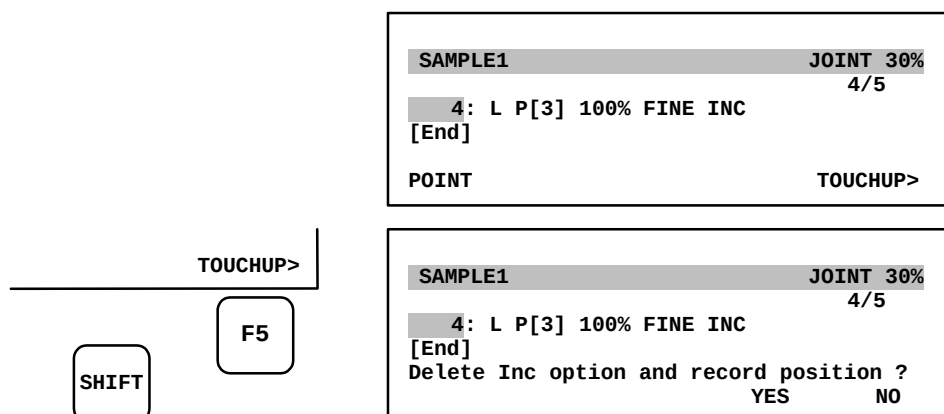
- 1 Move the cursor to the line number at which the motion instruction to be changed is displayed.



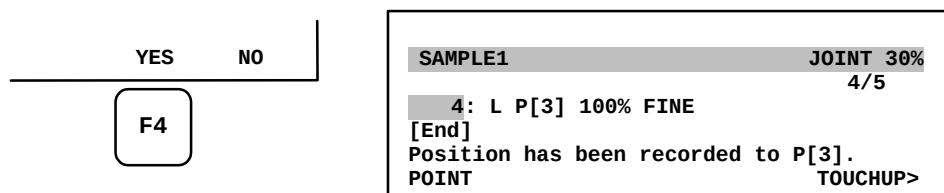
- 2 Move the robot to a new position and press the F5 [TOUCHUP] while pressing the SHIFT key. The new position is recorded.



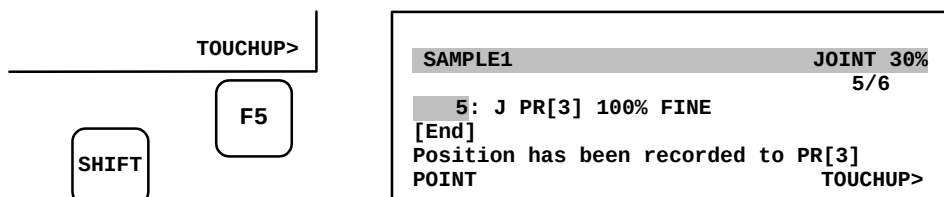
- 3 When the position data is taught to the motion instruction with a incremental option again, a incremental option is removed.



- YES : A incremental option is removed and position data is taught.
- NO : The position data is not taught.



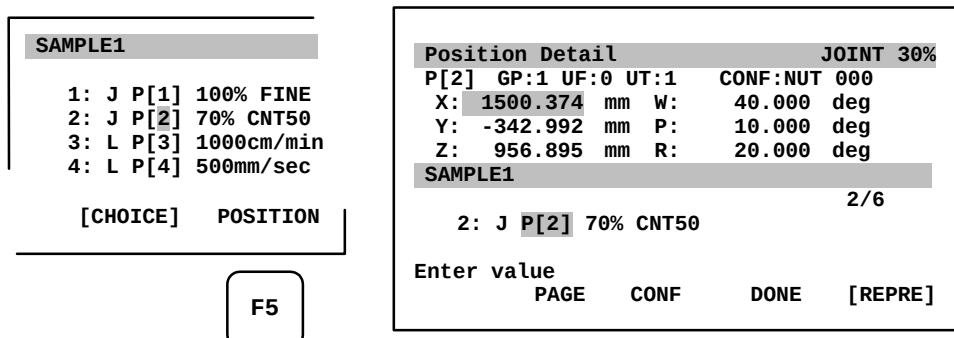
- 4 When position data is taught in the position register as a position variable, the position data in a register is changed by editing.



Procedure 5-15 Changing position data information

Step

- 1 To display position data information, move the cursor to the desired position variable, then press the F5, [POSITION] key. The position data information screen is displayed.



- 2 To change the position, move the cursor to the coordinates for each axis and enter new coordinates.

Position Detail

P[2] GP: 1 UF:0 UT:1

X 1500.374 mm W:

Y -342.992 mm P:

Z 956.895 mm R:

Position Detail JOINT 30%

P[2] GP:1 UF:0 UT:1 CONF:NUT 000

X: 1500.374 mm W: 40.000 deg

Y: -300.000 mm P: 10.000 deg

Z: 956.895 mm R: 20.000 deg

SAMPLE1

-

3

0

0

ENTER

- 3 To change the configuration value, press the F3 [CONFIG] key, move the cursor to the configuration field, then enter a new configuration value with the cursor keys (↑ and ↓).

CONF DONE [REPRE]

F3

Position Detail JOINT 30%

P[2] GP:1 UF:0 UT:1 CONF:NUT 000

X: 1500.374 mm W: 40.000 deg

Y: -300.000 mm P: 10.000 deg

Z: 956.895 mm R: 20.000 deg

SAMPLE1

2/6

2: J P[2] 70% CNT50

Select Flip or Non-flip by UP/DOWN key

PAGE POSITION DONE [PREP]

- 4 To change a coordinate system, press the F5 [REPRE] key and select the coordinate system to be changed.

1 Cartesian

2 Joint

CONF DONE [REPRE]

F5

Position Detail JOINT 30%

P[2] GP:1 UF:0 UT:1

J1: 0.125 deg J4: -95.000 deg

J2: 23.590 deg J5: 0.789 deg

J3: 30.300 deg J6: -120.005 deg

SAMPLE1

NOTE

JOINT display is valid when the robot is adjusted to the zero-degree position or when non-kinematic operation such as table operation control is executed.

- 5 After changing position data information, press the F4 [DONE] key.

CONF DONE [REPRE]

F4

Procedure 5-16 Changing a motion instruction

Step

- 1 Move the cursor to the instruction item of a motion instruction to be changed.

- 2 Press the F4 [CHOICE] key to display the submenu of the instruction items, then select the instruction item to be changed from the submenu.

The following screens show changing the motion type from linear motion to joint motion:

SAMPLE1 5: L P[5] 500cm/min [End] [CHOICE] F4	Motion Modify JOINT 30% <table border="1"> <tr><td>1 Joint</td><td>5</td></tr> <tr><td>2 Linear</td><td>6</td></tr> <tr><td>3 Circular</td><td>7</td></tr> <tr><td>4</td><td>8</td></tr> </table> SAMPLE1 5/6 5: L P[5] 500cm/min CNT30 [End] Select item [CHOICE]	1 Joint	5	2 Linear	6	3 Circular	7	4	8
1 Joint	5								
2 Linear	6								
3 Circular	7								
4	8								
Motion Modify <table border="1"> <tr><td>1 Joint</td></tr> <tr><td>2 Linear</td></tr> <tr><td>3 Circular</td></tr> <tr><td>4</td></tr> </table> ENTER	1 Joint	2 Linear	3 Circular	4	SAMPLE1 JOINT 30% 5/6 5: J P[5] 100% CNT30 [End] Enter value or press ENTER [CHOICE] POSITION				
1 Joint									
2 Linear									
3 Circular									
4									

- 3 The following screens show changing from the position variable to the position register.

SAMPLE1 5: J P[5] 100% CNT [End] [CHOICE] F4	Motion Modify JOINT 30% <table border="1"> <tr><td>1 P[]</td><td>5</td></tr> <tr><td>2 PR[]</td><td>6</td></tr> <tr><td>3</td><td>7</td></tr> <tr><td>4</td><td>8</td></tr> </table> SAMPLE1 5/6 5: J P[5] 100% CNT30 [End]	1 P[]	5	2 PR[]	6	3	7	4	8
1 P[]	5								
2 PR[]	6								
3	7								
4	8								
Motion Modify <table border="1"> <tr><td>1 P[]</td></tr> <tr><td>2 PR[]</td></tr> <tr><td>3</td></tr> <tr><td>4</td></tr> </table> ENTER	1 P[]	2 PR[]	3	4	SAMPLE1 JOINT 30% 5/6 5: J PR[] 100% CNT30 [End] Enter value DIRECT INDIRECT [CHOICE] POSITION				
1 P[]									
2 PR[]									
3									
4									

- 4 Change the feedrate.

SAMPLE1 2: J P[2] 100% FIN 7 0 ENTER	SAMPLE1 JOINT 30% 2/6 2: J P[2] 70% FINE [End] Enter Value [CHOICE]

- 5 Change the feedrate unit.

SAMPLE1 4: L P[4] 500cm/min [End] [CHOICE] F4	Motion Modify JOINT 30% <table style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 50%;">1 mm/sec</td><td style="width: 50%;">5 sec</td></tr> <tr><td>2 cm/min</td><td>6 msec</td></tr> <tr><td>3 inch/min</td><td>7</td></tr> <tr><td>4 deg/sec</td><td>8</td></tr> </table> SAMPLE1 4/6 4: L P[4] 500cm/min CNT30 [End]	1 mm/sec	5 sec	2 cm/min	6 msec	3 inch/min	7	4 deg/sec	8
1 mm/sec	5 sec								
2 cm/min	6 msec								
3 inch/min	7								
4 deg/sec	8								

- 6 Change the positioning type.

JOINT 30% 70% FINE [CHOICE] F4	Motion Modify JOINT 30% <table style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 50%;">1 Fine</td><td style="width: 50%;">5</td></tr> <tr><td>2 Cnt</td><td>6</td></tr> <tr><td>3</td><td>7</td></tr> <tr><td>4</td><td>8</td></tr> </table> SAMPLE1 2/6 2: J P[2] 70% FINE [End] Select item [CHOICE]	1 Fine	5	2 Cnt	6	3	7	4	8
1 Fine	5								
2 Cnt	6								
3	7								
4	8								

Procedure 5-17 Changing a circular motion instruction

Step

- 1 Place the cursor at the motion type of the circular motion instruction to be changed.
The following screens show changing the circular motion instruction to the linear motion instruction.

SAMPLE1 6: C P[5] : P[6] 500cm/min CNT30 [End] [CHOICE] F4	SAMPLE1 JOINT 30% 6/7 6: C P[5] : P[6] 500cm/min CNT30 [End] [CHOICE]

SAMPLE1 6: C P[5] : P[6] 500cm/min [CHOICE] F4	Motion Modify JOINT 30% <table style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 50%;">1 Joint</td><td style="width: 50%;">5</td></tr> <tr><td>2 Linear</td><td>6</td></tr> <tr><td>3 Circular</td><td>7</td></tr> <tr><td>4</td><td>8</td></tr> </table> SAMPLE1 6/7 6: C P[5] : P[6] 500cm/min CNT30	1 Joint	5	2 Linear	6	3 Circular	7	4	8
1 Joint	5								
2 Linear	6								
3 Circular	7								
4	8								

Motion Modify 1 Joint 2 Linear 3 Circular 4	ENTER

SAMPLE1	JOINT 30%
6: L P[6] 500cm/min CNT30	6/7
[End]	
Enter value or press ENTER	
[CHOICE]	POSITION

NOTE

When a circular motion is changed to a joint or linear motion, two motion instructions are created as a result. One instruction moves the tool to the passing point of the circular motion, while the other moves the tool to the end point.

- 2 The following screens show changing the linear motion instruction to the circular motion instruction.

SAMPLE1 6: L P[6] 500cm/min [End] [CHOICE] F4	Motion Modify JOINT 30% 1 Joint 5 2 Linear 6 3 Circular 7 4 8 SAMPLE1 6: L P[6] 500cm/min CNT30 6/7

Motion Modify 1 Joint 2 Linear 3 Circular 4	ENTER

SAMPLE1	JOINT 30%
6: C P[6]	7/7
: P[...] 500cm/min CNT30	
[End]	
Enter value or press ENTER	
[CHOICE]	POSITION

NOTE

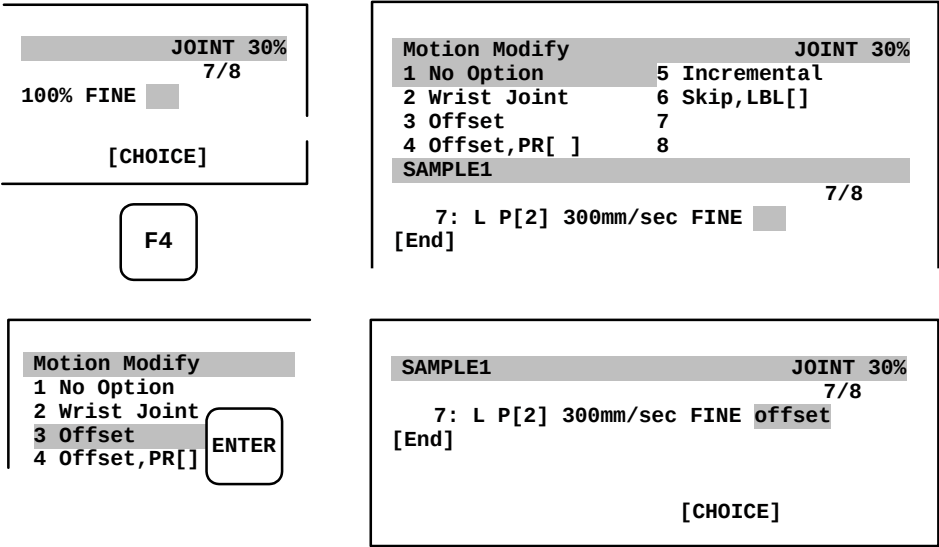
When a joint or linear motion instruction is changed to a circular motion instruction, the taught data for the end point of the circular motion is canceled.

Procedure 5-18 Adding and deleting an additional motion instruction

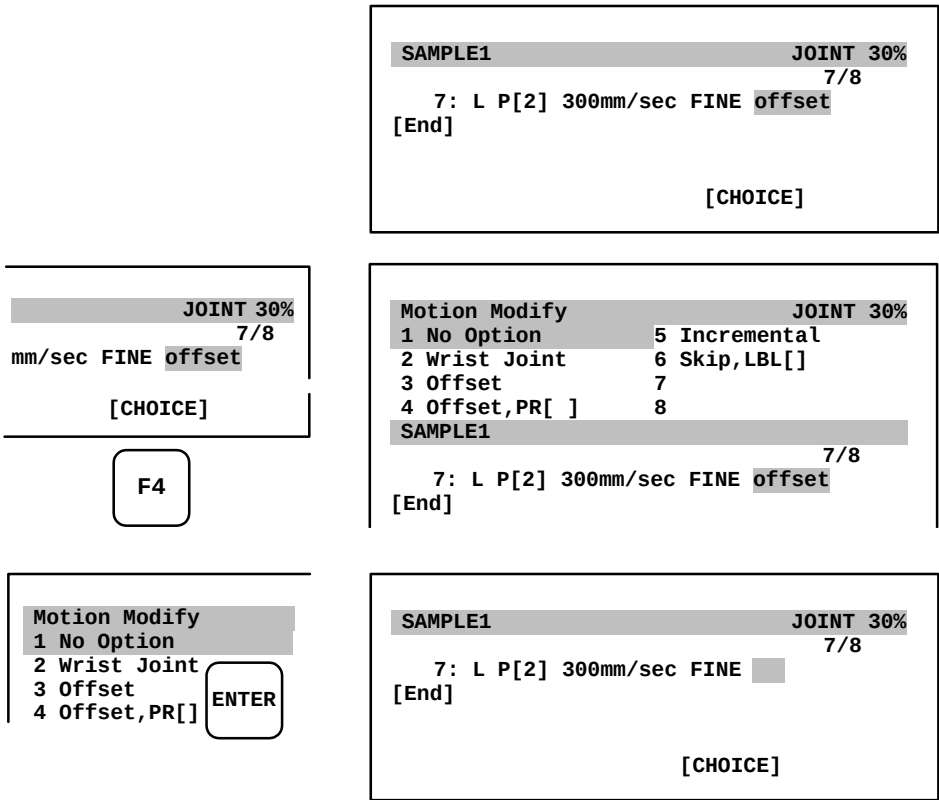
Step

- 1 Position the cursor to an additional motion instruction. To add an offset condition instruction, for example, follow the procedure below:

SAMPLE1	JOINT 30%
7: L P[2] 300mm/sec FINE	7/8
[End]	
[CHOICE]	



2 To delete an offset condition instruction, for example, follow the procedure below:



Procedure 5-19 Changing the move speed (between numeric specification and register specification)

PNS0001		JOINT 10%
		1/2
1: J P[1] 100% FINE		
[End]		
Enter value		
REGISTER		[CHOICE]

Step

- 1 To switch from numeric specification to register specification for the move speed of a move instruction, move the cursor to the speed value. Then, press the function key F1, [REGISTER].

PNS0001		JOINT 10%
		1/2
1: J P[1] R[...]% FINE		
[End]		
Enter value		
SPEED	DIRECT	INDIRECT [CHOICE]

- 2 Enter a desired register number (2 for example). For indirect specification, press F3 [INDIRECT]. (To return to direct specification mode, press F2 [DIRECT].)

PNS0001		JOINT 10%
		1/2
1: J P[1] R[2]% FINE		
[End]		
[CHOICE]		

- 3 To switch from register specification to numeric specification for the move speed of a move instruction

PNS0001		JOINT 10%
		1/2
1: J P[1] R[2]% FINE		
[End]		
Enter value		
SPEED	DIRECT	INDIRECT [CHOICE]

- 4 Move the cursor to the speed value. Then, press the function key F1 (SPEED).

PNS0001		JOINT 10%
		1/2
1: J P[1] ...% FINE		
[End]		
Enter value	REGISTER	[CHOICE]

- 5 Enter a desired speed value (20 for example).

PNS0001		JOINT 10%
		1/2
1: J P[1] 20% FINE		
[End]		
[CHOICE]		

5.4.3 Changing a Control Instruction

You can change the syntax, item, or variable of a control instruction.

Procedure 5-20 Changing a control instruction

Step

- 1 Move the cursor to the instruction item to be changed.

PROGRAM1		JOINT 30%
		11/20
10: J P[5] 100% FINE		
11: WAIT RI[1]=ON		
12: RO[1]=ON		
[CHOICE]		

- 2 Press the F4 (CHOICE) key to display the instruction menu and select the instruction item to be changed.

The following screens show changing the wait instruction.

[CHOICE]	
F4	

Wait statements		JOINT 30%
1 R[]	5 DO[]	
2 Constant	6 DI[]	
3 On	7 RO[]	
4 Off	8 ---next page---	
PROGRAM1		11/20
11: WAIT RI[1]=ON		
12: RO[1]=ON		
Select item	[CHOICE]	

Wait statements	
1 R[]	
2 Constant	
3 On	
4 Off	

ENTER

2

ENTER

[CHOICE]

F4

Wait statements	
1 <Forever>	
2 Timeout-LBL[]	
3	
4	

ENTER

2

ENTER

PROGRAM1	JOINT 30%
	11/20
11: WAIT RI[1]=R[...]	
Enter value	
DIRECT INDIRECT [CHOICE] [LIST]	

PROGRAM1	JOINT 30%
	11/20
11: WAIT RI[1]=R[2]	
12: RO[1]=ON	
[CHOICE]	

PROGRAM1	JOINT 30%
	11/20
11: WAIT RI[1]=R[2] TIMEOUT, LBL[...]	
12: RO[1]=ON	
Enter value	
DIRECT INDIRECT [CHOICE] [LIST]	

PROGRAM1	カサリ 30%
	12/20
11: WAIT RI[1]=R[2] TIMEOUT, LBL[2]	
12: RO[1]=ON	

5.4.4 Program Edit Instructions

The program edit instructions are used to edit a program.

Press the F5 (EDCMDT) key to display the program edit instruction menu and select a desired program edit instruction from the menu.

1 Insert
2 Delete
3 Copy
4 Find
5 Replace
6 Renumber
7 Comment
8 Undo
9 Remark

[EDCMD]

F5

Insert

Inserts blank lines, the number of which is specified, between the existing lines of a program. When blank lines are inserted, the program lines are renumbered.

Delete

Deletes a series of instructions from a program. After the instructions are deleted, the program lines are renumbered.

Copy

Copies a series of instructions and inserts the instruction range into another location in the program. When a series of instructions is copied, the instruction group is selected and recorded in memory. Once the series of instructions is copied, it can be inserted into other locations in the program repeatedly.

Find

A specified element of a program instruction is found. A specified element of a long program can be found quickly.

Replace

Replaces an item of the specified program instruction with another item. This instruction is used, for example, when setup data for the program is changed. (For example, when the I/O allocation is changed, and DO[1] is to be changed to DO[2] in the program)

Comment

On the program editing screen, the user can choose whether to display or hide a comment for the instructions listed below. Note that no comment can be edited.

- DI instruction, DO instruction, RI instruction, RO instruction, GI instruction, GO instruction, AI instruction, AO instruction, UI instruction, UO instruction, SI instruction, SO instruction
- Register instructions
- Position register instructions (including position registers in the position data format for move instructions)
- Palletizing instructions
- Move instruction register speed specifications

The instructions listed below are always accompanied by a comment, and do not allow display switching but allow editing.

- Move instruction position variable
- Label instructions
- Force control instructions

NOTE

- 1 The comment display area for an instruction item that is too long to be displayed on one line of the screen may be shortened.
- 2 No comment is displayed for a register indirect specification.
PR[R[1]] = ...

Renumber

Renumbers the position number in ascending order. Whenever a motion instruction is taught, the position number is increased regardless of its location in the program. When insertions and deletions are repeated, the position numbers are not sequentially arranged in a program. Renumbering arranges them sequentially in the program.

Undo

Program edit operations such as an instruction modification, line insertion, and line deletion can be cancelled to return to the state present before those edit operations are performed. If an undo operation is performed during editing of a program line, all operations performed for that line are undone. For example, if a line is inserted or deleted, the state before the insertion or deletion operation is restored.

If an undo operation is immediately followed by another undo operation, the state present before the first undo operation is performed is restored.

NOTE

If an undo operation is performed for a line during program editing, all operations performed for that line are undone. This means that if an instruction is taught in a blank line or the last line of a program, and an undo operation is performed for that line during editing, the taught instruction is deleted.

Remark

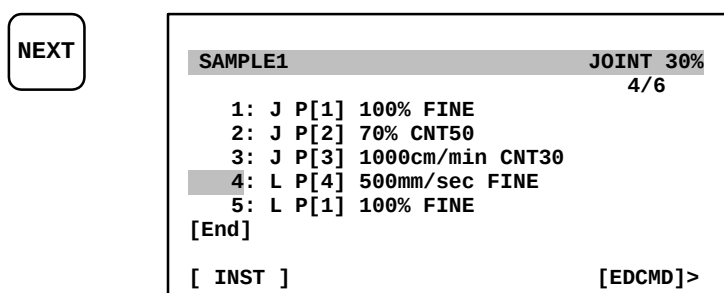
Remarks or unremarks a program instruction. An remarked instruction is not executed at program execution. About detail of this function, refer to 9.50 LINE REMARK instruction.

NOTE

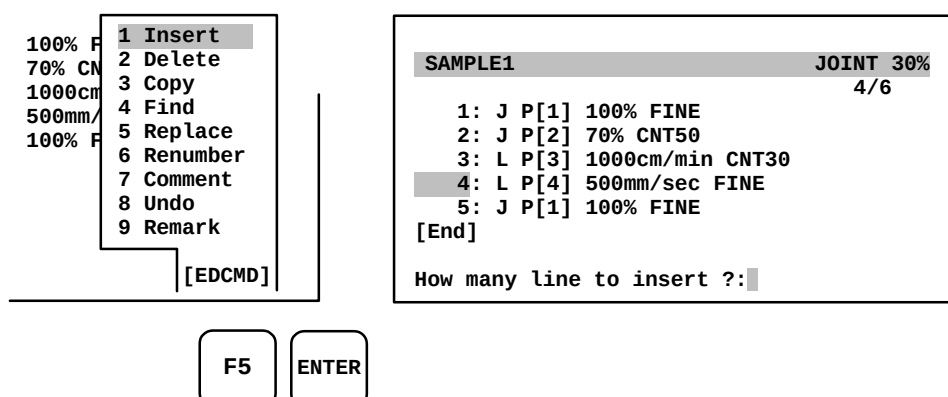
The Line Remark instruction can be used in the version 7DA7/05 or later.

Procedure 5-21 Inserting blank lines**Step**

- 1 Press the NEXT, > to display F5, EDCMD.

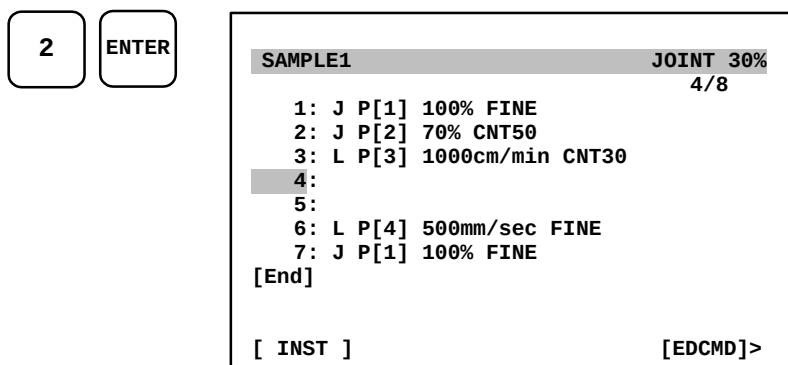


- 2 Press the F5, [EDCMD] key. The edit instruction menu is displayed.
- 3 Select Insert.



In the example below, two blank lines are inserted between the 3rd and 4th lines.

- 4 Move the cursor to the line where instructions are to be inserted. In this example, move the cursor to the 4th line.
- 5 Enter the number of blank lines to be inserted (two) and press the ENTER key.

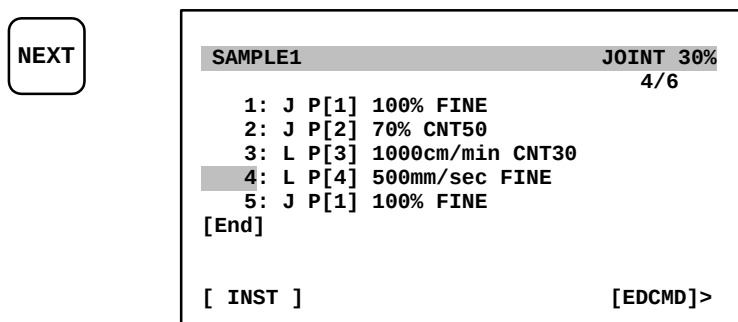


The two blank lines are inserted into the program and all the lines in the program are renumbered.

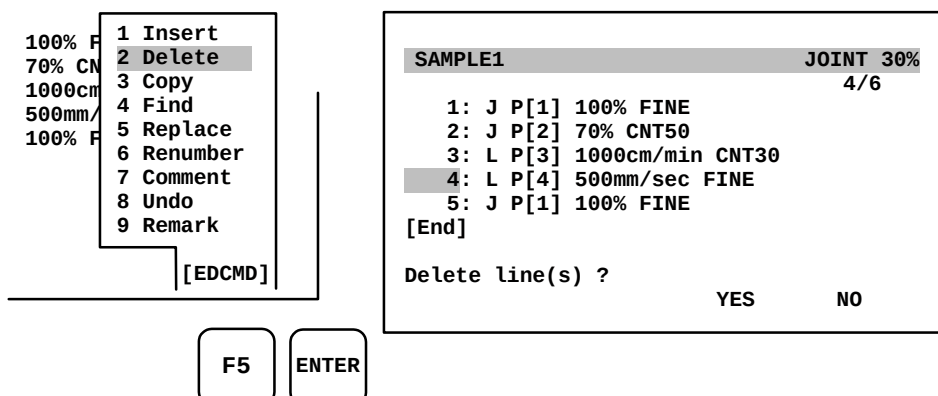
Procedure 5-22 Deleting instructions

Step

- 1 Move the cursor to the top of the line in which the instruction to be deleted is positioned. (Specify the line to be deleted with the cursor.)
- 2 Press the NEXT, > to display F5, EDCMD.



- 3 Press the F5, [EDCMD] key to display the editing instruction menu.
- 4 Select Delete



CAUTION

Once an instruction is deleted, the instruction is not restored. Be sure to confirm whether an instruction to be deleted should be done before doing it, or important data may be lost.

- 5 Specify the range of instruction lines to be deleted with the cursor keys (↑ and ↓).

```

3: L P[3] 1000cm/
4: L P[4] 500mm/s
5: J P[1] 100% FI
[End]

```



- 6 To cancel deleting the selected line, press the F5 (NO) key. To delete the selected lines, press the F4 (YES) key.

YES NO

F4

SAMPLE1 **JOINT 30%**

4/4

1: J P[1] 100% FINE

2: J P[2] 70% CNT50

3: L P[3] 1000cm/min CNT30

[End]

[INST] [EDCMD]>

Procedure 5-23 Copying and pasting instructions

Step

- 1 Press the NEXT, > until F5, EDCMD.

NEXT

SAMPLE1 **JOINT 30%**

1/6

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]

[INST] [EDCMD]>

- 2 Press the F5, [EDCMD] key. The editing instruction menu is displayed.
- 3 Select Copy.

The following screens show copying 2th to 4th lines to 5th to 7th lines.

100% F

70% CN

1000cm

500mm/

100% F

1 Insert

2 Delete

3 Copy

4 Find

5 Replace

6 Renumber

7 Comment

8 Undo

9 Remark

[EDCMD]

SAMPLE1 **JOINT 30%**

2/6

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]

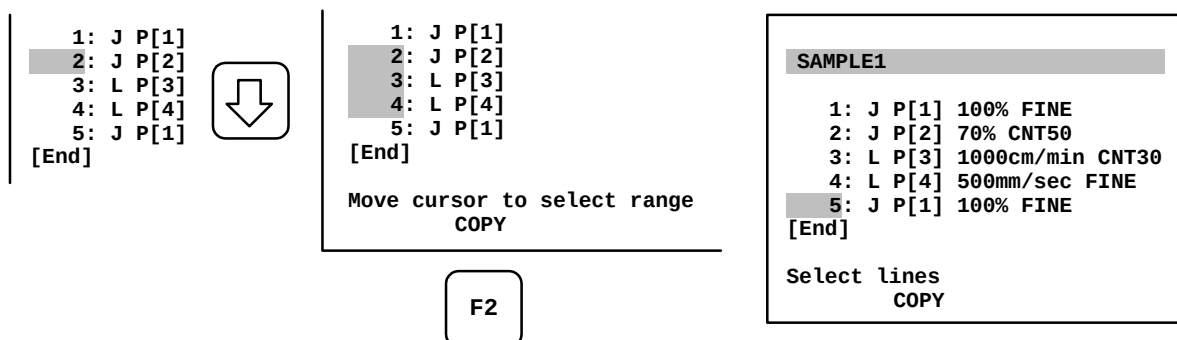
Select lines

COPY PASTE

F5

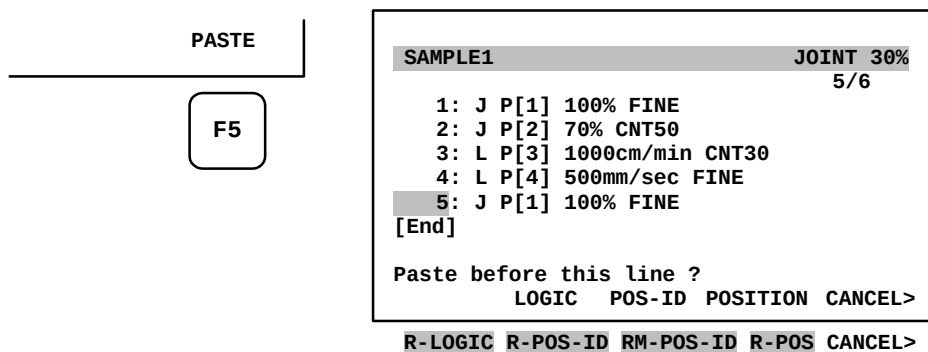
ENTER

- 4 Select the range of lines to be copied.

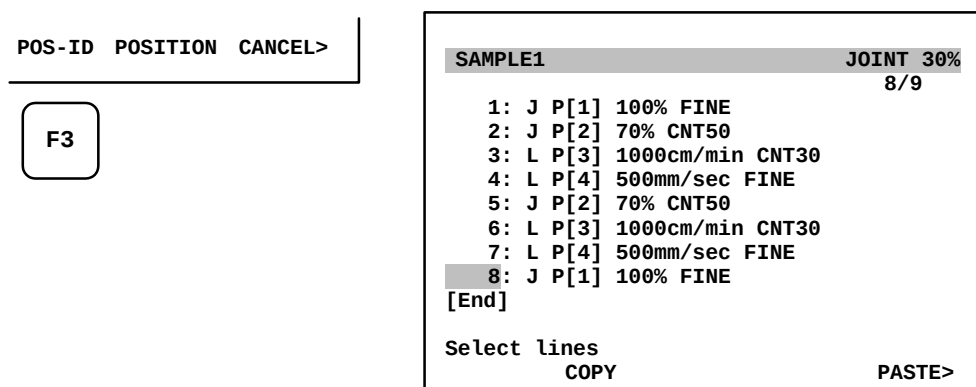


As a result of above steps, the selected instructions (2nd to 4th lines in this example) were copied in memory.

- 5 Decide where you want to paste the sentences copied in the memory.



- 6 Select the copying and pasting method (copying from the original).



- 7 By repeating the above steps 5 to 6, the same instruction group can be pasted at any number of locations in the program.
- 8 To terminate the pasting of instructions, press the PREV key

PREV

Pasting methods

The following copying and pasting methods are provided:

LOGIC POS-ID POSITION CANCEL>

- F2 (LOGIC) : Copies and pastes motion instructions with no position data specified.
- F3 (POS-ID) : Copies and pastes motion instructions with the position numbers unchanged.
- F4 (POSITION) : Copies and pastes motion instructions with the position numbers updated.

Pressing the next page key (NEXT) displays the following function key menu:

NEXT

R-LOGIC R-POS-ID RM-POS-ID RPOS RM-POS>

The selected instructions are copied in reverse order.

F3 and F5 have the following functions:

- F3 (RM-POS-ID) : Copies the move instructions at a copy source in reverse order without changing the position numbers of the move instructions. The move type, move speed, and so forth of each move instruction are changed so that a movement totally opposite to the movement of the copy source is made.
- F5 (RM-POS) : Copies the move instructions at a copy source in reverse order. Then assigns new position numbers. The move type, move speed, and so forth of each move instruction are changed so that a movement totally opposite to the movement of the copy source is made.

NOTE

The copy function for a reverse movement is not supported for the additional move instructions listed below. If the move instructions at a copy source include any of the move instructions below, RM-POS-ID or RM-POS generates a warning, and only a copy operation in reverse order is performed.

- Application instruction
- Skip and high-speed skip instructions
- Incremental instruction
- Continuous rotation instruction
- Pre-execution/post-execution instruction
- Multi-group operation

Example When the F4 (R-POSITION) is pressed

R-POS-ID RM-POS-ID R-POS

F4

```
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[2] 70% CNT50
6: L P[3] 1000cm/min CNT30
7: L P[4] 500mm/sec FINE
8: J P[1] 100% FINE
[End]
```

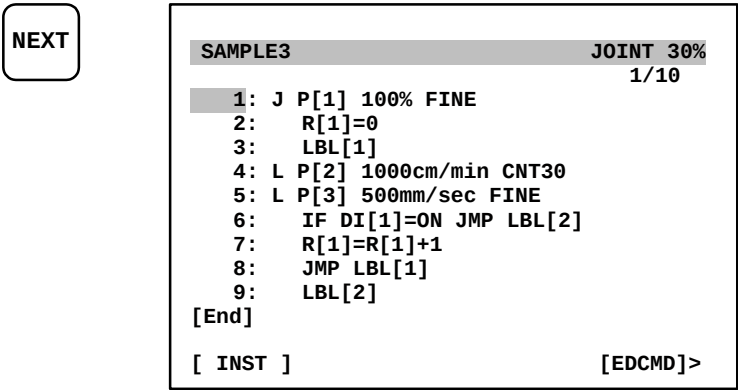
Select lines
COPY

PASTE>

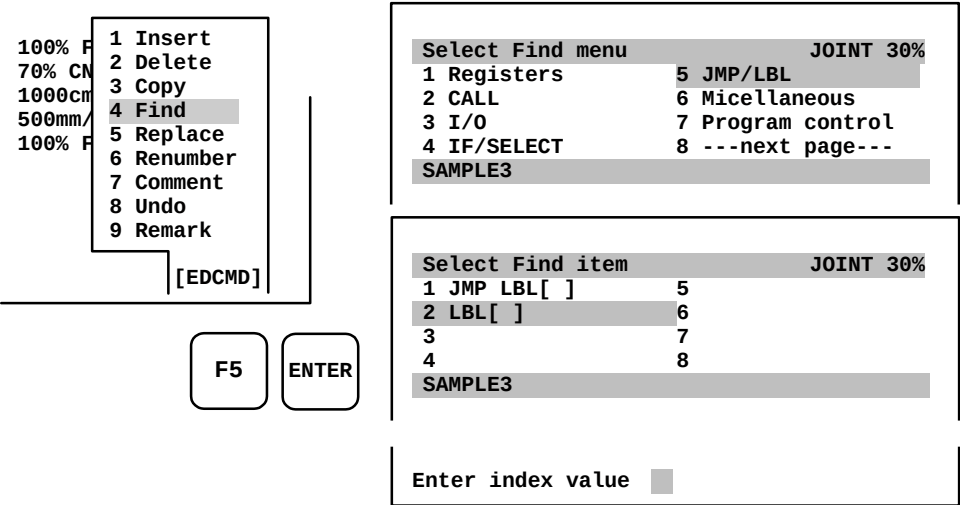
Procedure 5-24 Finding a program instruction item

Step

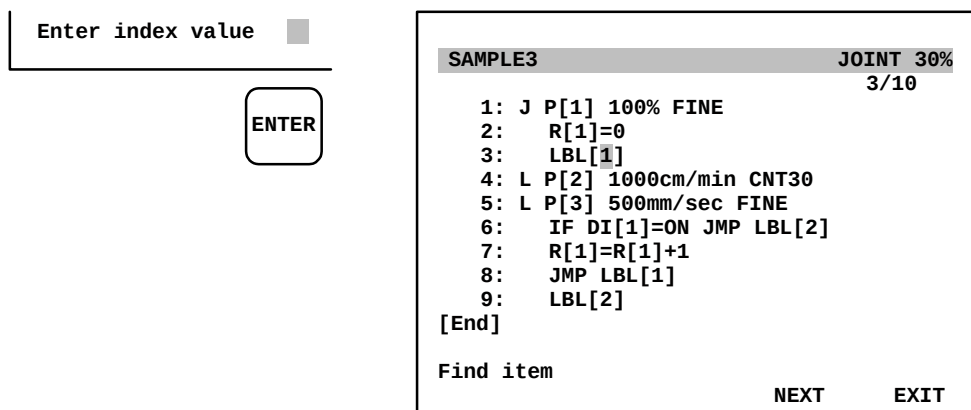
- 1 Press the NEXT, > until F5, EDCMD.



- 2 Press the F5, [EDCMD] key. The editing instruction menu is displayed.
- 3 Select Find.
- 4 Select a program instruction item to be found. The following screens show how to find instruction, LBL.

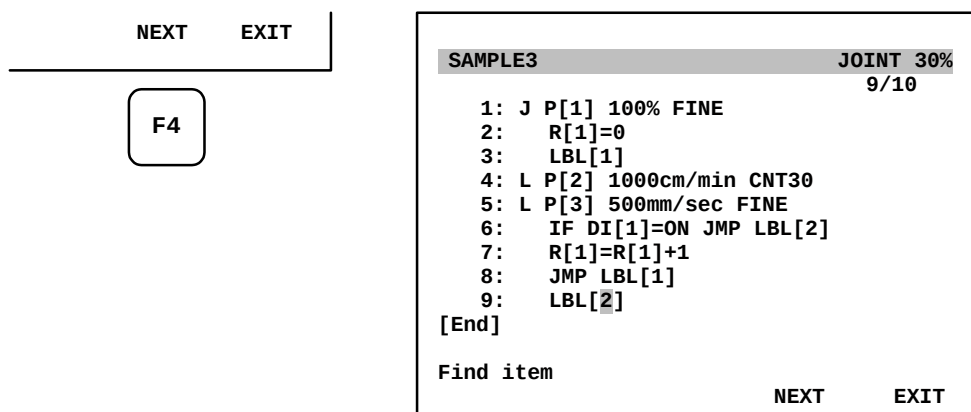


- 5 When the item to be found is an index, enter the value. To find an item regardless of whether the item is an index, press the ENTER key without entering anything.

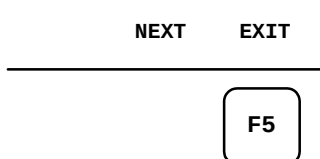


If the specified instruction is found in the program, the cursor stops at the instruction.

- 6 To find the same instruction again, press the F4 (NEXT) key.



- 7 To terminate finding an instruction, press the F5 (EXIT) key.



NOTE

The position of a track/offset instruction or touch sensor instruction cannot be found using the search instruction.

Procedure 5-25 Replacing a program instruction item

Step

- 1 Press the NEXT, > until F5, EDCMD.

```

SAMPLE3                                JOINT 30%
                                      1/9
1: J P[1] 100% FINE
2: J P[2] 70% CNT50
3: LBL[1]
4: L P[3] 1000cm/min CNT30
5: L P[4] 500mm/sec FINE
   : Skip LBL[2]
6: JMP LBL[1]
7: LBL[2]
8: J P[5] 100% FINE
[End]
[ INST ]                                [EDCMD]>

```

- 2 Press the F5, [EDCMD] key. The changing instruction menu is displayed.
- 3 Select Replace.
- 4 Select a program instruction item to be replaced and press the ENTER key.
In the screen below the feedrate specified in the motion instruction is changed to another value.

1 Insert
 2 Delete
 3 Copy
 4 Find
 5 **Replace**
 6 Renumber
 7 Comment
 8 Undo
 9 Remark

[EDCMD]

F5

ENTER

Select Replace menu JOINT 30%
 1 Registers 5 Motion modify
 2 CALL 6
 3 I/O 7
 4 JMP/LBL 8
 SAMPLE3

Modify motion menu JOINT 30%
 1 Replace speed 5
 2 Replace term 6
 3 Insert option 7
 4 Remove option 8
 SAMPLE3

The following replacement items are displayed:

- Replace speed: Changes the feedrate to another value.
 - Replace term: Changes the positioning type to another value.
 - Insert option: Inserts a supplementary motion instruction.
 - Remove option: Deletes a supplementary motion instruction.
- 5 Select Replace speed.

Modify motion menu
 1 **Replace speed**
 2 Replace term
 3 Insert option
 4 Remove option
 SAMPLE3

ENTER

Select interpolate JOINT 30%
 1 Unspecified type 5
 2 J 6
 3 L 7
 4 C 8
 SAMPLE3

- Unspecified type : Changes the feedrates in all motion instructions
- J : Changes the feedrates only in motion instructions for joint control.
- L : Changes the feedrates only in motion instructions for linear control.
- C : Changes the feedrates only in motion instructions for circular control.

6 Specify the target type of the operation instruction.

Select interpolate 1 Unspecified type 2 J 3 L 4 C SAMPLE3	Speed type menu JOINT 30% 1 All type 5 2 Speed value 6 3 R[] 7 4 R[R[]] 8 SAMPLE3

ENTER

- ALL type : No speed type is specified.
- Speed value : Operation statements that specify a speed with a numeric value are specified.
- R[] : Operation statements that specify a speed with a register are specified.
- R[R[]] : Operation statements that indirectly specify a speed value with registers are specified.

7 Specify a target speed format.

Speed type menu 1 All type 5 2 Speed value 6 3 R[] 4 R[R[]] SAMPLE3	Select motion item JOINT 30% 1 % 5 deg/sec 2 mm/sec 6 sec 3 cm/sec 7 4 inch/min 8 SAMPLE3

ENTER

8 Specify a target speed unit.

Select motion item 1 % 5 2 mm/sec 6 3 cm/min 4 inch/min SAMPLE3	Speed type menu JOINT 30% 1 Sped value 5 2 R[] 6 3 R[R[]] 7 4 8 SAMPLE3

ENTER

- Speed value : The selected statement is changed to an operation statement which specifies a speed with a numeric value.
- R[] : The selected statement is changed to an operation statement which specifies a speed using a register.
- R[R[]] : The selected statement is changed to an operation statement which indirectly specifies a speed by using registers.

9 Specify the speed format to be changed.

Speed type menu 1 Speed value 5 2 R[] 6 3 R[R[]] 4 SAMPLE3	Enter speed value:

ENTER

10 Enter a desired feedrate.

Enter speed value: **50**

5

0

ENTER

SAMPLE3
JOINT 30%

1/9

1: J P[1] 100% FINE
2: J P[2] 70% CNT50

Modify OK ?

ALL YES NEXT EXIT

The kinds of replacing items are displayed.

- F2 (ALL) : Replaces all the items in the current line and subsequent lines.
- F3 (YES) : Replaces the item at the cursor and finds the next item.
- F4 (NEXT) : Finds the next item.

12 Select a replacement method.

Modify OK ?

ALL YES

SAMPLE3
JOINT 30%

1/9

1: J P[1] 50% FINE
2: J P[2] 50% CNT50
3: LBL[1]
4: L P[3] 1000cm/min CNT30
5: L P[4] 500mm/sec FINE
: Skip LBL[2]
6: JMP LBL[1]
7: LBL[2]
8: J P[5] 50% FINE
[End]

[INST]
[EDCMD]>

F2

13 To terminate item replacement, press the F5 (EXIT) key.

YES NEXT EXIT

F5

CAUTION

The replacement instruction allows no move instruction to be replaced with the track/offset instruction or touch sensor instruction. If an attempt for such replacement is made, a memory write alarm is issued. To replace a move instruction, first delete the move instruction, then insert the touch sensor instruction or track instruction.

Procedure 5-26 Renumbering the position number

Step

- 1 Press the NEXT, >, then press the F5, EDCMD.

```

SAMPLE1                                JOINT 30%
                                      1/8
1: J P[8] 100% FINE
2: J P[6] 70% CNT50
3: L P[3] 1000cm/min CNT30
4: L P[5] 500mm/sec FINE
5: J P[1] 100% FINE
6: L P[5] 500mm/sec FINE
7: J P[8] 100% FINE
[End]
[ INST ]                                [EDCMD]>

```

- 2 Press F5, EDCMD. The changing instruction menu is displayed.
- 3 Select Renumber.

```

1 Insert
2 Delete
3 Copy
4 Find
5 Replace
6 Renumber
7 Comment
8 Undo
9 Remark
[EDCMD]

```

```

Renumber OK ?                                YES    NO

```

F5 ENTER

- 4 To renumber the program lines, press the F4 (YES) key. To cancel renumbering the program lines, press the F5 (NO) key.

```

YES    NO

```

F4

```

SAMPLE1                                JOINT 30%
                                      1/8
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[5] 100% FINE
6: L P[4] 500mm/sec FINE
7: J P[1] 100% FINE
[End]
[ INST ]                                [EDCMD]>

```

Procedure 5-27 Comment display switching**Step**

- 1 Press NEXT to display F5 (EDCMD).

```

PNS0001                                JOINT 10%
                                         1/10
1:  R[1]=DI[2]
2:  DO[3]=ON
3:  R[R[1]]=DI[R[2]]
4:  PR[1]=P[3]
5:  PR[1,2]=PR[R[3],R[4]]
6:  PL[1]=PL[R[3]]
7:  J PR[1] 100% FINE
8:  J P[1] 100% FINE
9:  LBL[1]
[End]
[ INST ]                                [EDCMD]>

```

- 2 Press F5 (EDCMD) to display the edit instruction menu.

```

PNS0001                                JOINT 10%
                                         1/10
1:  R[1]=DI[2]
2:  DO[3]=ON
3:  R[R[1]]=DI[R[2]]
4:  PR[1]=P[3]
5:  PR[1,2]=PR[R[3],R[4]]
6:  PL[1]=PL[R[3]]
7:  J PR[1] 100% FINE
8:  J P[1] 100% FINE
9:  LBL[1]
[End]
[ INST ]                                EDCMD

```

1 Insert
 2 Delete
 3 Copy
 4 Find
 5 Replace
 6 Renumber
 7 **Comment**
 8 Undo
 9 Remark

- 3 Select Item Comment.

```

PNS0001                                JOINT 10%
                                         1/10
1:  R[1:Comment]=DI[2:Comment]
2:  DO[3:Comment]=ON
3:  R[R[1]]=DI[R[2]]
4:  PR[1:Comment]=P[3:Comment]
5:  PR[1,2:Comment]=PR[R[3],R[4]]
6:  PL[1:Comment]=PL[R[3]]
7:  J PR[1:Comment] 100% FINE
8:  J P[1:Comment] 100% FINE
9:  LBL[1:Comment]
[End]
[ INST ]                                [EDCMD]>

```

- 4 To disable comment display, select Comment of the function key F5 (EDCMD) again.

Procedure 5-28 Undoing edit operations

Step

- 1 Press NEXT to display F5 (EDCMD).

```

PNS0001                                JOINT 10%
                                         1/10
1:  R[1]=DI[2]
2:  DO[3]=ON
3:  R[R[1]]=DI[R[2]]
4:  PR[1]=P[3]
5:  PR[1,2]=PR[R[3],R[4]]
6:  PL[1]=PL[R[3]]
7:  J PR[1] 100% FINE
8:  J P[1] 100% FINE
9:  LBL[1]
[End]
[ INST ]                                [EDCMD]>

```

- 2 Press F5 (EDCMD) to display the edit instruction menu.

```

PNS0001                                JOINT 10%
                                         1/10
1:  R[1]=DI[2]
2:  DO[3]=ON
3:  R[R[1]]=DI[R[2]]
4:  PR[1]=P[3]
5:  PR[1,2]=PR[R[3],R[4]]
6:  PL[1]=PL[R[3]]
7:  J PR[1] 100% FINE
8:  J P[1] 100% FINE
9:  LBL[1]
[End]
[ INST ]                                EDCMD

```

- 1 Insert
- 2 Delete
- 3 Copy
- 4 Find
- 5 Replace
- 6 Renumber
- 7 Comment
- 8 **Undo**
- 9 Remark

- 3 Select Item Undo.

```

PNS0001                                JOINT 10%
                                         1/10
1:  R[1]=DI[2]
2:  DO[3]=ON
3:  R[R[1]]=DI[R[2]]
4:  PR[1]=P[3]
5:  PR[1,2]=PR[R[3],R[4]]
6:  PL[1]=PL[R[3]]
7:  J PR[1] 100% FINE
8:  J P[1] 100% FINE
9:  LBL[1]
[End]
Undo?(Edit)                            YES      NO

```

- 4 To perform an undo operation, select F4, (YES). To cancel the undo operation, select F5, (NO). When F4, (YES) is selected, the edit operation is undone.

```

PNS0001                                JOINT 10%
                                         1/10
1:  R[1]=RI[2]
2:  DO[3]=ON
3:  R[R[1]]=DI[R[2]]
4:  PR[1]=P[3]
5:  PR[1,2]=PR[R[3],R[4]]
6:  PL[1]=PL[R[3]]
7:  J PR[1] 100% FINE
8:  J P[1] 100% FINE
9:  LBL[1]
[End]
[ INST ]                                [EDCMD]>

```

- 5 When an additional undo operation is performed in succession, the first undo operation performed can be cancelled; this means the state present before the first undo operation is performed is restored.

NOTE

If an edit operation is performed after an undo operation, the undo operation cannot be cancelled.

CAUTION

An undo operation automatically rewrites the program, so that the results may not be those expected by the operator. Before executing a program after an undo operation, carefully check the program.

- This function can undo the following operations:
 - a) Instruction modifications
 - b) Line insertion
 - c) Line deletion
 - d) Copying of program statements (reading)
 - e) Copying of program statements (insertion)
 - f) Program instruction replacement
 - g) Reassignment of position numbers
- An undo operation cancels all edit operations performed on the line where the cursor is currently placed, and restores the state present before those edit operations are performed.
- The undo function is disabled when any of the following operations is performed:
 - a) Power-off
 - b) Selection of another program
- Undo operation cannot be performed in any of the following states:
 - a) The teach pendant is disabled.
 - b) The program is write-protected.
 - c) Program memory is insufficient.
- The following edit operations cannot be undone:
 - a) Teaching and editing of palletizing instructions
 - b) Deletion of lines including palletizing instructions
 - c) Copying of lines including palletizing instructions (reading)
 - d) Copying of lines including palletizing instructions (insertion)
 - e) Replacement in a program including palletizing instructions
 - f) Number reassignment in a program including palletizing instructions
- If the power is turned off while an undo operation is being performed, the undo operation is stopped. Note that in this case, the program may become unusable.
- If any of the following instructions is performed after an edit operation, the undo function cannot be performed:

- a) Laser instruction
- b) Palletizing instruction
- c) Spot welding instruction
- d) Line tracking instruction
- If any of the following function is executed after an edit operation, the undo function cannot be performed:
 - a) Online position modification
 - b) Fine adjustment of welding speed

5.5 PROGRAM OPERATION

This section describes the following program operations:

- Changing program information
- Deleting a program
- Copying a program
- Displaying the attribute of a program

5.5.1 Changing Program Information

The program header information is changed with a program detail screen(see Section 4.1). Setting without the motion group can be done. The following items can be set:

- Program name: Name of program to be changed.
- Subtype: The subtype of a program to be changed.
- Comments: The comments in the program to be changed.
- Group mask: Specifies a motion group to be controlled in a program. You can also set so a program has no motion group.
- Write protection: Prevents the modification of a program.
- Interruption disable: Causes a program that has no motion group no to be paused by the alarm whose severity is SERVO or lower, the emergency stop, and the hold.
- Stack size: Memory size used to call program is changed.

Display the following items on the program information screen:

- Creation Date:
- Modification Date:
- Name of the file to be copied
- Positions: FALSE/TRUE
- Memory area size of program

Deleting a program

The unnecessary program can be deleted.

Copying a program

The program with another name in the same content can be reproduced.

Display of a program attribute

The following program header information can be displayed on the program list screen:

- Comment - The comment in a header information is displayed.
- Protection - The settings of "Write protect:" in a header information is displayed
- Last Modified - The settings of "Modification Date:" in a header information is displayed.
- Size - The number of lines of program and memory size are displayed.
- Copy Source - The settings of "Copy Source:" in a header information is displayed.

**CAUTION**

All of the free memory size displayed on the list screen may not be usable to store a program. Even if the size of free memory is not 0, for example, no program may be creatable.

Procedure 5-29 Changing program information

Condition

- The teach pendant must be enabled.

Step

- 1 Press the MENUS key to display the screen menu.
- 2 Select SELECT. The program list screen is displayed.
Alternatively, press the SELECT key to display the program list screen.

Select		JOINT 30%
61092 bytes free		3/5
No.	Program name	Comment
1	SAMPLE1	JB[SAMPLE PROGRAM1]
2	SAMPLE2	JB[SAMPLE PROGRAM2]
3	SAMPLE3	JB[SAMPLE PROGRAM3]
4	PROG001	PR[PROGRAM001]
5	PROG002	PR[PROGRAM002]
[TYPE] CREATE DELETE MONITOR [ATTR] >		
COPY	DETAIL	LOAD SAVE PRINT >

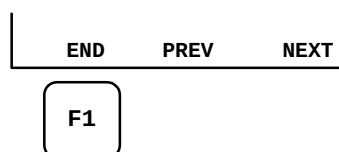
- 3 Press NEXT, > to display the next page, then press the F2, (DETAIL) key. The program information screen is displayed.

COPY	DETAIL	LOAD
	F2	

Program detail		JOINT 30%
		1/7
Creation Date:	10-MAR-1998	
Modification Data:	11-MAR-1998	
Copy Source:		
Positions: FALSE	Size:	312 Bytes
Program name:		
1	SAMPLE3	
2	Sub Type:	[None]
3	Comment:	[SAMPLE PROGRAM 3]
4	Group Mask:	[1,*,*,*,*,*,*]
5	Write protect:	[OFF]
6	Ignore pause:	[OFF]
7	Stack size:	[300]
END	PREV	NEXT

- 4 Specify each item (see Section 4.1).
If the motion instruction is taught in the program, you can not set the 3 "Group Mask:" of this program.

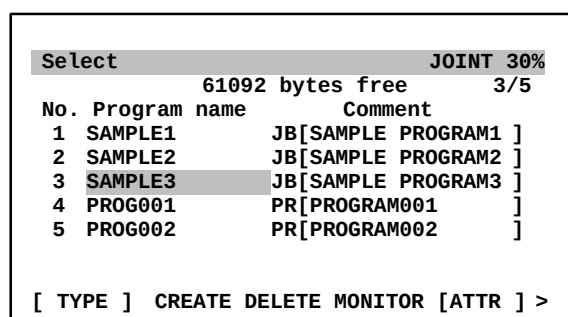
- 5 After specifying program information, press the F1, (END) key.



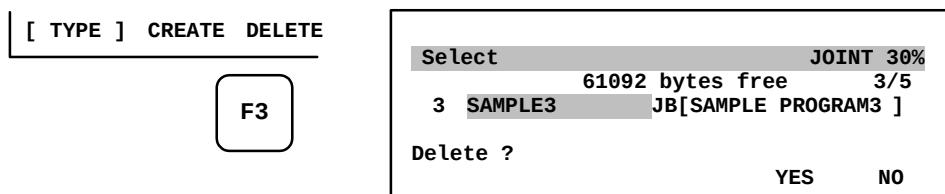
Procedure 5-30 Deleting a program

Step

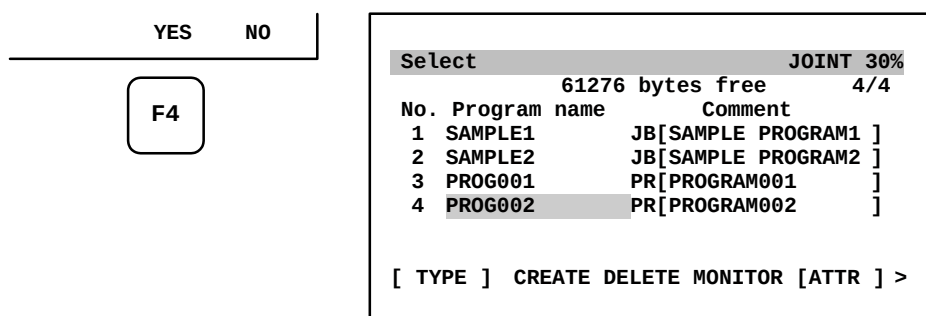
- 1 Press the MENUS key to display the screen menu.
 - 2 Select SELECT. The program list screen is displayed.
- The program list screen can also be displayed by pressing the SELECT key, instead of executing steps 1 and 2 above.



- 3 Move the cursor to the name of a program to be deleted, then press the F3 DELETE key.



- 4 Press the F4 (YES) key.
- 5 The specified program is deleted.



⚠ CAUTION

Once a program is deleted, the program cannot be restored. Make sure you delete only programs that you no longer want.

Procedure 5-31 Copying a program

Step

- 1 Press the MENUS key to display the screen menu.
- 2 Select SELECT. The program list screen is displayed.
- 3 Press F1 (copy) on the next page and then a program copy screen is displayed.

```

1 Words
2 Upper Case
3 Lower Case
4 Options --Insert--
PROGRAM COPY
--- Copy Teach Pendant Program ---
From:
SAMPLE3
To:
SAMPLE3
Old Value: SAMPLE3
RSR      PNS      STYLE      JOB      TEST

```

- 4 Enter the name of the program to be copied, then press the ENTER key.

```

PROGRAM COPY JOINT 30%
--- Copy Teach Pendant Program 1/1
From:
SAMPLE3
To:
PRG1
Copy OK ?
YES      NO

```

- 5 Press the F4 (YES) key.
- 6 The desired program is copied to the specified program, PRG1.

```

YES      NO
|
|
| F4
|

```

```

Select JOINT 30%
48956 bytes free 6/6
No. Program name Comment
1 SAMPLE1 JB[SAMPLE PROGRAM1 ]
2 SAMPLE2 JB[SAMPLE PROGRAM2 ]
3 SAMPLE3 JB[SAMPLE PROGRAM3 ]
4 PROG001 PR[PROGRAM001 ]
5 PROG002 PR[PROGRAM002 ]
6 PRG1 JB[SAMPLE PROGRAM3 ]
[ TYPE ] CREATE DELETE MONITOR [ATTR ] >

```

Procedure 5-32 Displaying the Attribute of the Program

Step

- 1 Press the MENUS key. The screen menu is displayed.
- 2 Select "0 — next —". "1 SELECT" in the next page is displayed.
You can select a program list screen by pressing the SELECT key instead of the above 1 to 2 procedure.

Program Selection Screen

Select		JOINT 30%
61276 bytes free		1/4
No.	Program name	Comment
1	SAMPLE1	JB[SAMPLE PROGRAM1]
2	SAMPLE2	JB[SAMPLE PROGRAM2]
3	PROG001	PR[PROGRAM001]
4	PROG002	PR[PROGRAM002]
[TYPE] CREATE DELETE MONITOR [ATTR] >		
COPY DETAIL LOAD SAVE PRINT >		

- 3 Press F5,[ATTR].
- 4 Select Size.
- 5 The number of lines and size of a program is displayed at the place that the comments are displayed.

- 1 Comment
- 2 Protection
- 3 Last Modified
- 4 Size
- 5 Copy Source
- 6 Name Only

ATTR

Select		JOINT 30%
61276 bytes free		1/4
No.	Program name	Size
1	SAMPLE1	JB[10/ 1000]
2	SAMPLE2	JB[10/ 1000]
3	PROG001	PR[10/ 1000]
4	PROG002	PR[10/ 1000]
[TYPE] CREATE DELETE MONITOR [ATTR] >		

F5

ENTER

- 6 When you want to display the other item, select the desired item in the procedure 4.

5.6 BACKGROUND EDITING

While the robot is being operated, the background editing function allows another program to be edited in the background. With this function, another program can be modified and checked without stopping robot operation, thus increasing productivity and maintenance efficiency.



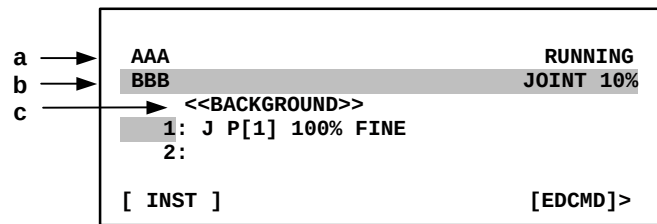
WARNING

This function allow editing when the teach pendant is disabled. However, when the teach pendant is disabled, any edit operations performed by an operator near the robot are very dangerous. To ensure operator safety, be sure to perform edit operation outside the robot movement range.

Outline of this function

This function is outlined below.

- Background editing is started by selecting a special program name for background editing when the teach pendant is disabled. The special program name is "-BCKEDT-".
- During background editing, the following data is displayed on the top of the edit screen of the teach pendant:
 - Program name selected in the background
 - <<BACKGROUND>> for indicating that background editing is in progress



a: Execution status of the program selected (status line)

b: Program name selected in the background

c: Indication that background editing state is set

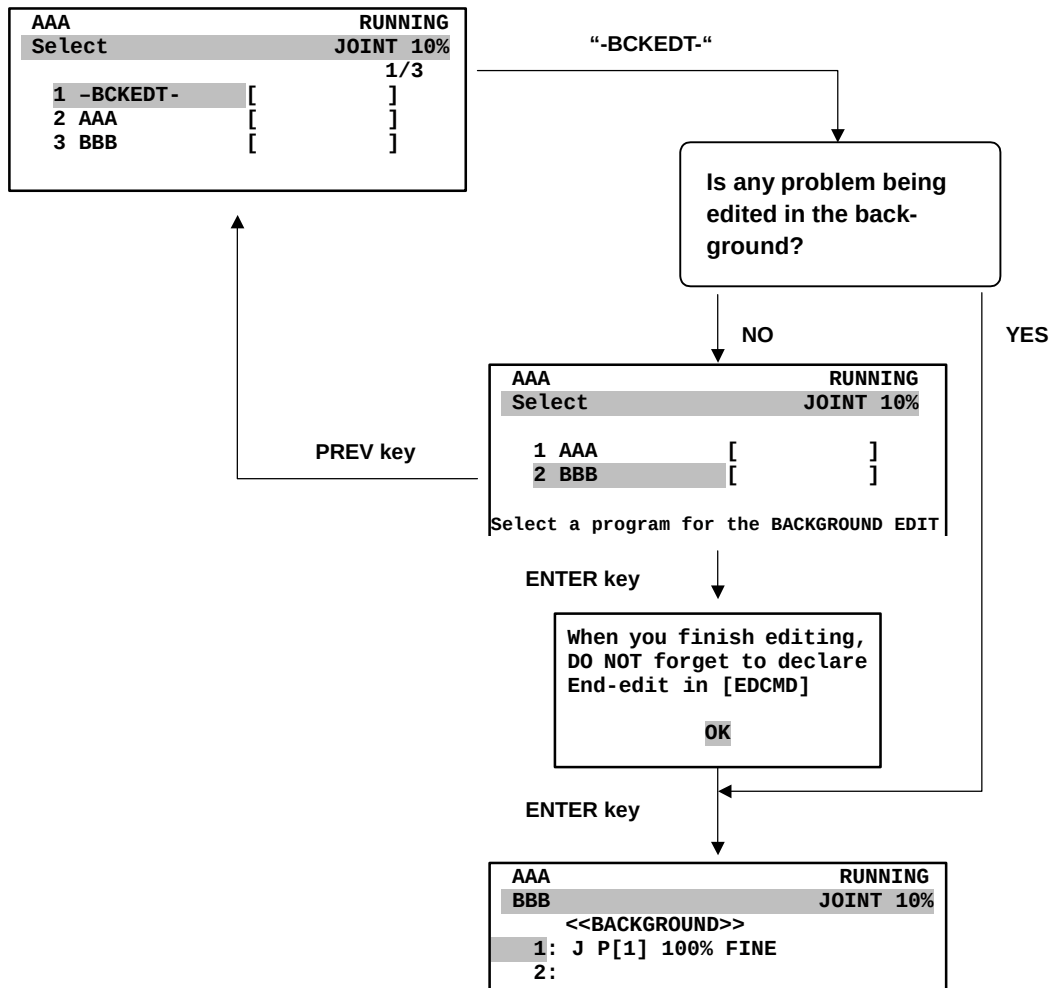
- No modifications to a program being edited in the background are reflected in the original program until the background editing is completed.
- To terminate background editing, press the F5 [EDCMD] key on the edit screen to display a menu, then select End_edit from the displayed menu.
Here, the user can choose whether to reflect the results of background editing in the original program or discard the results of background editing.
- No multiple programs can be edited in the background at a time. The background editing of a program must be terminated by End_edit operation before another program can be edited in the background.
- If another program is selected without performing End_edit operation during background editing, the results of background editing are preserved.
Background editing can be restarted by reselecting the special program name ("BCKEDT-") for background editing on the program list screen.
- When the teach pendant is disabled, and the edit screen is displayed, the user can switch between the display of the program selected in the foreground (not background) and the display of the preserved results of background editing.
- When the teach pendant is enabled, the special program name for background editing can be selected from the program list screen, and can be executed with the teach pendant.
- When the teach pendant is disabled, the special program for background editing cannot be externally selected and executed.
- When an external start signal is applied during background editing, the program selected in the foreground is started.
- The program started during automatic operation or executed by subprogram calling is the original program selected in the background.
- Even if a program is externally selected with the external program selection function (PNS) during background editing, the background editing can be continued without being interrupted.

The operation flows of the following cases are explained using figures below:

- When background editing is started with the teach pendant disabled
- When background editing is started with the teach pendant enabled
- When a program is externally selected during background editing
- When a start signal is externally applied during background editing
- When the teach pendant is enabled during background editing
- When the teach pendant is disabled during background editing
- When the screen is switched using the edit key on the teach pendant
- When background editing is terminated with the teach pendant disabled
- When background editing is terminated with the teach pendant enabled

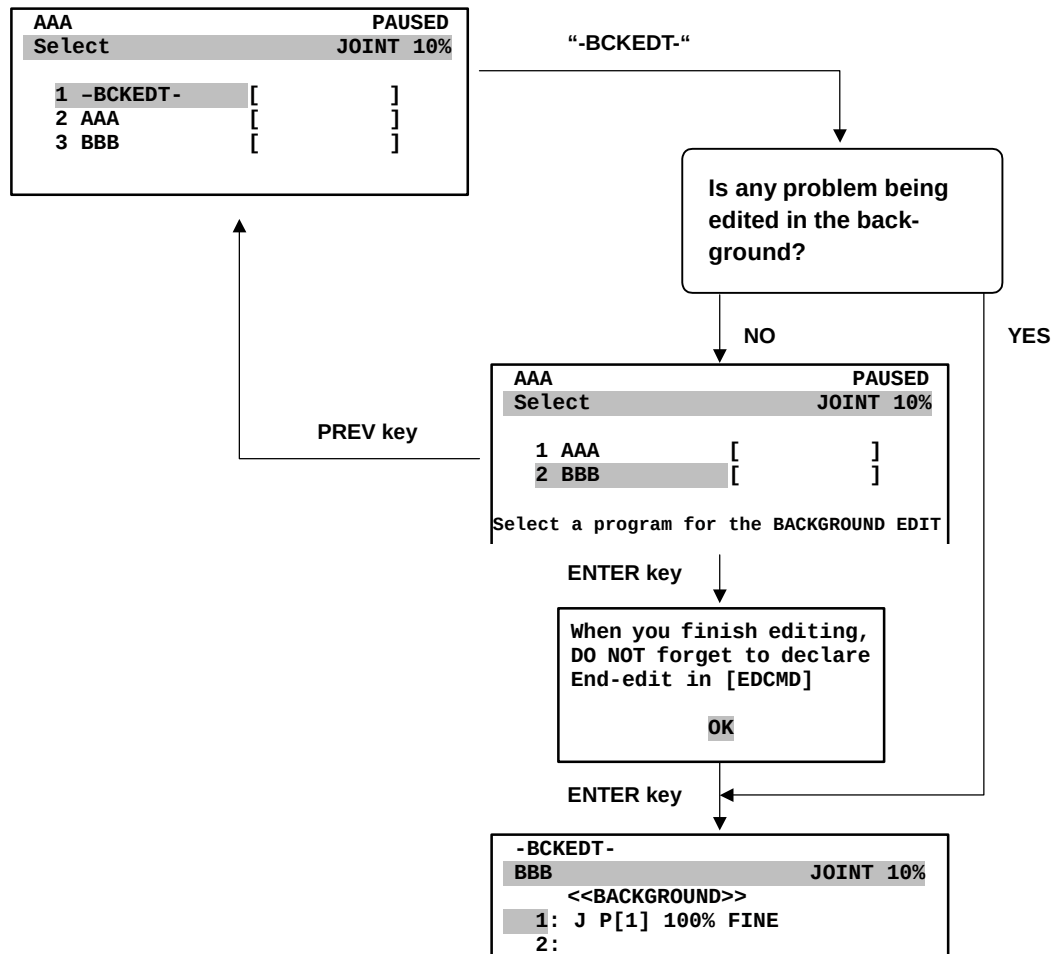
When background editing is started with the teach pendant disabled

When a program is selected in background editing, the program selected in the foreground is not modified. Even if no program is selected in the foreground, background editing is started.



When background editing is started with the teach pendant enabled

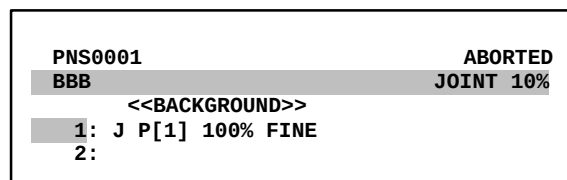
If the special program for background editing is selected when the teach pendant is enabled, the program is selected in the foreground, and its test execution is enabled.



When a program is externally selected during background editing

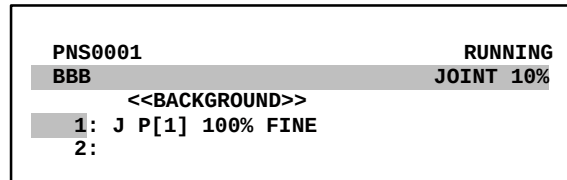
If a program is externally selected during background editing (with the teach pendant disabled), the status line displays the state of the selected program.

The state of background editing remains unchanged.



When a start signal is externally applied during background editing

If a start signal is externally applied during background editing (with the teach pendant disabled), the program selected in the foreground is started, and the status line displays RUNNING. The state of background editing remains unchanged.



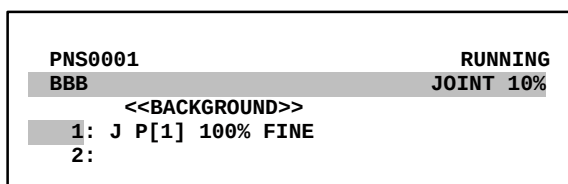
When the teach pendant is enabled during background editing

If a program is selected in the foreground, background editing and the program being executed are suspended, and the program selected in the foreground is displayed on the screen.

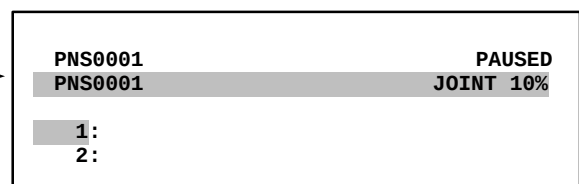
If an alarm is issued from the program being executed, for example, the point of alarm generation can be immediately located and corrected by enabling the teach pendant according to this function.

To return to background editing, disable the teach pendant, then press the edit key or reselect "-BCKEDT-" from the program list screen.

Teach Pendant: Disable



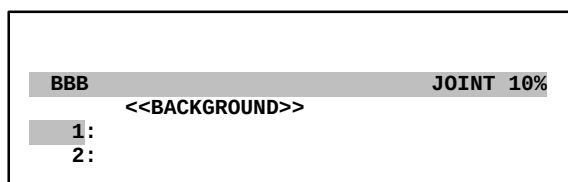
Teach Pendant: Enable



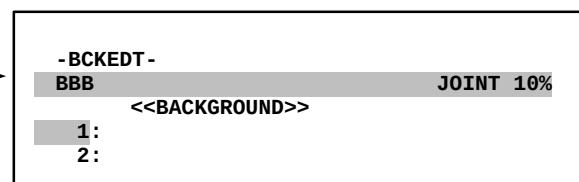
Disable the teach pendant, then
a. Press the EDIT key on the program edit screen.
b. Select "-BCKEDT-" on the program list screen.

If no program is selected in the foreground, the special program ("-BCKEDT-") is selected to allow the program being edited in the background to be executed. The status line displays the state of "-BCKEDT-".

Teach Pendant: Disable



Teach Pendant: Enable

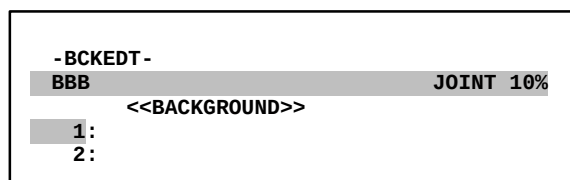


When the teach pendant is disabled during background editing

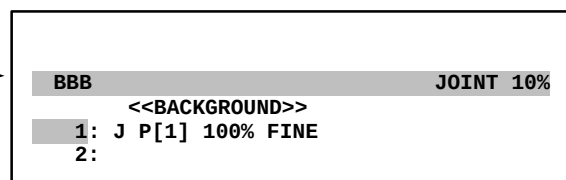
If "-BCKEDT-" is selected in the foreground, the foreground enters the program nonselection state when the teach pendant is disabled. (The status line disappears.) So, the program being edited in the background cannot be executed externally.

The background editing can be continued without modification.

Teach Pendant: Enable



Teach Pendant: Disable

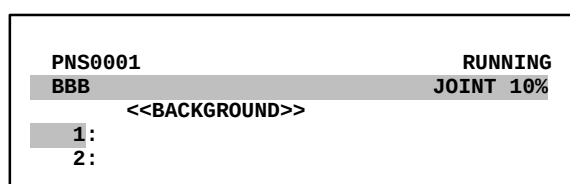


When the screen is switched using the edit key on the teach pendant

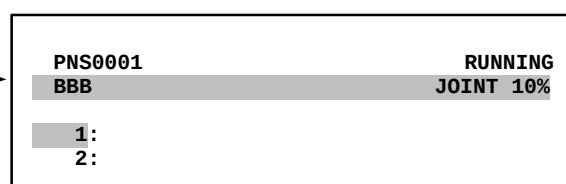
If the teach pendant is disabled, and the program edit screen is displayed, pressing the EDIT key switches screen display between the display of the program selected in the foreground and the display of suspended background editing.

If there is a program in the foreground and background as well, the screen display switches between foreground display and background display each time the edit key is pressed, as shown below.

Teach Pendant: Disable



Teach Pendant: Disable



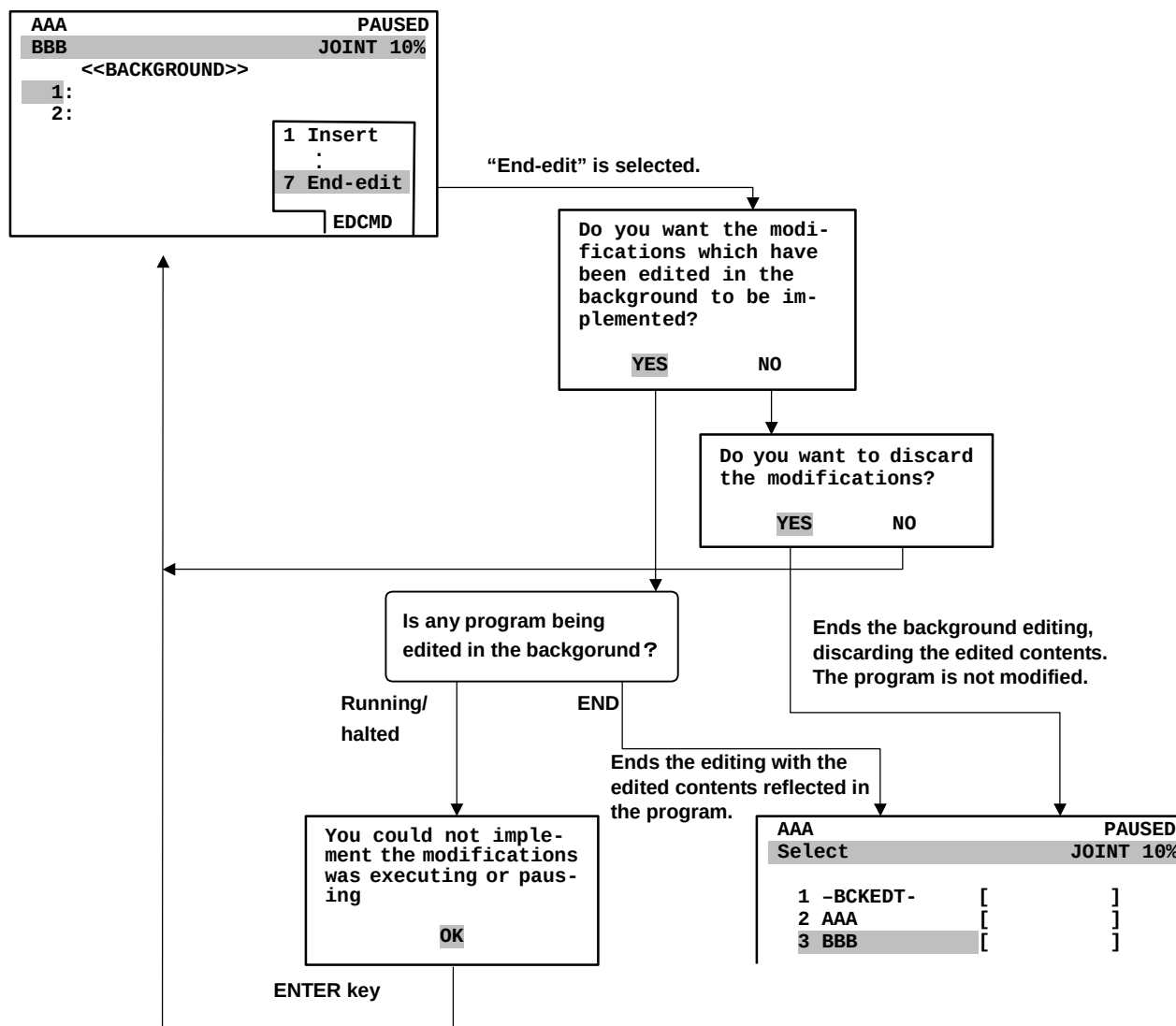
Edit
key

If no program is selected in the foreground, pressing the edit key does not switch screen display; the error Program is not selected occurs.

If no program is selected for background editing, pressing the edit key does not switch screen display; the error Not editing background program occurs.

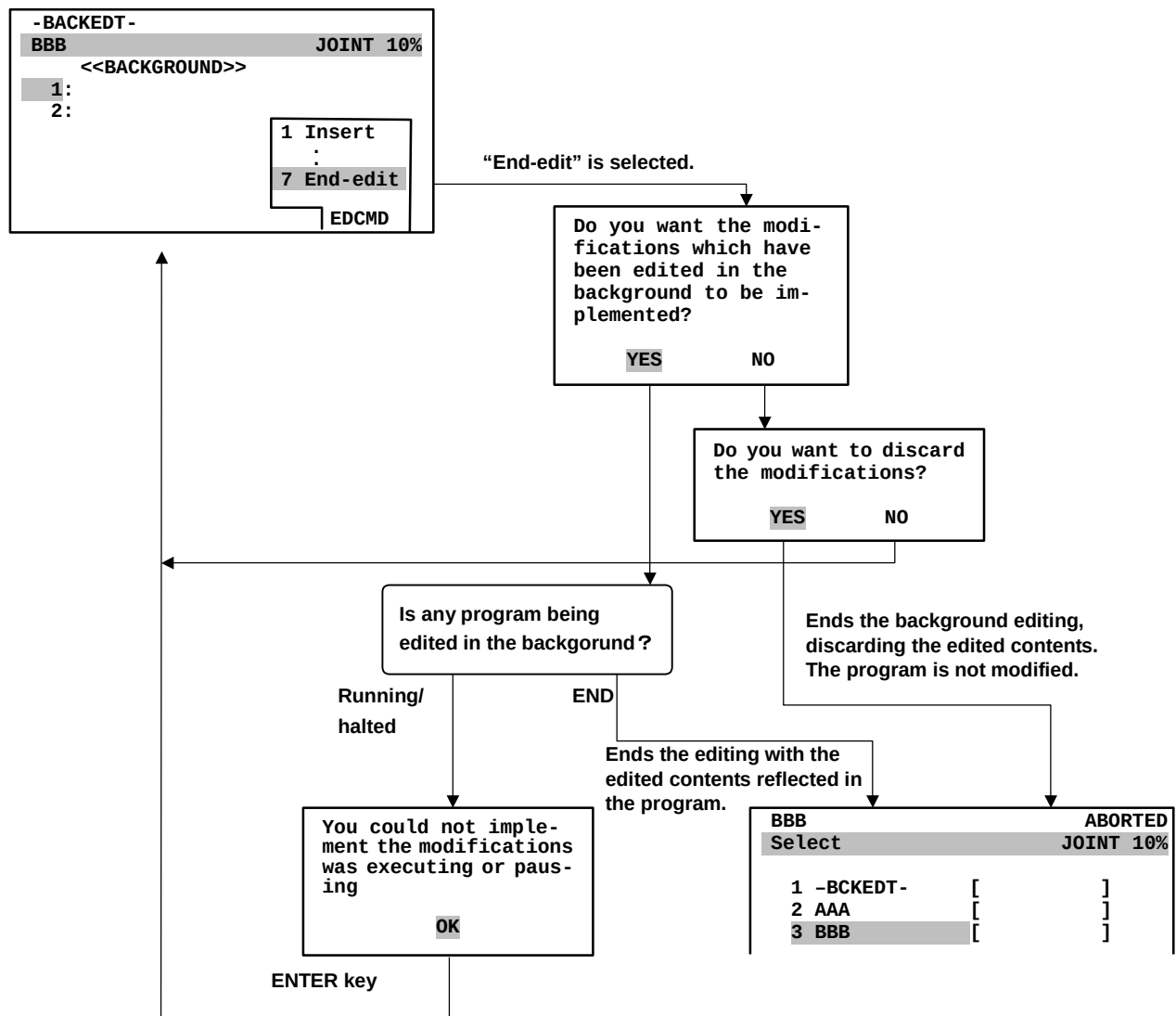
When background editing is terminated with the teach pendant disabled

When background editing is terminated, the program list screen appears. At this time, the user can specify whether to reflect the results of background editing in the original program.



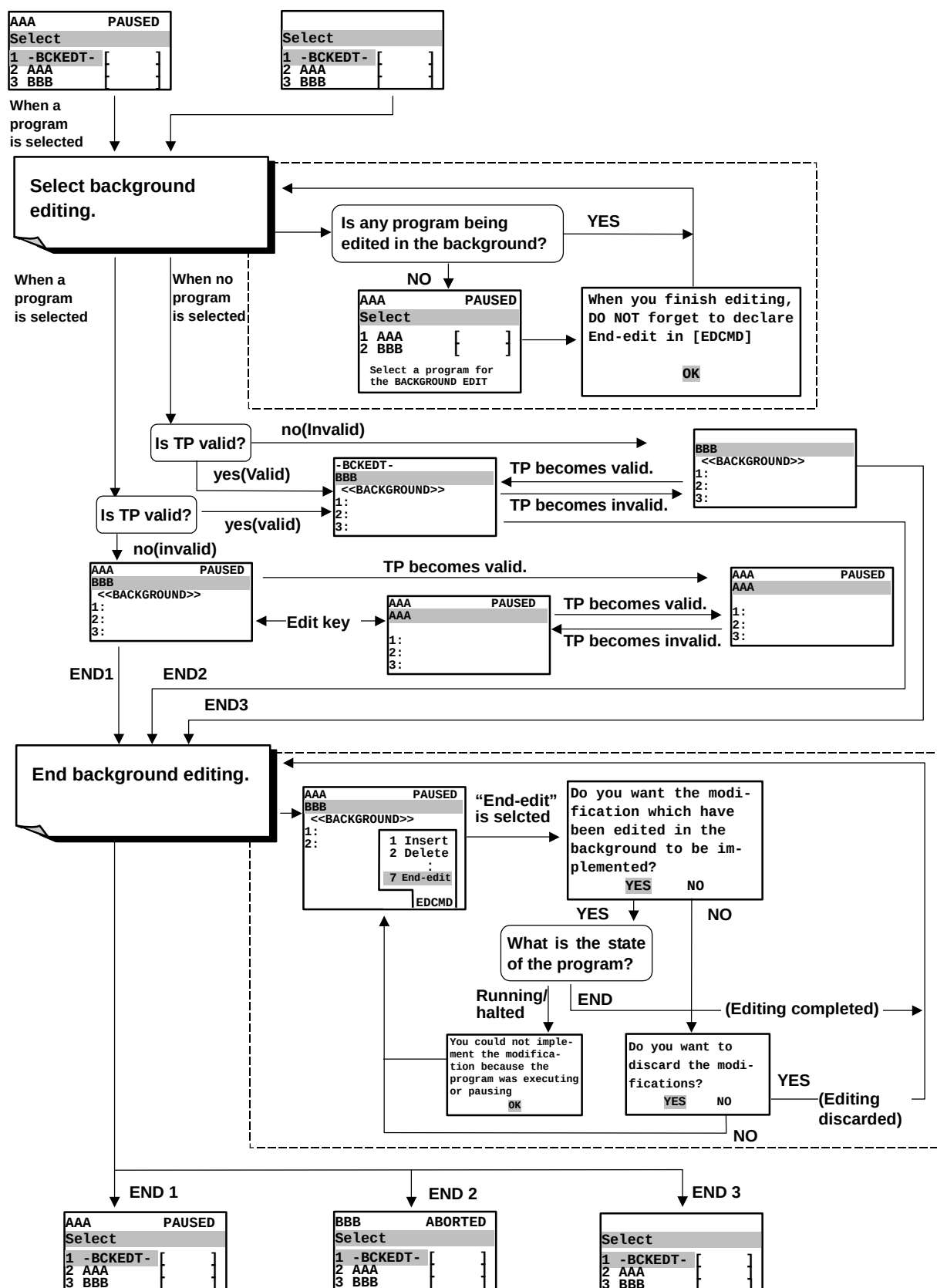
When background editing is terminated with the teach pendant enabled

When background editing is terminated, the program list screen appears. The program edited in the background is selected in the foreground, and the status line displays the state of the program.



Operation flow

The operation flow of this function is shown on the next page.



Notes

When using this function, note the points below.

- When a program is selected for background editing, the selected program is internally copied to the special program for background editing. So, memory larger than the size of a selected program needs to be allocated beforehand.
- When the background editing of a program is terminated, the original program is backed up, and the background program is reflected in the original program. So, memory larger than the size [(original program) + (increment produced by background editing)] needs to be allocated beforehand.
- If background editing cannot be terminated for a cause such as insufficient memory, the following error and its cause are displayed in the alarm display lines (line 2 and 3) on the teach pendant:

TPIF-054 Could not end editing

MEMO-126 No more available memory

- When the power to the robot is turned off then back on while background editing is being terminated (while the original program is being updated)

To prevent the updating of the original program from being stopped halfway, restore the original program from the backup program when the power is turned on. If the results of background editing need to be reflected, check the results of background editing, then perform another editing termination operation.

If an attempt to restore the original program fails, the following error is displayed:

TPIF-055 Could not recovery original program

In this case, check the results of background editing, then perform another editing termination operation.

If the power is turned off then back on when editing is terminated, check the state of the original program before starting continuous operation.

- If the original program is executed when background editing is terminated, the robot may stop, depending on the timing of the execution. When terminating background editing, carefully check that the original program is not executed.

Four cases can be considered for the timing relationship between background editing termination operation and program execution.

- Case 1: The program is being executed when background editing is terminated.

In this case, the message "You could not implement the modification because the program was executing or pausing" is displayed in the central part of the teach pendant, and the results of background editing cannot be reflected.

- Case 2: The program is started exactly when the results of background editing have been reflected

In this case, the program reflecting the results of background editing is executed.

- Case 3: An attempt is made to start the program while the results of background editing are being reflected

One of the following errors occurs, and the robot stops:

SYST-011 Failed to run task

MEMO-004 Specified program is in use

- Case 4: When the original program is deleted, and a program is re-created to reflect the results of background editing, an attempt is made to start the program.

One of the following error occurs, and the robot stops:

SYST-011 Failed to run task

MEMO-027 Specified line does not exist

- When the original program is write-protected (Write-protect is ON), editing cannot be terminated. In this case, one of the following error occurs:

TPIF-054 Could not end editing

TPIF-008 Memory protect violation

- Background editing can be terminated even when the special program for background editing is write-protected.

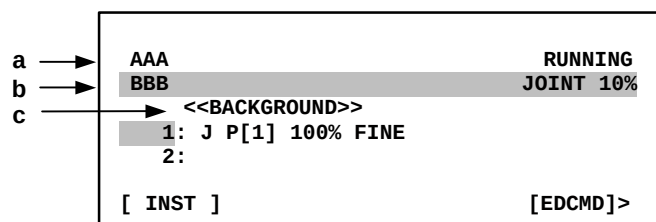
- The status line displays the execution state of a selected program. So, if a subprogram being executed is terminated forcibly, and the main program is selected in the foreground, the status line continues to display the subprogram name.
If program start operation is initiated here, the execution of the selected main program is started, and the status line displays the execution state of the main program.
If the disabled edit key or teach pendant is enabled on the background screen in the state above, the status line does not display the subprogram but the main program selected in the foreground.
- When the teach pendant is disabled, a program can be created/deleted. However, when a program is created, the following error occurs; no selection is made in the foreground, and no direct transition to the edit screen is made:
TPIF-104 Teach Pendant is disabled
- If the teach pendant is disabled after the special program for background editing is selected and executed with the teach pendant enabled, the end state is set.
If the teach pendant is disabled when a subprogram is executed from the special program, the execution is terminated, and the program list screen appears.
- When there is a suspended program in the background, the special program for background editing ("BCKEDT") cannot be read from the external memory device. In this case, the following message appears:
This program is being edited
Before reading the special program from the external memory device, terminate background editing.
- When using background editing function, the program is switched between selected and displayed on the edit screen depending on the status of the teach pendant enable/disable.
Notes about switching this selection are explained.
 - The teach pendant is enabled, and the program being edited in the background is selected from the program list screen, the paused program is aborted and the selected program is cleared.
So, the paused program cannot be restarted.
 - When background editing is continued after the teach pendant changed to enabled or when background editing is finished, this above operation is needed. So the paused program cannot be restarted.
 - When you need to restart the paused program, please continue the background editing or finish background editing with the teach pendant disabled.

The operation flow is explained as below.

The teach pendant is disabled (Program being executed)

During program AAA being executed, program BBB is edited in the background.

- 1) Select "-BACKEDT-", and select program BBB for editing.
- 2) The teach pendant screen is changed to the background editing screen of program BBB as below.



- a : Program name selected in the foreground.
- b : Program name selected in the background.
- c : Indication of background editing state.

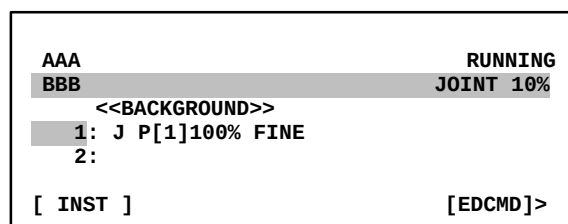
This state is that program AAA is selected in the foreground and program BBB is selected in the background.

Note the operation after the teach pendant is enabled from above state.

- 3) Enable the teach pendant. When program AAA is selected in the foreground and program BBB is selected in the background, if the teach pendant is enabled, the teach pendant edit screen is changed from program BBB being edited in the background to program AAA selected in the foreground automatically.

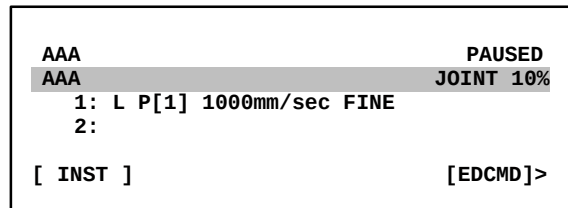
The purpose of this automatic changing screen is that program AAA being executed in the foreground can be operated by changing the teach pendant enabled even though the other program being edited in the background.

Screen of program BBB being edited in the background



The teach pendant is enabled.

Program AAA selected in the foreground is displayed automatically.



- 4) Next, select “-BCKEDT-” from the program list screen to continue the background editing of program BBB.

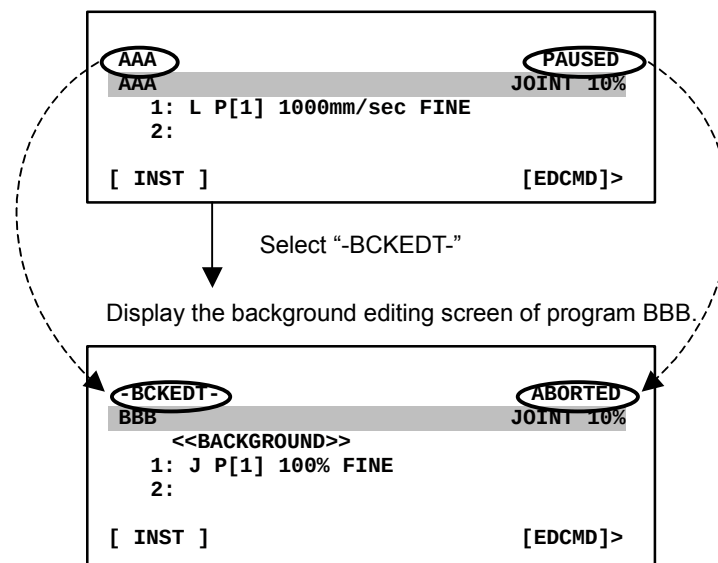
※At this moment, note the point below.

If select “-BCKEDT-” with the teach pendant enabled, the program in the foreground is changed from program AAA to program -BCKEDT-. That is to say, program AAA is changed to nonselected program at this point, and execution status is changed from paused to aborted at once. (This behavior is equal to select another program with being a paused program.)

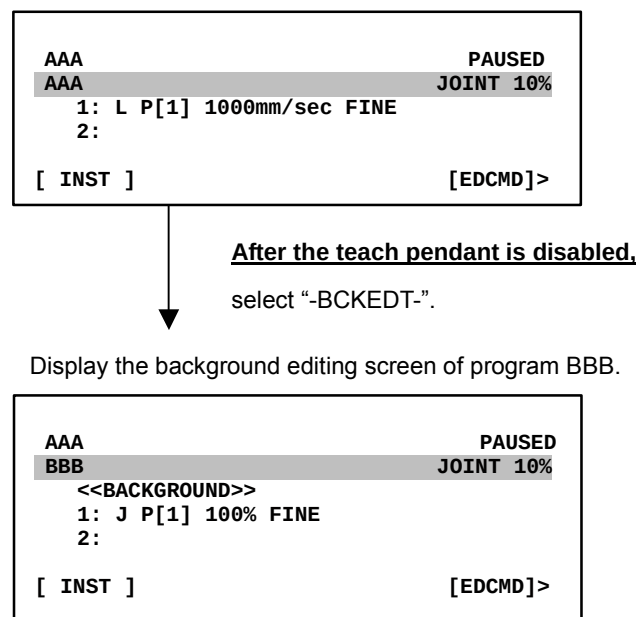
Therefore if background editing of program BBB is finished afterward, the edit screen of program AAA is not displayed.

Program AAA need to be selected from the program list screen to edit or execute it. However if program AAA is selected, it cannot be restarted at paused line.

The edit screen of program AAA selected in the foreground.



Please select “-BCKEDT-” after teach pendant is disabled to continue the background editing without program AAA is changed to nonselected program. In that case a program selected in the foreground remains program AAA as below.



5.7 SINGULAR POINT CHECK FUNCTION

If a move statement is taught, or a position modification is made based on Cartesian coordinate position data when the robot is positioned near a singular point, the robot may move with an attitude different from the taught attitude when the move statement is executed. (See Subsection 4.3.2.)

To prevent such trouble, the singular point check function checks to see if a taught position is a singular point when the position is taught. Then, the function teaches such a position according to joint type based on the user's choice.

Function

To enable this function, set the system variable \$MNSING_CHK to TRUE.

If a move statement is taught with SHIFT + POINT key or a position modification is made with SHIFT + TOUCH UP key when the robot is at a singular point, this function checks if the taught position is a singular point. This check is made when the following conditions are satisfied:

- The additional instructions do not include incremental instructions, position compensation instructions, and tool compensation instructions.
- The UF (user coordinate system number) of position data is 0.
- The registered position is Cartesian format.

If a check finds that the taught position is a singular point, the top two lines of the teach pendant display the following warning message:

TPIF-060 Can't record on cartesian (G:i)

MOTN-023 In singularity

i: Move group number at a singular point

At the same time, the following prompt message is displayed at the lower part of the teach pendant:

Record current position on joint

At this time, the function keys YES and NO are displayed. Select one of the two keys.

- YES: Records position data according to joint type.
- NO: Does not perform position teaching/modification.

The position data of a program that has multiple motion groups is checked for singular points in ascending order of group numbers.

If multiple groups are at singular points, a warning message and prompt message are displayed for each group.

Notes

This function is not applied to the teaching of palletizing representative stack points and path points.

5.8 OTHER EDITING FUNCTION

This chapter describes following functions.

- Auto position renumbering
- Fixed program name
- Filtered program list

5.8.1 Auto Position Renumbering

This function renumbers Position number automatically when:

- New motion statement is taught by F1(POINT).
- New position is taught by SHIFT+F5(TOUCHUP).
- Program line that includes position is deleted.
- Program line that includes position is pasted.

Function

This function is disabled in default setting.

To enable this function, set \$POS_EDIT.\$AUTO_RENUM2 to be TRUE. (Default: FALSE)

(ex) Teach new position by SHIFT + [POINT]

```

PNS0005                                JOINT 10 %
                                      3/6
1: J P[1] 100% FINE
2: J P[2] 100% FINE
3:
4: J P[3] 100% FINE
5: J P[4] 100% FINE
[End]

POINT                                TOUCHUP>

```

Case that this function is enabled:
Position number is renumbered automatically.

```

PNS0005                                JOINT 10 %
                                      3/6
1: J P[1] 100% FINE
2: J P[2] 100% FINE
3: J @P[3] 100% FINE
4: J P[4] 100% FINE
5: J P[5] 100% FINE
[End]

POINT                                TOUCHUP>

```

Case that this function is disabled (default):

```

PNS0005                                JOINT 10 %
                                      3/6
1: J P[1] 100% FINE
2: J P[2] 100% FINE
3: J @P[5] 100% FINE
4: J P[3] 100% FINE
5: J P[4] 100% FINE
[Eed]

POINT                                TOUCHUP>

```

5.8.2 Fixed Program Name

This function restricts the name of program. Only the program name that starts from registered word become valid. Registered word for the program can be customized in system configuration menu. (Refer 3.15 System Config Menu) The program that does not start from registered word cannot be created. [TPIF-038 Invalid char in program name] happens when the invalid program name is specified.

Function

This function is disabled in default setting.

To enable this function, set \$ PGINP_PGCHK to be TRUE. (Default: FALSE)

```

System/Config                          14/48
14 WAIT timeout:                       30.00 sec
15 RECEIVE timeout:                   30.00 sec
16 Return to top of program:          TRUE
17 Original program name(F1): [RSR   ]
18 Original program name(F2): [PNS   ]
19 Original program name(F3): [STYLE ]
20 Original program name(F4): [JOB   ]
21 Original program name(F5): [TEST  ]
22 Default logical command: <*DETAIL*>
23 Maximum of ACC instruction:      150

[ TYPE ]

```

In this setting, only the programs that begins from RSR, PNS, STYLE, JOB, TEST become valid.

5.8.3 Filtered Program List

This function filters program list that displayed in program list screen. Only the programs that start from registered words become displayed in program list screen.

5.8.3.1 The way to make original program name of program creation screen into initial strings

Registered word for the program can be customized in system configuration menu. (Refer 3.15 System Config Menu)

Function

This function is disabled in default setting.

To enable this function, set \$PGINP_FLTE to be 1 or 2. (Default: 0)

1. Press [TYPE] key in program list screen.

Case \$PGINP_FLTR = 1:

Program type and registered words are displayed as choices.

Select			1/22
No.	Program name	Comment	
1	All	[	]
2	TP Programs	[	]
3	Macro	MR	[Get PC Data
4	Cond		[SET AT HOME I-O
5	'RSR'		[Request PC Menu
6	'PNS'	MR	[Send PC Data
7	'STYLE'	MR	[Send PC Event
8	'JOB'	MR	[Send PC SysVar
9	'TEST'		
TYPE			CREATE DELETE MONITOR [ATTR]>

Case \$PGINP_FLTR = 2:

Registered words are displayed as choices

Select			1/22
No.	Program name	Comment	
1	-BCKEDT-	[	]
2	GETDATA	MR	[Get PC Data
			[SET AT HOME I-O
1	All	[	]
2	RSR	MR	[Request PC Menu
3	PNS	MR	[Send PC Data
4	STYLE	MR	[Send PC Event
5	JOB	MR	[Send PC SysVar
6	TEST		
TYPE			CREATE DELETE MONITOR [ATTR]>

2. Select "PNS" -> Only the programs that start from selected word are displayed.

Select			1/4
No.	Program name	Comment	
1	-BCKEDT-	[	]
2	PNS0001	[	]
3	PNS0002	[	]
4	PNS0003	[	]
[TYPE]			CREATE DELETE MONITOR [ATTR]>

5.8.3.2 The way to register over six initial strings

It is possible to register only five words to initial strings by "5.8.3.1 the way to make original program name in create program display into initial strings". But it is possible to register 30 words to initial strings by this way.

It is possible to use this way if system software version is 7DA7/09 or later.

Procedure 5-33 Filtering program list by registering over six initial strings

1. Set word that you want to register as initial string

- Select [Menu→'0.Next' →'6.System' →'Variables']. System variables menu is displayed.
- Input strings that you want to register as initial string to each \$PGINT_TYPE in system variables menu. For example, \$PGINP_TYPE[1]='ABC', \$PGINP_TYPE[2]='DEF', \$PGINP_TYPE[3]='GHI', \$PGINP_TYPE[4]='JKL', \$PGINP_TYPE[5]='MNO'.

2. Setup system variable (\$PGINT_FLTR).

- Set \$PGINT_FLTR in system variables menu.
- The value of \$PGINT_FLTR is 0 in default. In this case, only program types like "TP programs", "Macro" are displayed.
- Change \$PGINT_FLTR to 5 or 6 in system variables menu.
List of F1[TYPE] in select menu will be added word that you registered in \$PGINP_TYPE[].

Case \$PGINP_FLTR = 5:

Program type and registered words are displayed as choices.

Select			1/22
No.	Program name	Comment	
1	All	[	]
2	TP Programs	[	]
3	Macro	MR	[Get PC Data
4	Cond		[SET AT HOME I-O
5	'ABC'		[Request PC Menu
6	'DEF'	MR	[Send PC Data
7	'GHI'	MR	[Send PC Event
8	'JKL'	MR	[Send PC SysVar
9	'MNO'		
TYPE			CREATE DELETE MONITOR [ATTR]>

Case \$PGINP_FLTR = 6:

Registered words are displayed as choices

Select			1/22
No.	Program name	Comment	
1	-BCKEDT-	[	]
2	GETDATA	MR	[Get PC Data
			[SET AT HOME I-O
1	All	[	]
2	ABC	MR	[Request PC Menu
3	DEF	MR	[Send PC Data
4	GHI	MR	[Send PC Event
5	JKL	MR	[Send PC SysVar
6	MNO		
TYPE			CREATE DELETE MONITOR [ATTR]>

3. Select registered string from list of F1[TYPE] in select menu

In example, try to select "ABC". Only programs that have initial name 'ABC' will be displayed in select menu. It is possible to register 30 words to \$PGINP_TYPE.

Select			1/4
No.	Program name	Comment	
1	-BCKEDT-	[	]
2	ABC0001	[	]
3	ABC0002	[	]
4	ABC0003	[	]
[TYPE]			CREATE DELETE MONITOR [ATTR]>

6 EXECUTING A PROGRAM

This chapter describes testing a program and automatic operation.

Contents of this chapter

- 6.1 PROGRAM HALT AND RECOVERY
- 6.2 EXECUTING A PROGRAM
- 6.3 TESTING
- 6.4 MANUAL I/O CONTROL
- 6.5 OPERATING THE HAND MANUALLY
- 6.6 AUTOMATIC OPERATION
- 6.7 ONLINE POSITION MODIFICATION

6.1 PROGRAM HALT AND RECOVERY

Program halt refers to stopping a running program.

A program halt is caused by:

- An alarm occurring accidentally while the program is running.
- An intentional stop of a running program by the operator.

The operating robot stops in one of the following ways:

- Fast stop : The robot is quickly decelerated until it stops.
- Slow stop : The robot is slowly decelerates until it stops.

Program halt states are classified into two types:

- Forced termination (end): Display the termination status of a program execution. ABORTED is displayed on the screen of the teach pendant.

If the main program is terminated while a subprogram is being executed, information on return of control to the main program is lost.

SAMPLE1	LINE 7	ABORTED
SAMPLE1		JOINT 30%

- Halt (temporary stop): The execution of a program is stopped temporarily. PAUSED is displayed on the screen of the teach pendant.

The temporarily stopped program can be restarted. The subprogram called with a program call instruction returns control to the main program.

SAMPLE1	LINE 7	PAUSED
SAMPLE1		JOINT 30%

To start from another line in the same program or another program, abort a program to release the paused state.

There are following methods to halt a program intentionally:

- Press the emergency stop button on the teach pendant or the machine operator's panel or release the deadman switch.

Peripheral device I/O *IMSTP input

- Press the HOLD button on the teach pendant or use the input signal *HOLD of the peripheral I/O: These inputs halt the execution of the program.
- Select 1 ABORT(ALL) from the function menu. Peripheral device I/O *CSTOPI input. This method aborts the program.

6.1.1 Halt by an Emergency Stop and Recovery

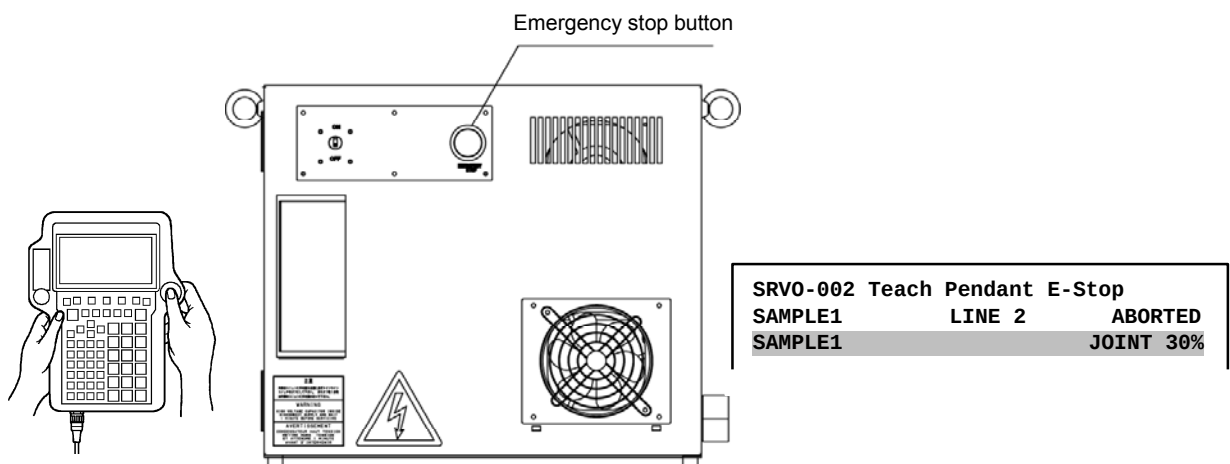
To stop the robot immediately (Please refer to "STOP TYPE OF ROBOT" in SAFETY for detail of stop type), press the emergency stop button on the machine operator's panel/box or teach pendant. When this is done an emergency stop alarm occurs.

Procedure 6-1 Emergency stop and recovery

Emergency stop procedure

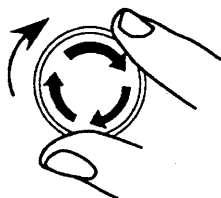
Step

- 1 Press the emergency stop button on the teach pendant or the machine operator's panel. This halts the running program, PAUSED is displayed on the teach pendant. The emergency stop button is locked to keep it pressed (on state). The emergency stop alarm message is displayed on the screen of the teach pendant.

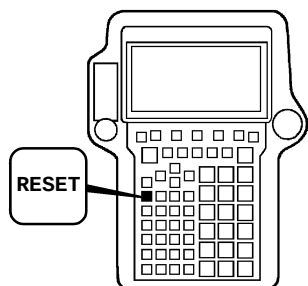


Recovery procedure

- 2 Eliminate the cause of the emergency stop. For example, correct the program.
- 3 Rotate the emergency stop button clockwise to unlock the button.



- 4 Press the RESET key on the teach pendant. The alarm message then disappears from the screen of the teach pendant.



6.1.2 Halt by a Hold and Recovery

To decelerate the robot slowly until it stops, press the HOLD key on the teach pendant or the operator's panel.

Pressing the HOLD key causes the following:

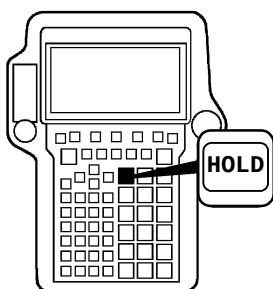
- The robot decelerates slowly until it stops (the program is halted).
- A setting can be made to cause an alarm to turn off the servo power. To make this setting, select SETUP General on the general item setting screen. (→ See Section 3.16, "SETTING THE GENERAL ITEMS".)

Procedure 6-2 Hold and recovery

Hold procedure

Step

- 1 Press the HOLD key on the teach pendant. The running program is halted, and PAUSED is displayed on the teach pendant.
The alarm message is only displayed when the halt alarm is enabled.



Recovery procedure

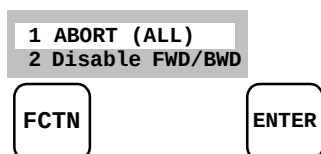
- 2 To release the halt state, restart the program.

Procedure 6-3 Terminating (aborting) a program forcibly

Abort a program

Step

- 1 To release the paused state and make a program aborted, press the function key to display the function menu.
- 2 Select ABORT(ALL). The program is aborted then the halt state is released.



6.1.3 Halt Caused by an Alarm

An alarm is issued when a failure is detected or when the emergency stop signal or another alarm signal is input from a peripheral device while the operator teaches or plays back a program. When an alarm is generated, it is indicated on the teach pendant, and processing such as robot operation and program execution is stopped to ensure safety.

Displaying an alarm

The operator can check whether an alarm has occurred by watching the FAULT lamps on the teach pendant and the first line and second line on the screen.

The kind of a alarm is recognized by a alarm code. The cause and corrective action of a alarm can be known by a alarm code. This manual does not explain the detail of alarm code. Refer to the Alarm Code List Operator's Manual to know the detail.

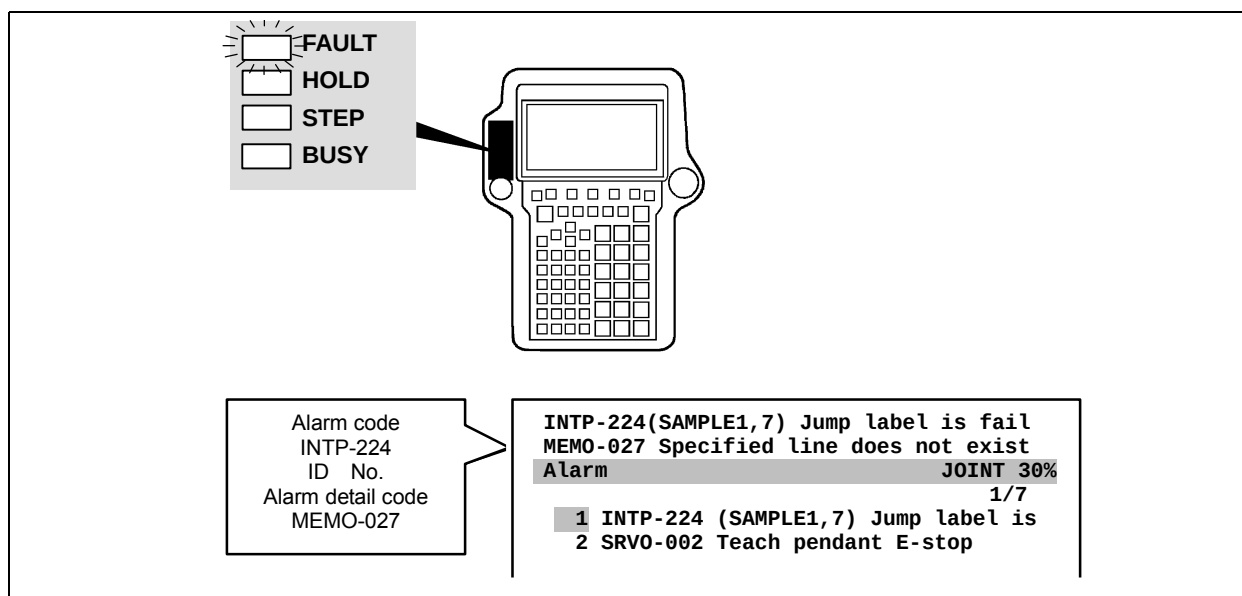
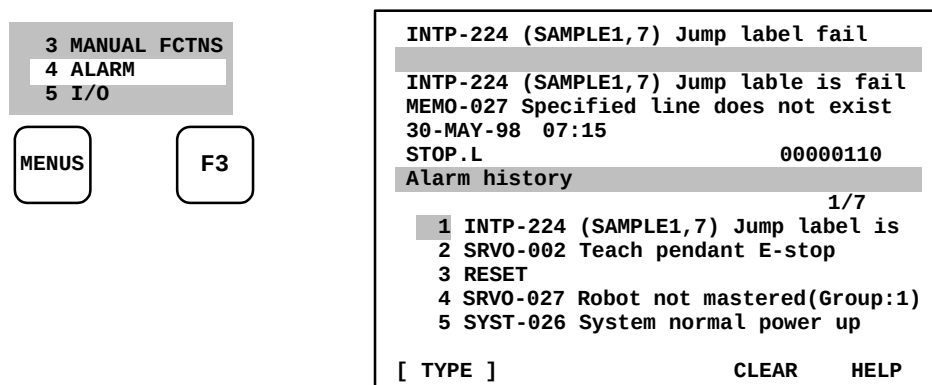


Fig. 6.1.3 (a) Display and Indication of an Alarm

Alarm history

To display the alarm history, select an alarm history screen [4 ALARM].

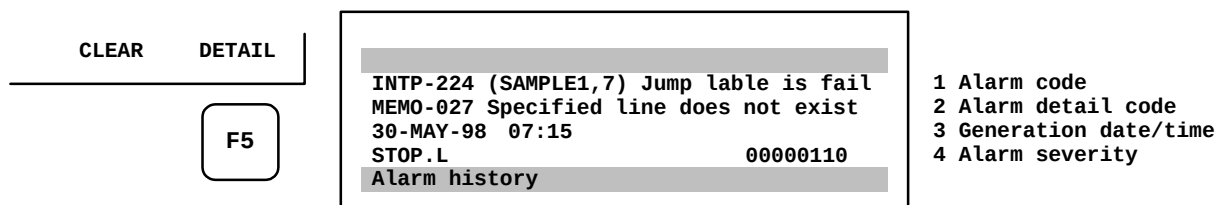


NOTE

The WARN alarm history is not recorded when system variable \$ER_NOHIS = 1.

Alarm detail information

Alarm has the detail information. To display the alarm detail information, press F5, HELP in the alarm history screen [4 ALARM].



- Alarm code: Identifies an alarm.
- Alarm detail code: Identifies an alarm detail.
- Generation date: The generation date of the alarm is indicated. (It is not supported currently.)
- Alarm severity: Indicates the severity of an alarm.

Resetting an alarm

After eliminating the cause of an alarm, press the RESET key to reset the alarm. The alarm indicated in the first and second lines of the teach pendant disappears. When the servo power is turned off, it is turned on. Resetting an alarm usually enables the robot.

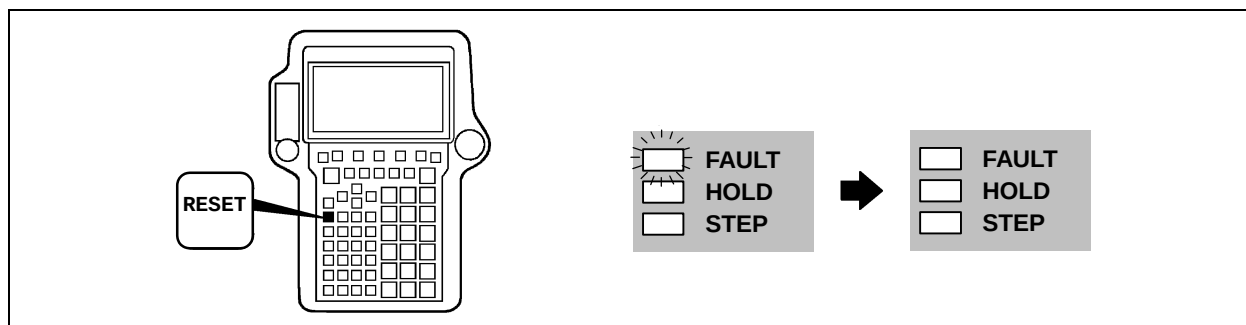


Fig. 6.1.3 (b) RESET Key

Disabling the output of peripheral I/O alarm signals

The output of alarm signals (FAULT output) can be disabled.

- Set \$ER_NO_ALM.\$NOALMENBLE to 1 (enabled).
- Specify the number of alarms for which output is to be disabled in \$ER_NO_ALM.\$NOALM_NUM.
- Specify the codes of the alarms for which output is to be disabled in \$ER_NO_ALM.\$ER_CODE1 to \$ER_NO_ALM.\$ER_CODE10.

11 002 (Meaning: SERVO-002 alarm)
Alarm ID Alarm No.

Halt alarm

The halt alarm function issues an alarm and turns off the power to the servo system when the operator presses the HOLD key to halt the robot.

Specify the fault alarm function in [6 SETUP General] on the general item setting screen (see Section 3.16, "SETTING THE GENERAL ITEMS").

Alarm severity

The alarm severity indicates the severity of an alarm and the cause of the alarm. Whether program execution and robot operation are stopped, and whether the servo power is turned off depend on the alarm severity.

Table 6.1.3 (a) Alarm Severity

	Program	Robot operation	Power to servo system	Range
NONE	none	none		—
WARN				—
PAUSE.L PAUSE.G	pause	decelerate the robot slowly until it stops	none	Local
STOP.L STOP.G				Global
SERVO		stop the robot immediately	off	Global
ABORT.L ABORT.G	abort	decelerate the robot slowly until it stops	none	Local
				Global
SERVO2 SYSTEM		stop the robot immediately	off	Global

Range	Indicates the range in which an alarm is issued when more than one program is executed (multitasking function).		
	Local	An alarm is issued only to the program that caused the alarm.	
	Global	An alarm is issued to all programs.	

NOTE

Some alarms do not observe the above rules.

Table 6.1.3 (b) Description of Alarm Severity

Severity	Description
WARN	A WARN alarm warns the operator of a comparatively minor or unimportant failure. The WARN alarm does not affect the operation of the robot. When a WARN alarm occurs, no corresponding LED on the teach pendant or the machine operator's panel lights. To prevent a possible failure in the future, action should be taken for this alarm.
PAUSE	When a PAUSE alarm occurs, the execution of the program is halted, and the operation of the robot is stopped. Appropriate action must be taken for the alarm before the program is restarted.
STOP	When a STOP alarm occurs, the execution of the program is halted, and the robot is decelerated until it is stopped. Appropriate action must be taken for the alarm before the program is restarted.
SERVO	When a SERVO alarm occurs, the execution of a program is paused(or aborted) and the power to the servo system is turned off to stop the robot immediately. The most common cause of a SERVO alarm is hardware failure.
ABORT	When an ABORT alarm occurs, the execution of the program is forcibly terminated, and the robot is decelerated until it is stopped.
SYSTEM	A SYSTEM alarm is issued when a major system failure occurs. When a SYSTEM alarm occurs, every robot in the system is disabled. Contact the FANUC Service Division. After taking appropriate action for the alarm, turn on the power again.

Sequence of program restart by peripheral I/O

After eliminating the cause of an alarm, the program is restarted by peripheral I/O as follows.

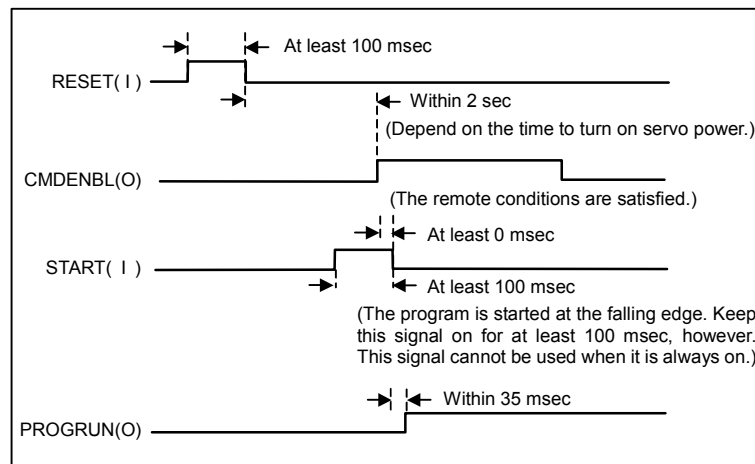


Fig. 6.1.3 (c) Sequence of program restart by peripheral I/O

6.2 EXECUTING A PROGRAM

To execute a program is to play back a taught program. A taught program is played back just like a recorded video tape is played back.

6.2.1 Starting a Program

A program can be started by:

- Using the teach pendant (SHIFT key and FWD or BWD key)
- Setting the START button on the operator's panel. : option panel only
- Using the peripheral device (RSR 1 to 8 input, PROD_START input, and START input)

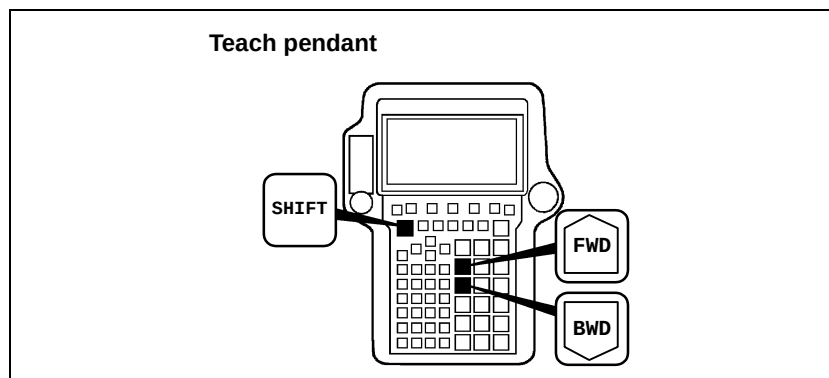


Fig. 6.2.1 (a) Starting a program

For safety's sake, a program can be started only in a device having motion control. Motion control can be switched by using the teach pendant enable switch on the operator's panel and the remote Local mode switch.

(For the remote local mode switch, see Section 3.15, "SYSTEM CONFIG MENU".)

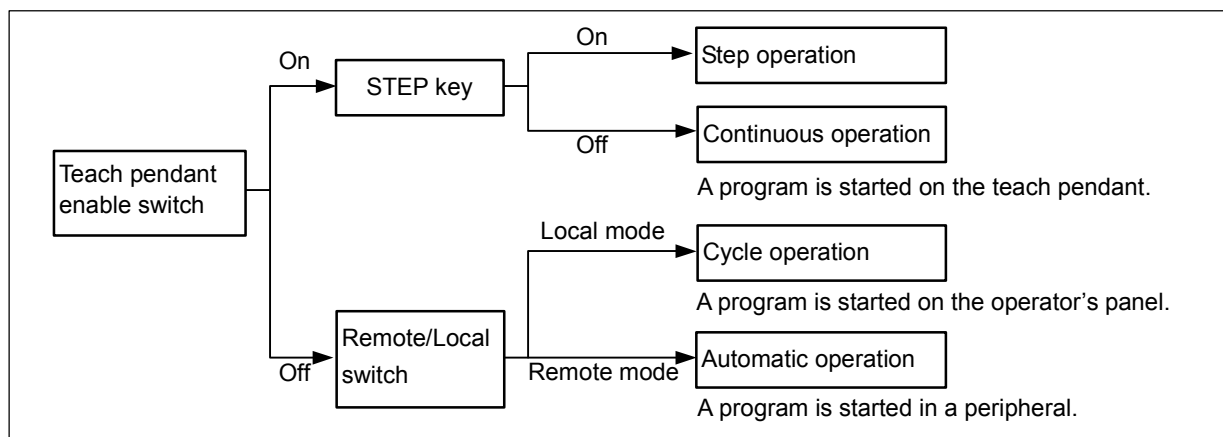


Fig. 6.2.1 (b) How to Set the Right to Start a Program

**CAUTION**

When the start right is switched by using the enable switch on the teach pendant or the remote/Local mode switch, any programs that are currently running are temporarily halted.

6.2.2 Robot Motion

The robot moves just as it is instructed by the motion instructions in the program. See Section 4.3, "MOTION INSTRUCTIONS".

The following factors determine the motion of the robot:

- Feedrate override: Robot motion speed (operating speed)
- Cartesian coordinate system: Work area where the robot moves

Feedrate override

The feedrate override determines the operating speed. The feedrate override is specified as a percentage of the feedrate specified in the program (programmed speed).

The current feedrate override is displayed in the upper right corner of the screen of the teach pendant, as shown in Fig. 6.2.2 (a).

Pressing the feedrate override key displays a popup window in reverse video in the upper right corner of the screen to call the operator's attention. The popup window in reverse video automatically disappears a few seconds later or after another key is pressed.

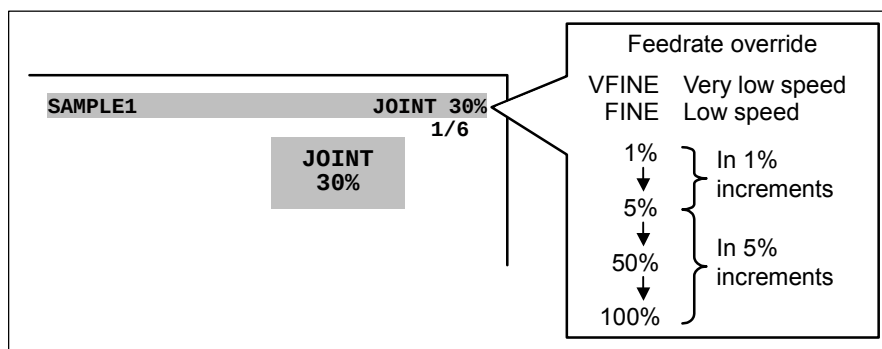


Fig. 6.2.2 (a) Screen Display for Feedrate Override

A feedrate override of 100% would cause the robot to operate at the maximum speed specified in the current setting. Table 6.2.2 shows the change in feedrate override when the override key is pressed.

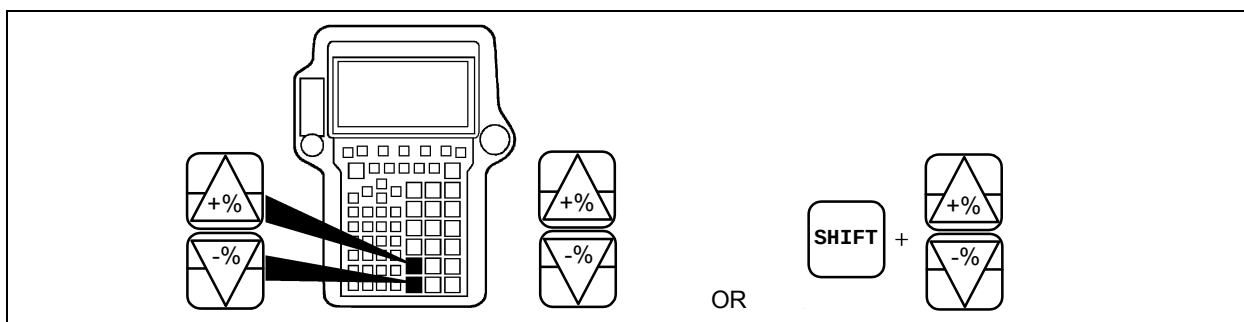
Table 6.2.2 Feedrate Override

When the override key is pressed	VFINE → FINE → 1% $\xrightarrow{\text{In 1\% increments}}$ 5% $\xrightarrow{\text{In 5\% increments}}$ 50% → 100%
When the override key is pressed while pressing the SHIFT key(*1)	VFINE → FINE → 5% → 50% → 100%

*1 Enabled only when \$SHFTOV_ENB is 1

To change the feedrate override, press the override key. Whenever the override key is pressed while the SHIFT key is pressed, the feedrate is changed in the order: VFINE, FINE, 5%, 50%, 100%. However, the feedrate is changed in this way only when system variable \$SHFT OV_ENB = 1.

Note that FINE and VFINE are enabled only during a jog feed. When FINE or VFINE is specified, the robot moves at a feedrate override of 1%.

**Fig. 6.2.2 (b) Override Keys**

A feedrate override must be determined according to the condition of the machining cell, type of robot motion, and the skill of the operator. Therefore, an inexperienced robot operator should use a low feedrate override.

The feedrate override can only be increased up to the maximum value specified in \$SCR.\$RUNOVLIM.

When the safety speed signal (*SFSPD input) (→ see Section 3.3) is turned off, the speed override value falls to the \$SCR.\$FENCEOVRD value. In this case, the speed override can be increased only up to the upper limit specified in \$SCR.\$SFRUNOVLIM.

The system provides a function for allowing the original speed override to be restored when the safety fence is closed. (→ See Section 3.17.)

Operating speed

The operating speed is the speed at which the robot moves while the program is played back. The operating speed is obtained from the following expressions:

Operating speed (joint control motion) (deg/sec) = $\text{Maximum joint feedrate} \times \frac{\text{Programmed feedrate}}{100} \times \frac{\text{Feedrate override}}{100}$ Operating speed (motion under path control) (mm/sec) = $\text{Programmed feedrate} \times \frac{\text{Feedrate override}}{100}$ Operating speed (motion under attitude control) (deg/sec) = $\text{Programmed feedrate} \times \frac{\text{Feedrate override}}{100}$

Fig. 6.2.2 (c) Operating Speed

Checking a Cartesian coordinate system

When position data is played back according to Cartesian coordinates, the coordinate system number of the Cartesian coordinate system to be used is checked.

When one of the coordinate system numbers 0 to 9 is specified and the specified coordinate system number does not agree with the currently selected coordinate system number, the program is not executed.

The coordinate system number is specified for position data when the position is taught. To change a written coordinate system number, use the tool change function/coordinate system change function.

- Tool coordinate system number (UT)

The number of a mechanical interface coordinate system or tool coordinate system is specified as a tool coordinate system number (UT). This number determines the tool coordinate system.

- 0 : The mechanical interface coordinate system is used.
- 1 to 9 : The tool coordinate system having the specified tool coordinate system is used.
- F : The coordinate system having the currently selected tool coordinate system number is used.

- User coordinate system number (UF)

The number of a world coordinate system or user coordinate system is specified as a user coordinate system number (UF). This number determines the coordinate system for the work area.

- 0 : The world coordinate system is used.
- 1 to 9 : The user coordinate system having the specified user coordinate system is used.
- F : The coordinate system having the currently selected user coordinate system number is used.

Position data information

Pressing the F5, [DETAIL] key displays position data information.

Position Detail				JOINT 30%	
P[1]	GP:1	UF:1	UT:1	CONF:	NUT 000
X:	1500.374	mm	W:	40.000	deg
Y:	-242.992	mm	P:	10.000	deg
Z:	956.895	mm	R:	20.000	deg
EDCMD					

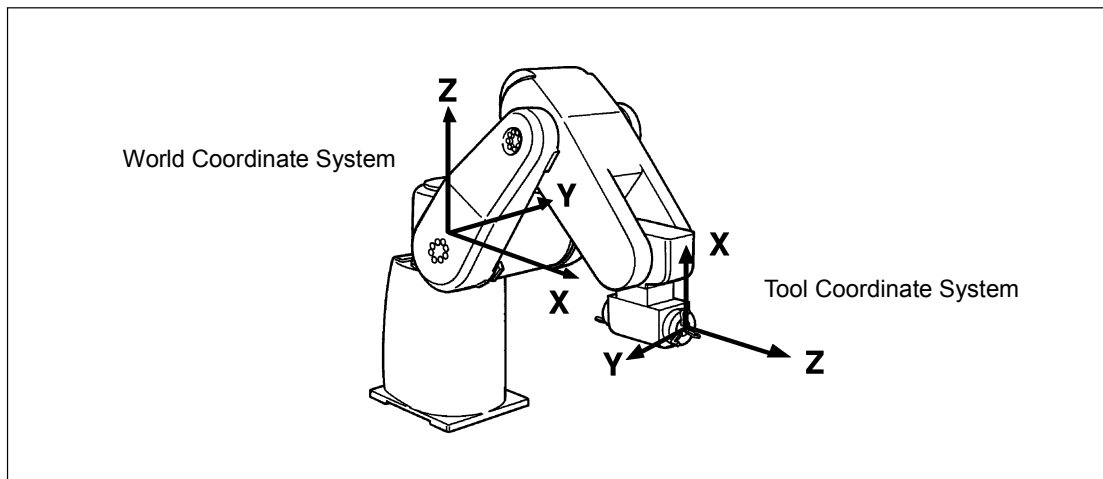


Fig. 6.2.2 (d) Selecting a Tool Coordinate System and User Coordinate System

6.2.3 Resuming a Program

Resuming a program means to restart a halted program. Before a program is halted, the system records the program. As a result, the following is possible:

- Control can be passed to the main program called with the program call instruction.
- The path for a circular motion can be reproduced.

Path for circular motion

In circular motion, the robot moves from the current position to the target point along the path that passes through the passing point. After the robot motion is interrupted by program halt, the robot is moved by jog feed, and the program is resumed. In this case, the robot moves along a path that is similar to the one that was specified before the program was halted.

(The path for a circular motion is recalculated on the assumption that the pass point is the current position after jogging, and that the start point is that used before the interruption.)

When a step test halted at the end of a circular motion is resumed after jog feed, the tool is returned to the end point of the circular motion, by means of a linear motion. (For a step test, see Subsection 6.3.2.) The motion is executed at the travel speed specified in the circular motion instruction.

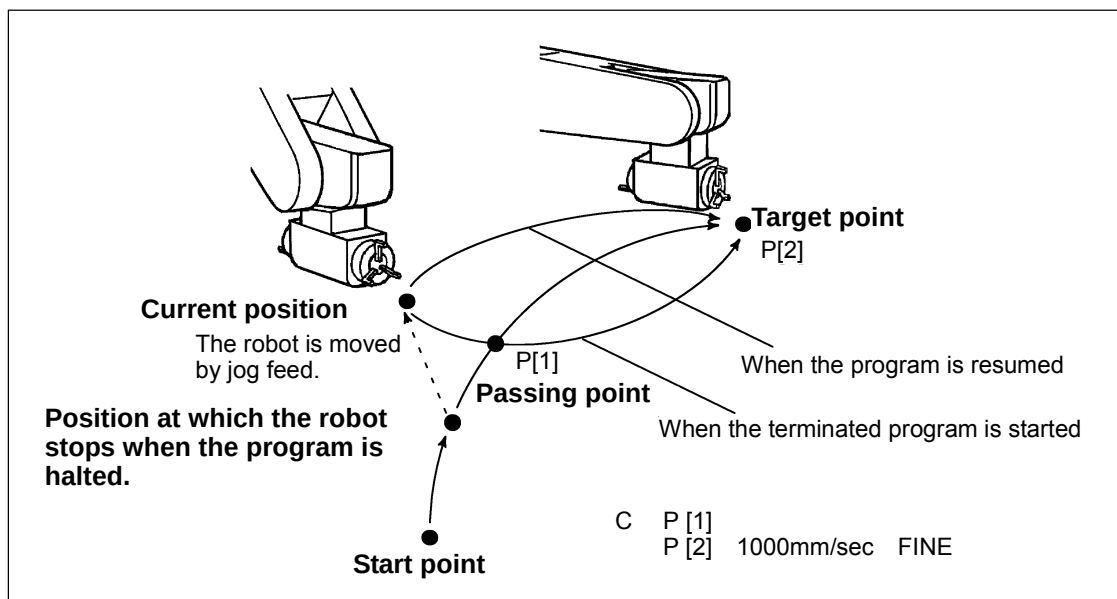


Fig. 6.2.3 Path for a Circular Motion

Releasing the halt state

The halt state of the program is released when:

- 1 PROGRAM ABORT is selected from the function menu.
- Switching of the start right (→ See Subsection 6.2.1.)
- Creating new program when the teach pendant is enabled. For program creation, see Section 5.3.
- Selecting another program when the teach pendant is enabled. For program selection, see Subsection 5.4.1.

Moving the cursor in the halt state

When the cursor is moved to a desired line in the halted program and the program is to be resumed, the system asks the operator whether the program is to be resumed at the line to which the cursor has been moved. When YES is selected in response to this message, the program is halted at the line to which the cursor has been moved. When NO is selected, the cursor is returned to the line it was at before it was moved (original line), then the program is halted at that line. For both YES and NO, when the program is resumed, program execution starts at the line to which the cursor has been moved.

Procedure 6-4 Releasing the halt state

Condition

- The program must be halted. (PAUSED is displayed on the screen.)

SAMPLE1	LINE 2	PAUSED
SAMPLE1		JOINT 30%

Step

- 1 Press the FCTN key to display the function menu.
- 2 Select 1 PROGRAM ABORT. The program is terminated. (ABORTED is displayed on the screen.)

1 ABORT (ALL)	2
FCTN	

SAMPLE1	LINE 2	ABORTED
SAMPLE1		JOINT 30%

Procedure 6-5 Moving the cursor in the halt state**Condition**

- The program must be halted. (PAUSED is displayed on the screen.)

SAMPLE1	LINE 2	PAUSED
SAMPLE1		JOINT 30%

Step

- 1 Move the cursor to the line where the program is to be resumed.
- 2 Restart the program.
The system asks the operator whether the program is to be resumed at the line to which the cursor has been moved.

```

3: L P[3]
4: L P[4]
5: J P[1]
[End]

```

The cursor is on a different line from where the program PAUSED[2].
Are you sure you want to run from this line ?

YES NO

- 3 Select YES to resume the program at the line to which the cursor has been moved. This line is then specified as the current line.

YES NO

SAMPLE1	LINE 4	PAUSED
SAMPLE1		JOINT 30%
4/6		
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]		

Select NO to resume the program at the line the cursor was at before it was moved (original line). The cursor is then returned to the original line.

YES	NO

Restart position check function

When a program is restarted in AUTO mode, this function compares the current robot position with the robot position present when the program was halted. If the comparison shows that the difference in position is beyond a set tolerance, the function issues a warning not to start the program.

If a warning is issued, select the restart method from the choices listed below. Make a choice with the teach pendant.

(1) Restart the program with no special action.

(2) Change the mode and return the robot to the stop position, then restart the program.

When restarting the program, on the restart position check screen of the setting menu, set the tolerable distance between the current robot position and the position at which the robot was halted.

SETUP RESUME TOL.		JOINT 30%
		1/6
1 Group		:1
2 Enable Tolerance checking		:TRUE
3 Distance Tolerance(mm)		[250.0]
4 Orientation Tolerance(deg)		[20.0]
Axes Tolerance		
5 Rotary Axes(deg)		[20.0]
6 Translation Axes(mm)		[250.0]
[TYPE]		

- Group
For each group, you can enable or disable the restart position check function and set tolerances. Set a target group number for setting. When the restart position check function is enabled for more than one group, a warning is issued if a tolerance of one group is exceeded.
- Enabling/disabling tolerance check
To enable the restart position check function, select YES. (The default setting is YES.)
- Tolerable distance (mm)
At program restart, when the difference in distance between the current robot position and the position at which the robot was halted is greater than the value set here, a warning is issued, and the program is not started.
- Tolerable attitude (deg)
At program restart, when the difference in joint angle between the current robot position and the position at which the robot was halted is greater than the value set here, a warning is issued, and the program is not started.
- Tolerance for axes: Rotation axis (deg)
When the difference in angle between the current position of a rotation axis in the robot and the position at which the robot was halted is greater than the value set here at program restart, a warning is issued, and the program is not started.
- Tolerance for axes: Linear axis (mm)
When the difference between the current position of a linear axis in the robot and the position at which the robot was halted is greater than the value set here at program restart, a warning is issued, and the program is not started.

When a program is restarted, this function compares the current robot position with the position at which the robot was halted. If the comparison shows that any of the distance, attitude, and axis position data exceeds a tolerance, a warning is issued, and the program is not started. In this case, the following message appears on the teach pendant:

The robot position is out of stop tolerance.
Please select action.
Choosing CONTINUE will require cycle start.

STOP CONTINUE

(1) When STOP is selected

When “STOP” is selected, this pop-up menu is disappeared, and the program is still paused. After select “STOP”, if start signal input, the tolerance check is executed and the pop-up menu is appeared again.

To resume the program, please move the robot to the position within the tolerance by jog feed, then input start signal.

(2) When CONTINUE is selected

The popup menu disappears, and the program remains halted. When the start signal is input under these circumstances, the program is started.

If jog feed is performed after CONTINUE is selected, checking is made again when the program is restarted next.



CAUTION

This function cannot be used with the tracking function at the same time.

In case that the multiple program selection is enabled, while the message to select the action is displayed on teach pendant for one of the selected programs, other selected programs cannot be started.

6.3 TESTING

Testing refers to checking the operation of the robot alone before automatically operating the robot in the site line. Testing the program is very important. It must be done to ensure the safety of the workers and the peripheral devices.

The following two methods can be used for testing:

- Step test: Execute the program line by line using the teach pendant or operator’s panel.
- Continuous test: Execute the program from the current program line to the end of the program (up to the end-of-program symbol or program end instruction) using the teach pendant or operator’s panel.

The teach pendant must be enabled before testing is performed using the teach pendant. The teach pendant is enabled when.

- The teach pendant enable switch is on.

Before test operation can be started from the operator’s panel/box, the operator’s panel must be in the enabled state. The operator’s panel can be placed in this state provided the following conditions are satisfied:

- The enable switch on the teach pendant is set to OFF.
- The mode is local.
- The peripheral device I/O *SFSPD input is on.

Before starting a program containing motion instructions, the following operation conditions must be satisfied:

- The input signal ENBL for the peripheral I/O must be on.
- An alarm must not be occurring

The typical test procedure is as follows:

- 1 Turn on the machine lock switch, perform step operation, and check program instructions and I/O.
- 2 Turn off the machine lock, and perform step operation from the teach pendant to check the robot operation, program instructions, I/O, and so forth.

- 3 Perform continuous operation at low speed.
- 4 Perform continuous operation at high speed and check the position of the robot and the operation timing.

6.3.1 Specifying Test Execution

To specify test execution is to specify the requirements for test execution of a program.

TEST CYCLE Setup		JOINT 30%
		1/4
Group:1		
1 Group Motion:		ENABLE
2 Digital/Analog I/O:		ENABLE
3 Step statement type:		STATEMENT
4 Step path node:		OFF
[TYPE] GROUP DISABLE ENABLE		

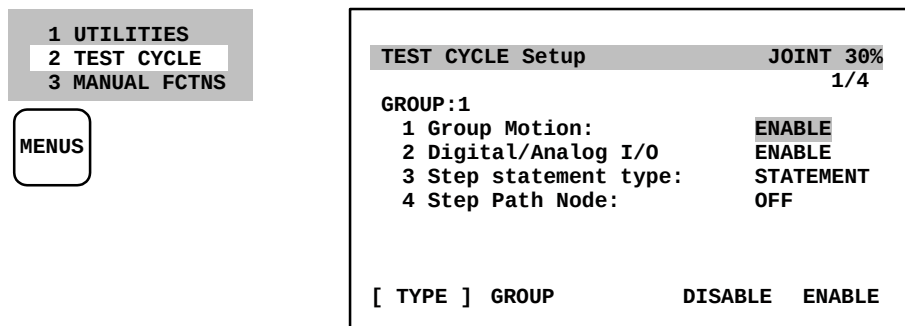
Table 6.3.1 Setting of test execution

Items	Descriptions
Group motion	This function specifies whether the robot is disabled. - DISABLE : The robot is disabled; it ignores all motion instructions. (Robot is locked.) - ENABLE : The robot is enabled, it usually accepts motion instructions. When the robot lock function is ON, the power to the servo system is assumed to be on. Pressing the RESET key resets all the servo alarms. NOTE Even when the robot lock is ON, the robot can not be operated when the emergency stop button is pressed.
Digital/Analog I/O	Digital/Analog I/O specifies whether to communicate with a peripheral device via digital I/O , analog I/O, group I/O and robot I/O signal lines or not. When this is set to disable, the robot does not send or receive the digital I/O and the analog I/O signal with a peripheral device. Internally, all the I/O signals are given the simulated flag(S) and the simulated flag can not be released until the setting is set to enable. (See Section 6.4, "MANUAL I/O CONTROL") When you set the disable flag, the output to the peripheral device does not change. You can simulate the output without changing the state of the peripheral device. When you set the flag to enable, the output returns to the state it was in before the disable flag was set. Control of the peripheral device returns to the controller. When you set the disable flag, the input from the peripheral device to the controller is retained by the controller. When you set the flag to enable the input returns to the state it was in before the disable flag was set.
Step statement type	Step statement type specifies how to execute a program in single step mode. - STATEMENT : The program execution is paused at each line. - MOTION : The program execution is paused at every motion instruction. - ROUTINE : Almost the same as STATEMENT, however, the pause is not done in a program that is called by a CALL instruction. - TP & MOTION: At all KAREL instruction except for motion instructions, a program does not pause. NOTE "TP & MOTION" is not used currently.
Step path node	When "Step path node" is set to be ON, the robot pauses at every node during execution of the KAREL instruction, "MOVE ALONG".

Procedure 6-6 Specifying test execution

Step

- 1 Press the MENUS key to display the screen menu.
- 2 Select 2 TEST CYCLE. The test cycle screen is displayed.



- 3 Specify requirements for test execution.
- 4 To change the group number, press F2 GROUP.

6.3.2 Step Test

To perform a step test (step operation) is to execute the program line by line. After one line of the program is executed, the program is halted. After executing a logic instruction, the next line becomes the current line and the cursor moves to the next line, but for the motion instruction, the cursor stays at the line that execution is completed.

Specifying the step mode (single step)

To specify the step mode, press the STEP key on the teach pendant. When the step mode is specified, the STEP LED on the teach pendant is lit. The STEP LED is off when continuous operation is specified.

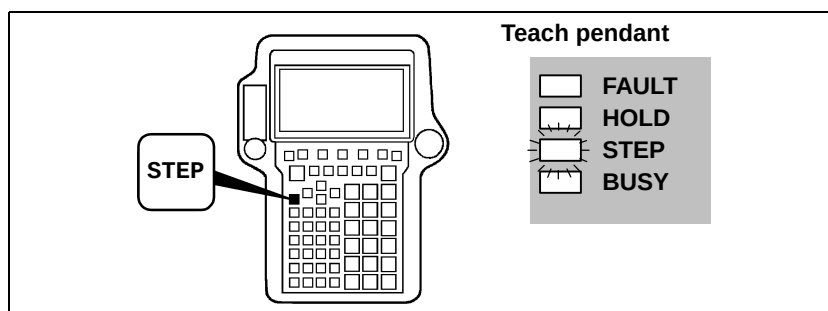


Fig. 6.3.2 (a) STEP Key

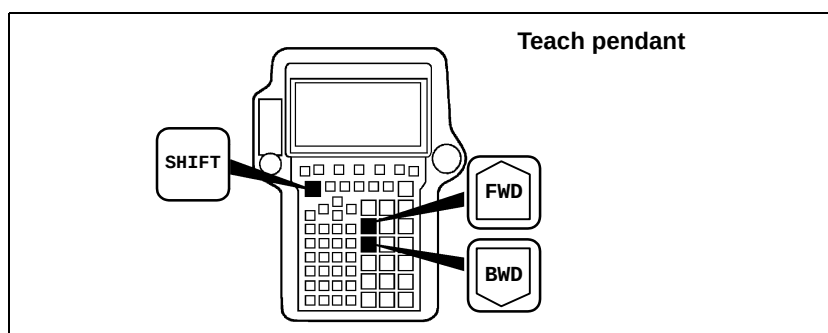


Fig. 6.3.2 (b) Starting Step Operation

Step operation can be performed in two ways: Forward execution and backward execution.

Forward execution

In forward execution, the program is executed in normal order. To perform forward execution of the program, press and hold down the SHIFT key, then press and release the FWD key on the teach pendant.



When a program is started, the program is executed for one line pointed to by the cursor, then the program is halted.

When a motion instruction is executed, the cursor is held at the executed line. When a logic instruction is executed, the cursor is moved to the next line.

Each time forward execution of the program is started, the next line of the program is executed.

When executing the circular motion instruction in step mode, the robot pauses near the through position on an arc. Moreover, if the robot is paused just before the through position, the robot does not stop at the through position after resuming a program.

Backward execution

In backward execution, the program is executed in reverse order. To perform backward execution of the program, press and hold down the SHIFT key, then press and release the BWD key on the teach pendant.



- During backward execution, only the motion instructions can be executed. However, a skip instruction, TIME BEFORE instruction, TIME AFTER instruction, soft float instruction, and other optional move instructions are ignored while the program is executed. After one line of the program is executed, the cursor is moved to the previous line.
- The instruction before the line where the following program instructions is taught can not be executed in backward execution. When you execute these instructions in backward execution, the cursor moves to the line following the line that contains taught these instructions:
 - Halt instruction (PAUSE)
 - Abort instruction (ABORT)
 - Program end instruction (END)
 - Jump instruction (JMP LBL[])
 - User alarm instruction (UALM[])
 - Execution instruction (RUN)
- The following program instructions cannot be executed:
 - Incremental instruction (INC)
- A blank line does not affect the execution of the program (Both Forward and Backward execution)

When the terminated program is restarted, the motion instruction in the line pointed to by the cursor is executed, then the program is halted.

Each time backward execution of the program is started, the program is executed using the motion format and feedrate specified in the current line, and the position data and positioning path of the motion instruction in the previous line.

- When the motion instruction in the current line specifies a circular motion, the robot moves to the target point along the path which passes through the passing point (Start point of an arc motion in normal program execution) specified in the current line.
- When the motion instruction in the previous line specifies a circular motion, the robot moves to the destination position specified in the previous line using the motion format and feedrate specified in the current line.

To disable backward execution of the program while the program is being executed, insert the halt instruction (PAUSE) into the desired location. After the halt instruction is executed, the cursor returns to the position it was at before the program was executed.

When the halt instruction is specified in the line before the line at the cursor is pointed at, backward execution of the program is disabled. To restart backward execution of the program, move the cursor to the line before the line that contains the halt instruction (two lines before the line at which the cursor is pointed).

Inter-program reverse program execution

With the inter-program reverse operation function, control can be returned from a subprogram to the main program that called the subprogram by performing reverse operation (SHIFT + BWD).

NOTE

- 1 Even if a subprogram exists during reverse operation of a main program, the subprogram cannot be called.
- 2 When program termination occurs within a subprogram, control cannot be returned to the main program.

When reverse execution is performed from a subprogram to the main program, the cursor stops at the line of the instruction that calls the subprogram taught in the main program.

Sample program

Example: When reverse operation is performed starting from the fourth line of a subprogram

```

Main_Prg
1:
2:R[1]=R[1]+1
3:J P[1] 100% FINE
4:IF R[1]=100, JMP LBL[100]
5:CALL Sub_Prog
6:
.
.
[End]
Sub_Prog
1:DO[1]=ON
2:DO[2]=ON
3:L P[2] 1000mm/sec FINE
4:L P[3] 1000mm/sec FINE
[End]

```

- 1 Start reverse operation with the cursor positioned to the fourth line of the subprogram.
- 2 Reverse operation (SHIFT + BWD) from P[3] to P[2]. The cursor is positioned to the third line of the subprogram.
- 3 Reverse operation (SHIFT + BWD) to the fifth line of the main program (CALL SUBPROGRAM). The cursor is positioned to the fifth line of the main program.
- 4 Reverse operation (SHIFT + BWD) from P[2] to P[1]. The cursor moves from the fifth line to third line of the main program.

Program end in backward execution

If the system variable \$BWD_ABORT is set to TRUE, when the first line of the program is finished to be executed during the backward execution, this program ends.

Procedure 6-7 Step test

Condition

- The teach pendant must be enabled.
- The single-step mode must be set.
- The system must be in the operation enable state.
- No one must be in the operating area. No obstacle must be placed in the operating area.

Step

- 1 Press the SELECT key. The program list screen is displayed.
- 2 Select the program to be tested and press the ENTER key. The program edit screen is displayed.
- 3 Press the STEP key to select the step mode. The STEP LED lights. (Check that the STEP LED lights when the STEP key is pressed.)

- 4 Move the cursor to the program start line.
- 5 Press and hold down the deadman switch, then turn on the teach pendant enable switch.

**WARNING**

The execution of the program instructions starts in the next step. The execution causes the robot to make a motion, which may produce unpredictable results. The operator should check that no persons and no unnecessary equipment is in the work area and that each part of the protective fence is sound. Otherwise, injury or property damage would occur. If the program needs to be stopped before it terminates, the operator should release the SHIFT key or deadman switch or press the HOLD or emergency stop button.

- 6 Start the program.
 - To perform forward execution of the program, press and hold down the SHIFT key, then press and release the FWD key. Do not release the SHIFT key until execution of the program is completed.
 - To perform backward execution of the program, press and hold down the SHIFT key, then press and release the BWD key. Do not release the SHIFT key until execution of the program is completed.
- 7 After one line of the program is executed, the program is halted.
 - When a motion instruction is executed, the cursor stops at the executed line. The next time forward execution of the program is performed, the next line of the program is executed.
 - When a control instruction is executed, the cursor moves to the next line.
- 8 To release the step mode, press the STEP key.
- 9 Turn off the teach pendant enable switch, then release the deadman switch.

6.3.3 Continuous Test

To perform a continuous test is to execute the program in the normal order from the current program line to the end of the program (end-of-program symbol or the program end instruction). Backward execution of the program is disabled during a continuous test.

A continuous test can be started using the teach pendant or operator's panel.

To perform a continuous test using the teach pendant, press and hold the SHIFT key, then press and release the FWD key. The program is then executed from the current line.

To start continuous test operation (cycle operation) from the operator's panel/box, momentarily press the start button on the operator's panel. Program execution then starts from the current line.

NOTE

The continuous text execution can be executed in the forward direction only.

Procedure 6-8 Continuous test (using the teach pendant)

Condition

- The teach pendant must be enabled.
- The continuous mode must be set. (The STEP lamp must be off.)
- The system must be in the operation enable state.
- No one must be in the operating area. No obstacle must be placed in the operating area.

Step

- 1 Press the SELECT key. The program list screen is displayed.
- 2 Select the program to be tested and press the ENTER key. The program edit screen is displayed.
- 3 Set the continuous mode. Check that the STEP LED is off. (If the STEP lamp is on, press the STEP key to turn it off.)

- 4 Move the cursor to the program start line.
- 5 Press and hold down the deadman switch, then turn on the teach pendant enable switch.

**WARNING**

The execution of the program instructions starts in the next step. The execution causes the robot to make a motion, which may produce unpredictable results. The operator should check that no persons and no unnecessary equipment is in the work area and that each part of the protective fence is sound. Otherwise, injury or property damage would occur. If the program needs to be stopped before it terminates, the operator should release the SHIFT key or deadman switch or press the HOLD or emergency stop button.

- 6 Press and hold down the SHIFT key, then press and release the FWD key. Hold down the SHIFT key until the execution of the program is completed. When the SHIFT key is released, the program is halted.
The program is executed to the end, then forcibly terminated. The cursor is returned to the first line of the program.

Procedure 6-9 Continuous test operation (started from the operator's panel)

Condition

- The operator's panel must be in the enabled state.
- Continuous operation mode must be set. (The step lamp must not be lit.)
- The system must be ready for operation.
- Nobody must be within the work area. There must be no obstacles.

Step

- 1 Press the select key. The program list screen is selected.
- 2 Select a program to be tested, and press the enter key. The program edit screen appears.
- 3 Set continuous operation mode. Check that the step lamp is not lit. (If the STEP lamp is on, press the STEP key to turn it off.)
- 4 Position the cursor to the first line.
- 5 Place the system in local mode. (For how to switch to local mode, see the description of Remote/Local setting in Section 3.15, "SYSTEM CONFIG MENU.")

**WARNING**

The execution of the program instructions starts in the next step. The execution causes the robot to make a motion, which may produce unpredictable results. The operator should check that no persons and no unnecessary equipment is in the work area and that each part of the protective fence is sound. Otherwise, injury or property damage would occur. If the program needs to be stopped before it terminates, the operator should release the SHIFT key or deadman switch or press the HOLD or emergency stop button.

- 6 Press the start button on the operator's panel/box. Program execution is performed up to the end of the program then terminated forcibly. The cursor returns to the first line of the program.

6.3.4 Program Look/Monitor

When the program is executed, the screen of the teach pendant becomes a monitor screen by which the execution of the program is displayed. In the monitor screen, the cursor moves to follow the line which is executed and you can not edit a program.

Program Monitor Screen

PROGRAM1	LINE 1	RUNNING
PROGRAM1		JOINT 30%
		1/10
1: J P[1]	100% FINE	
2: J P[2]	100% FINE	
3: J P[3]	100% FINE	
4: J P[4]	100% FINE	
5: J P[5]	100% FINE	
6: J P[6]	100% FINE	
LOOK		

Press F2,LOOK, then the program looking screen is displayed and the cursor of the program which is being executed stops(Program is kept to be executing). You can look at the desired part except the line which is executed with the arrow keys.

Program looking screen

PROGRAM1	LINE 8	RUNNING
PROGRAM1		JOINT 30%
		1/10
1: J P[1]	100% FINE	
2: J P[2]	100% FINE	
3: J P[3]	100% FINE	
4: J P[4]	100% FINE	
5: J P[5]	100% FINE	
6: J P[6]	100% FINE	
Under the LOOK mode		
MONITOR		

The message "Under the LOOK mode" is highlighted at the prompt line while looking at the program. To return to the monitor screen, press F2,MONITOR. When the monitor screen is displayed, the cursor specifies the line which is executed at that time.

If the execution of the program is paused or ended, the program edit screen is displayed in place of the program looking screen.

Program edit screen

PROGRAM1	LINE 6	ABORTED
PROGRAM1		JOINT 30%
		6/10
1: J P[1]	100% FINE	
2: J P[2]	100% FINE	
3: J P[3]	100% FINE	
4: J P[4]	100% FINE	
5: J P[5]	100% FINE	
6: J P[6]	100% FINE	
POINT		TOUCHUP>

6.4 MANUAL I/O CONTROL

Under manual I/O control, signals are transmitted between the robot and peripherals before the program is executed.

The manual I/O control refers to the following items:

- Forced output
- Simulated output and simulated input
- Wait release

6.4.1 Forced Output

Forced output is to manually turn digital output signals on or off. For the group output and the analog output, specify the value.

Procedure 6-10 Forced output

Condition

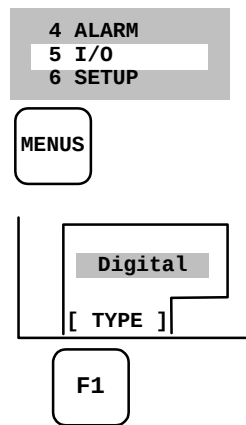
- Assignment of the signals to be output must be completed.

Step

- 1 Press the MENUS key to display the screen menu.
- 2 Select 5, I/O. The I/O screen is displayed.

Manual forced digital output

- 3 Press the F1, [TYPE] key to display the screen change menu.
- 4 Select Digital. The digital output screen or digital input screen is displayed.
If the input screen is displayed, press the F3, (IN/OUT) key to change the input screen to the output screen.

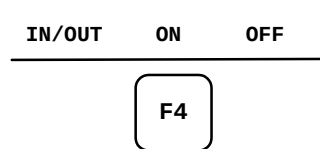


I/O Digital Out					JOINT 30%
	#	SIM	STATUS		1/512
DO[	1]	U	OFF	[	]
DO[	2]	U	OFF	[	]
DO[	3]	U	OFF	[	]
DO[	4]	U	OFF	[	]
DO[	5]	U	OFF	[	]
DO[	6]	U	OFF	[	]
DO[	7]	U	OFF	[	]
DO[	8]	U	OFF	[	]
DO[	9]	U	OFF	[	]
[TYPE]	CONFIG	IN/OUT	ON	OFF	

WARNING

Forced output activates connected equipment. Before executing the forced output, the operator should check which equipment is connected to the digital output and what operation the forced output would cause. Otherwise, injury or property damage could occur.

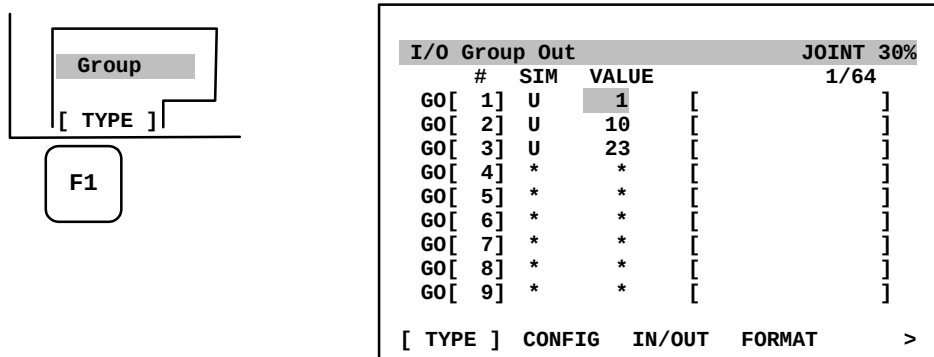
- 5 Move the cursor to the status field for the signal number to be changed, then press the F4 (ON) or F5 (OFF) key to change the signal output setting.



I/O Digital Out					JOINT 30%
	#	SIM	STATUS		1/512
DO[	1]	U	ON	[	]
DO[	2]	U	OFF	[	]
DO[	3]	U	OFF	[	]
DO[	4]	U	ON	[	]
DO[	5]	U	ON	[	]
DO[	6]	U	OFF	[	]
DO[	7]	U	OFF	[	]
DO[	8]	U	ON	[	]
DO[	9]	U	OFF	[	]
[TYPE]	CONFIG	IN/OUT	ON	OFF	

Manual forced group output

- 6 Press F1, [TYPE]. The screen change menu is displayed.
- 7 Select Group. The group output screen is displayed.



- 8 Move the cursor to the setting field of the signal number you want to change, enter the value. Pressing F4,FORMAT toggles between the decimal expression and the hexadecimal expression.

I/O Group Out				JOINT 30%	
#	SIM	VALUE		1/64	
GO[1]	U	1	[		
GO[2]	U	10	[		
GO[3]	U	23	[		
GO[4]	*	*	[		
GO[5]	*	*	[		
GO[6]	*	*	[		
GO[7]	*	*	[		
GO[8]	*	*	[		
GO[9]	*	*	[		

6.4.2 Simulated I/O

The Simulated I/O function changes the state of signals internally without making digital, analog, group or robot I/O communicate with peripherals. This function is used to execute the program or to test the I/O instruction when connection of I/O with peripherals is not completed. Simulated input/output can be used for digital, analog, group and robot I/O.

To enable simulated input/output, set the simulated flag, S.

Simulated output

The simulated output function internally changes the signal state using the I/O instruction of the program or manual output, but does not change the state of output to peripherals. This function holds the state of output to peripherals when the simulated flag is set.

When the simulated flag is reset, the output is restored to the original state.

Simulated input

The simulated input function internally changes the signal state with the I/O instruction of the program or manual input. The state of input from peripherals is ignored, and the signal state is not changed internally.

When the simulated flag is reset, the input enters the current state.

Refer to 6.3.1, "Specifying test execution" to specify whether I/O signal is disable in the test execution.

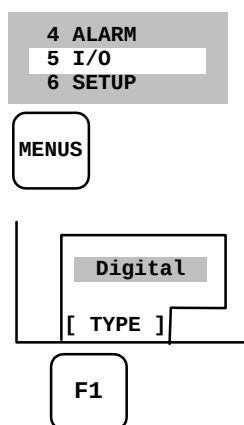
Procedure 6-11 Simulated input / output

Condition

- The input/output signal has been allocated.

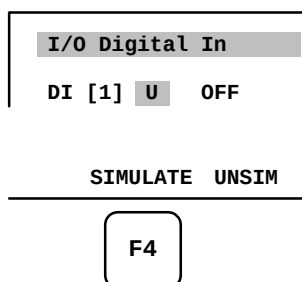
Step

- 1 Press the **MENUS** key. The screen menu is displayed.
- 2 Select **I/O**. The I/O screen is displayed.
- 3 Press **F1**, **[TYPE]**. The screen change menu is displayed.
- 4 Select **Digital**. Digital I/O screen is displayed.



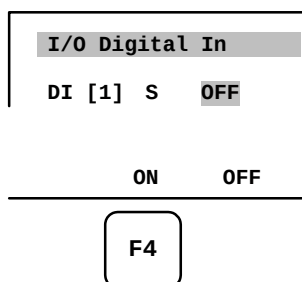
I/O Digital In				JOINT 30%
#	SIM	STATUS	1/512	
DI[1]	U	OFF	[digital signal 1]	
DI[2]	U	OFF	[digital signal 2]	
DI[3]	U	OFF	[digital signal 3]	
DI[4]	U	ON	[digital signal 4]	
DI[5]	U	ON	[digital signal 5]	
DI[6]	U	OFF	[digital signal 6]	
DI[7]	U	OFF	[digital signal 7]	
DI[8]	U	ON	[digital signal 8]	
DI[9]	U	ON	[digital signal 9]	
[TYPE]	CONFIG	IN/OUT	ON	OFF

- 5 Move the cursor to the **SIM** field for the signal number to be changed and press the **F4 (S)** or **F5 (U)** key to change the simulated setting.



I/O Digital In				JOINT 30%
DI[1]	S	OFF	[digital signal 1]	
[TYPE] CONFIG IN/OUT SIMULATE UNSIM				

- 6 Move the cursor to the **STATUS** field for the number of the signal to be simulated input/output and press the **F4 (ON)** or **F5 (OFF)** to change the simulated input/output setting.



I/O Digital In				JOINT 30%
DI[1]	S	ON	[digital signal 1]	
[TYPE]	CONFIG	IN/OUT	ON	OFF

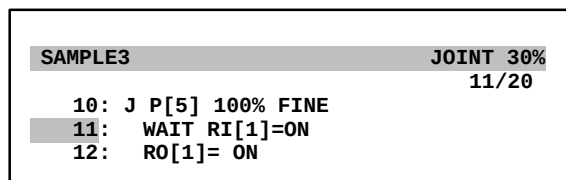
6.4.3 Wait Release

When a wait instruction in a program waits until the I/O conditions are satisfied, the wait release function skips this instruction, and halts program execution at the next line. Wait release is enabled only when a program is being executed.

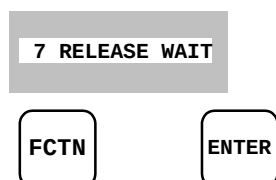
Wait release is performed by choosing from the function menu.

Procedure 6-12 Wait release**Condition**

- Program execution is currently in the I/O wait state.

**Step**

- 1 Press the function key to display the function menu.
- 2 Select 7 RELEASE WAIT.
The I/O wait is skipped, and the cursor moves to the next line. The program is then halted.
When program execution is restarted, the next instruction is executed.



6.5 OPERATING THE HAND MANUALLY

To operate the hand manually using the teach pendant, hand instruction must be assigned to the manual operation screen when macro instructions are set.

When teaching the operation of the hand, check whether the hand can actually hold a workpiece at the target position by operating the hand.

The macro instructions are optional functions.

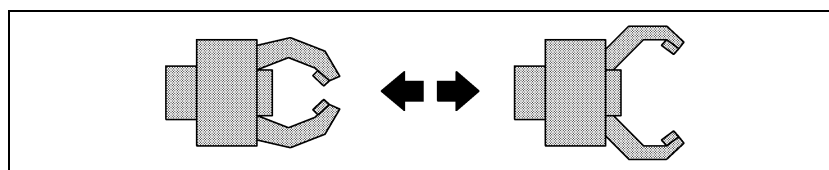


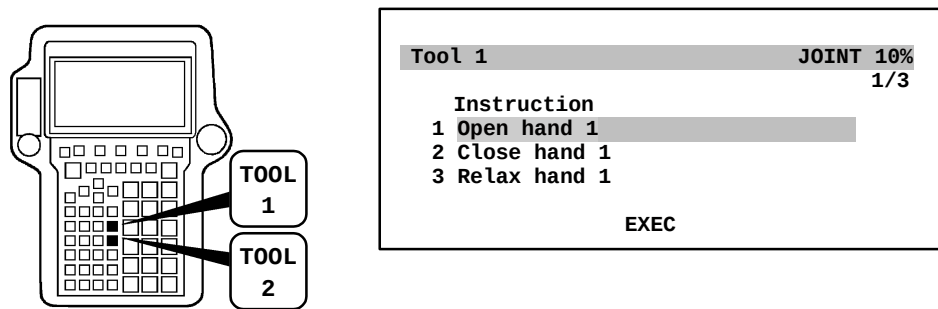
Fig. 6.5 Operating the Hand Manually

Procedure 6-13 Operating the hand manually**Condition**

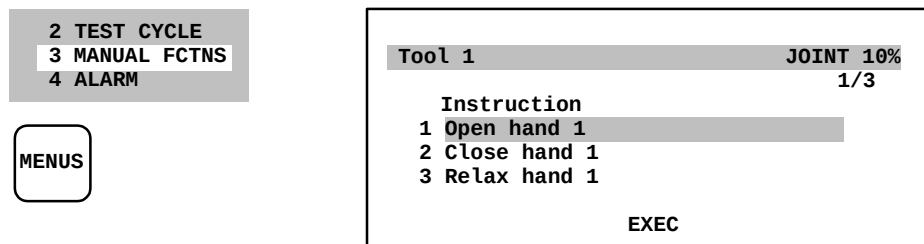
- The teach pendant must be enabled.
- No one must be in the operating area. All obstacles must be removed from the operating area.
- The hand instruction must be defined as a macro for manual operation (MF).

Step

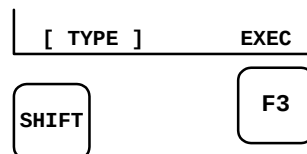
- 1 Press and hold down the deadman switch, then turn on the teach pendant enable switch.
 - Press the TOOL1 (or TOOL2) key. The Tool 1 (or Tool 2) screen is displayed.



- Otherwise, select MANUAL FCTNS from the screen menu to display the manual operation screen.



- 2 Move the cursor to the desired macro instruction. Press and hold down the SHIFT key, then press the F3 (EXEC) key.



6.6 AUTOMATIC OPERATION

Peripheral I/O can be used to automatically start a program and operate a production line. (See Section 3.8, "SETTING AUTOMATIC OPERATION".)

- The robot start request signals (RSR1 to RSR8 inputs) select and start a program. When a program is being executed or halted, the selected program is placed in the wait state. It is started once the currently executed program terminates.
- The program number selection signals (PNS1 to PNS8 inputs and PNSTROBE input) select a program. When a program is being halted or executed, these signals are ignored.
- The automatic operation start signal (PROD_START input) starts execution of the currently selected program from the first line. When a program is being halted or executed, this signal is ignored.
- Cycle stop signal (CSTOPI input) forcibly stops the currently executed program. Any programs enqueued by RSR are canceled.
 - If CSTOPI for ABORT on the system configuration menu is set to FALSE, the program currently being executed is executed up to the end of the program, then is terminated forcibly. Programs placed in the wait state by RSR are cleared. (Standard setting)
 - If CSTOPI for ABORT on the system configuration menu is set to TRUE, the program currently being executed is immediately terminated forcibly. Programs placed in the wait state by RSR are cleared.
- The external start signal (START input) starts a currently halted program.
 - If START for CONTINUE only on the system configuration menu is set to FALSE, the currently selected program is started from the current line. A temporarily stopped program is also started. (Standard setting)

- If START for CONTINUE only on the system configuration menu is set to TRUE, any temporarily stopped program is started. This signal is ignored when there is no temporarily stopped program.

To start a program by peripheral I/O input, the robot must be in the remote mode. The remote mode is set when the following remote conditions are satisfied:

- The teach pendant enable switch is turned off.
- Place the system in remote mode. (For how to switch to remote mode, see the description of Remote/Local setup in Section 3.15, "SYSTEM CONFIG MENU."
- Peripheral device I/O *SFSPD input is on.
- ENBL input of peripheral I/O is on.
- System variable \$RMT_MASTER is 0 (peripherals).

NOTE

The value of \$RMT_MASTER may be 0 (peripheral device), 1 (CRT/keyboard), 2 (host computer), or 3 (no remote device).

To start a program containing motion instructions, the following ready conditions must be satisfied:

- ENBL input of peripheral I/O must be on.
- The servo power is turned on. (No alarm is being issued.)

It is convenient to monitor the input acceptable signal (CMDENBL output) for starting a program using the peripheral I/O. The CMDENBL signal is output when the following conditions are satisfied:

- Remote condition
- Operation enable condition
- Continuous mode (step mode is disabled)

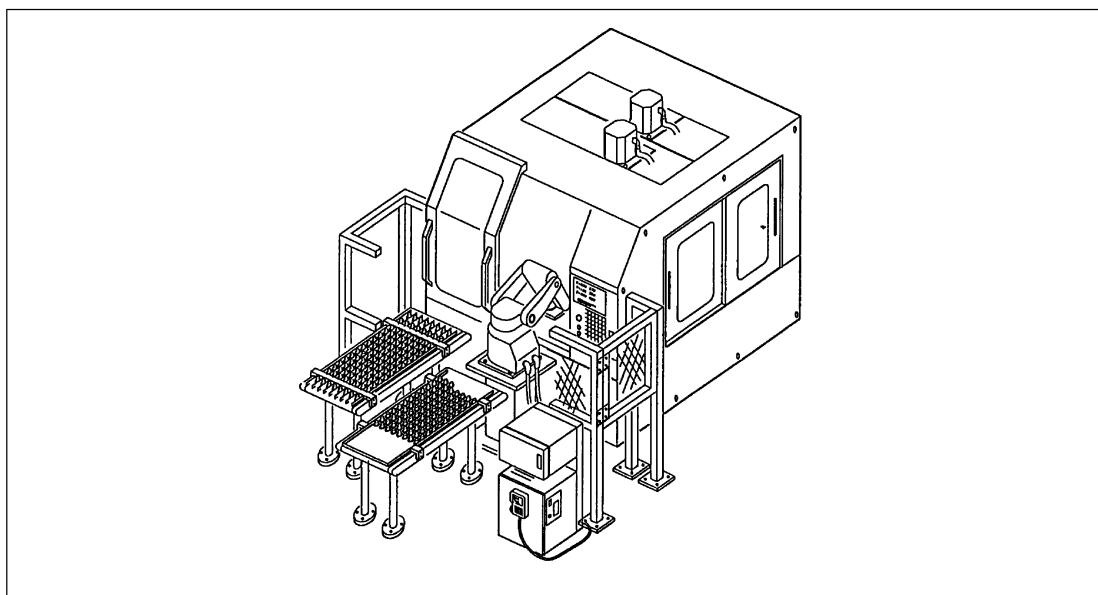


Fig. 6.6 Automatic Operation of Robot System

6.6.1 Automatic Operation by Robot Start Request (RSR)

The robot start request (RSR) function allows a remote device to select and start a program through the peripheral device I/O. This function uses eight robot start request signals (RSR1 to RSR8).

- 1 When a signal from RSR1 to RSR8 is input, the controller determines whether the input RSR signal is valid. If the signal is invalid, it is ignored. When a program started by a non-RSR signal, such as a start signal from the teach pendant or the operator's panel, or a dedicated signal START is being executed or halted, an RSR signal input is ignored.

Whether RSR is valid or invalid is set in system variables \$RSR1 to \$RSR8. These values can be changed on the RSR setting screen or by using a programmed RSR instruction.

- 2 Eight RSR registration numbers can be assigned to RSR. A base number is added to each RSR registration number to indicate an RSR program number (four-digit integer).

For example, when the RSR2 signal is input, a program having the following name is selected:

RSR + (RSR2 registration number + base number) (four digits)

NOTE

The name of a program to be started must be of "RSR + RSR program number" format. (Example: RSR0121)

The base number is set in \$SHELL_CFG.\$JOB_BASE. It can be changed by using Base number on the RSR setting screen or by using a programmed parameter instruction.

- 3 The RSR acknowledge output signal (ACK1 to ACK8) corresponding to one of the RSR1 to RSR8 input signals is output as a pulse signal. Even when one of the ACK1 to ACK8 signals is being output, RSR input is accepted.
- 4 When programs are in the terminated state, a selected program is started. When another program is being executed or halted, the request (job) is placed in a queue. It is started when the program currently being executed terminates.
Jobs (RSR programs) are executed in the order in which the programs were enqueued.
- 5 Programs in the queue are canceled (cleared) by the cycle stop signal (CSTOPI input) or forced program termination.

The start of a program by RSR is enabled in the remote mode. (Normally, in remote mode, the CMDENBL input is on.)

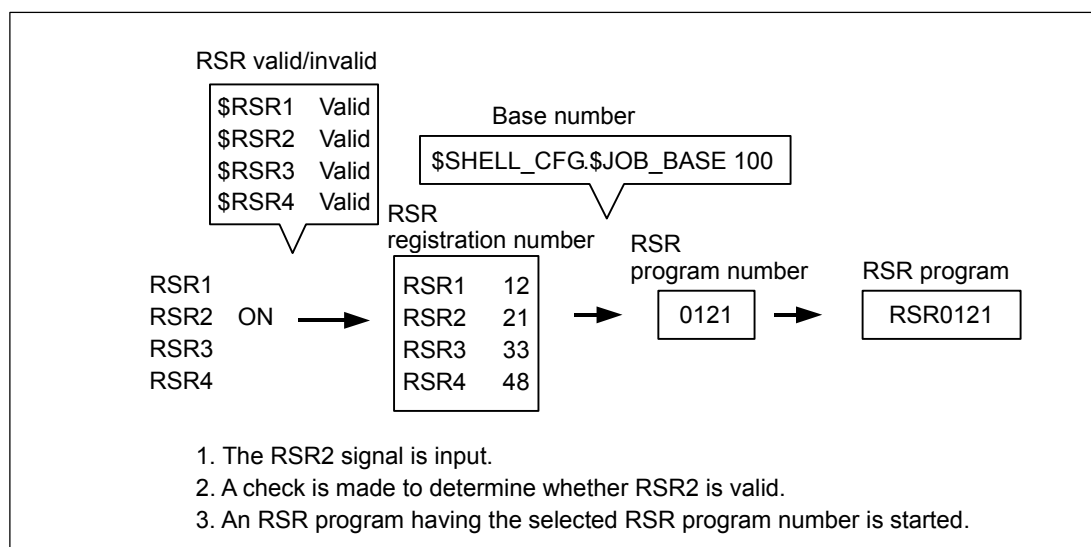


Fig. 6.6.1 Robot Start Request

Procedure 6-14 Automatic operation by robot start request (RSR)

Condition

- RSR settings are completed. (See Subsection 3.8.1.)
- Remote mode is set.
- The system is ready for operation.
- Nobody must be within the work area. There must be no obstacles.

**WARNING**

Applying this procedure starts automatic operation which causes the robot to move. An unpredictable operation could occur. Check to ensure that nobody is in the work area, that there are no unnecessary objects in the work space, and that the safety fence is normal. Also, check that all the automatic operation conditions are set correctly. Otherwise, personal injury or damage to the facilities could occur.

Step

- 1 Set the enable switch on the teach pendant to OFF.
- 2 Place the system in remote mode. (For how to switch to remote mode, see the description of Remote/Local setup in Section 3.15, "SYSTEM CONFIG MENU.")
- 3 Send the robot start signal (RSR1 to RSR8 input) of a target RSR number to the controller. The RSR program is placed in a queue.
- 4 To stop the program currently being executed, use the emergency stop button or hold button, or the immediate stop (*IMSTP input), hold (*HOLD input), or cycle stop (CSTOPI input) signal.
- 5 To cancel a job in the queue, use the cycle stop signal (CSTOPI input).
- 6 To restart a halted program, use the external start signal (START input).

6.6.2 Automatic Operation with Program Number Selection (PNS)

The program number selection (PNS) function enables selection or checking of a program, using the peripheral I/O, from the remote controller. Eight input signals, PNS1 to PNS8, specify a PNS program number.

- 1 When the PNSTROBE pulse signal is input, the controller reads the PNS1 to PNS8 input signals. When a program is being executed or halted, the signals are ignored. While the PNSTROBE pulse input signal is on, no program can be selected from the teach pendant.
- 2 The received PNS1 to PNS8 inputs are converted into a decimal number to obtain a PNS number. A program number (four digits) can be obtained by adding a base number to the PNS number, as shown below:

$$(\text{Program number}) = (\text{PNS number}) + (\text{base number})$$

The selected program has the following name:

PNS + (program number)

When zero is input through the PNS1 to PNS8 input signals, no program is selected on the teach pendant.

NOTE

The name of a started program must be of (PNS + PNS program number) format. (Example: PNS0138)

The base number is set in \$SHELL_CFG.\$JOB_BASE. It can be changed by using Base number on the PNS setting screen or a programmed parameter instruction.

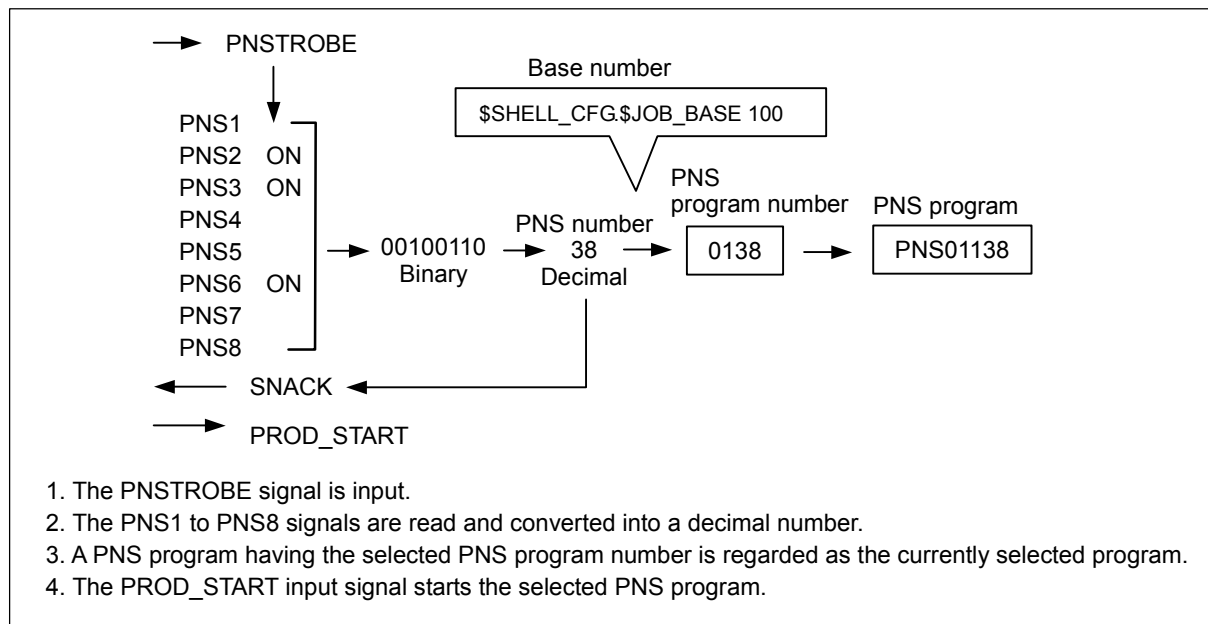


Fig. 6.6.2 Program Number Selection

- 3 The selected program number output signals (SNO1 to SNO8) are output for PNS confirmation. The PNS acknowledge output (SNACK) signal is output as a pulse signal. This signal causes the external device to read SNO1 to SNO8 output signals. Even while the SNACK signal is being output, PNS and PROD_START input signals are accepted.
 - 4 When confirming that the output values of SNO1 to SNO8 match the input values of PNS1 to PNS8, the remote controller sends the automatic operation start input (PROD_START) signal.
 - 5 The controller receives the PROD_START input signal, then starts the program.
- Program start by PNS is enabled in remote mode. (Normally, in remote mode, the CMNDENBL input signal is on.)

Procedure 6-15 Automatic operation by program number selection

Condition

- PNS setting must be completed (See Subsection 3.8.2).
- The remote condition must be satisfied.
- The operation enable condition must be satisfied.
- No one must be within the work area. There must be no obstacles.



WARNING

Start automatic operation as follows: When the robot starts operation, an unexpected situation may occur. To prevent any problem from occurring, be sure to check that no one is in the work area, that the work area is free from unnecessary equipment, that the safety barrier is in place, and that all the automatic operation conditions are correctly specified. Otherwise, the robot may injure a person or damage the equipment in the work area.

Step

- 1 Turn off the teach pendant enable switch.
- 2 Place the system in remote mode. (For how to switch to remote mode, see the description of Remote/Local setup in Section 3.15, "SYSTEM CONFIG MENU.")
- 3 Send the program number selection signals (PNS1 to PNS8 inputs) indicating a target PNS number and the PNS strobe signal (PNSTROBE input) to the controller. A PNS program is then selected.

The controller outputs the selected program number signals (SNO1 to SNO8 inputs) and PNS acknowledge signal (SNACK output) for confirmation.

- 4 Send an external start signal (PROD_START input). The selected program is then started.
- 5 To stop the program currently being executed, use the emergency stop button or hold button, or the immediate stop (*IMSTP input), hold (*HOLD input), or cycle stop (CSTOPI input) signal.
- 6 To restart a halted program, use the external start signal (START input).

6.6.3 External Override Selection Function

The external override selection function changes feedrate override by turning on or off digital input (DI) signals.

Two DI signals are defined. These two signals can be combined in four different ways. So four types of feedrate override can be selected.

OVERRIDE SELECT		JOINT 30%
		1/7
1	Function Enable :ENABLE	
2	Signal 1 : DI[1][ON]	
3	Signal 2 : DI[32][OFF]	
	Signal 1	Signal 2
4	OFF	OFF
5	OFF	ON
6	ON	OFF
7	ON	ON
		Override
		15 %
		30 %
		65 %
		100 %
[TYPE]		ENABLE DISABLE

When the function changes the feedrate override, the feedrate override is not displayed, namely, the popup menu is not displayed at the upper right corner of the screen.

To enable the external override selection function, the following requirements must be satisfied:

- The external override selection function must be enabled. (OVERRIDE SELECT on the setting screen)
- The remote mode must be set.

When the external override selection function is enabled, the following occurs:

- The override key of the teach pendant is practically disabled.(The changed value is quickly returned to the setting value by the external override selection.)
- The override instruction has no effect to the override value.
- You can not change the settings of DI signal number and Override. Before these settings can be modified, Function Enable : DISABLE must be set.
- When this function is effective at turning off the power of the controller, the override will get the value which had been set by this function when turning on it again.
- It is possible to specify the same number as two DI signal numbers. In this case, only the combination of ON-ON or OFF-OFF has the meaning.

Moreover, note the following:

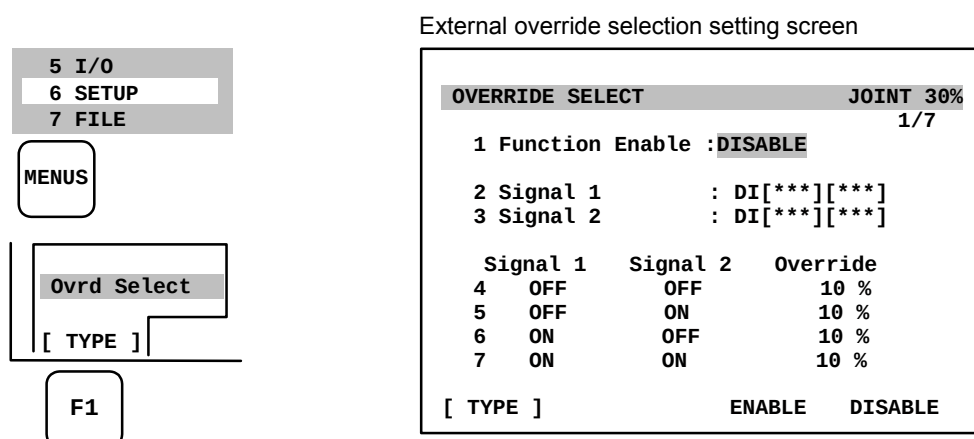
- After this function is disabled because the remote condition is not satisfied, the override keeps to remain the value specified by this function in effective until the value is changed by the teach pendant or override instruction.

Set this function on the external override selection setting screen (6 OVERRIDE SELECT).

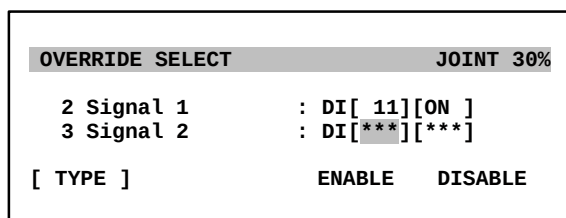
Procedure 6-16 Selecting an external override

Step

- 1 Press the MENUS key to display the screen menu, then select 6 SETUP.
- 2 Select Ovrd Select from the screen change menu.

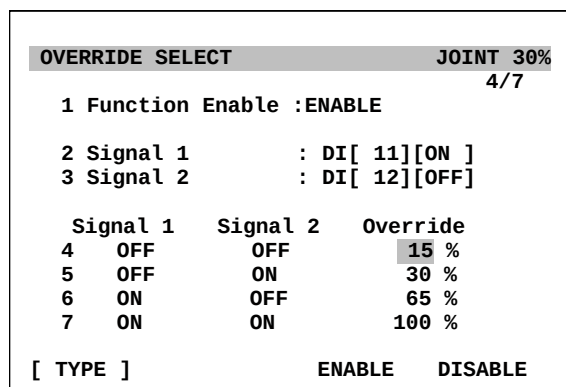


- 3 Set items.
 - a Enable or disable the function.
 - b Assign DI signals.



The states of DI signals are indicated. When *** is displayed, the setting of the function cannot be changed.

- c Feedrate override to be changed by turning on or off the signals



6.7 ONLINE POSITION MODIFICATION

Online position modification replaces all the position data and move speeds in the move instructions within a certain range in a program at one time, according to the position modification condition, during program execution.

The following program information can be modified:

- Position data (position compensation)
- Move speed

Position data is modified by adding a position compensation value. A movement speed is modified by rewriting it. Up to ten positions modification conditions can be defined.

Position compensation value

A position compensation value is the difference between the current position and the correct position. The position data coded in the move instructions within a specified range of a program is rewritten by adding a position compensation value to the data.

If the position data resulting from modification falls outside the allowable axial movement range, an alarm is generated when the program is executed.

**CAUTION**

If position compensation is performed during execution, it may take a while for the compensation to be reflected in actual operation.

The specifiable ranges (+/-) for the position compensation values are set in system variables \$PRGADJ.\$X_LIMIT to \$R_LIMIT. The standard value is +/-26 mm for (X, Y, Z) and +/-0.5 degrees for (W, P, R). Any position compensation value falling outside these ranges cannot be set.

Move speed

Move speeds in the move instructions within a specified range of a program are replaced with specified speeds. The move speed for axial movement is replaced by the value specified in Joint speed, while the move speed for linear and circular movement is replaced by the value specified in Linear/Circular speed.

**CAUTION**

Once a speed has been rewritten, the original speed cannot be restored.

Position modification status

The position modification statuses are classified into the following three types:

- EDIT indicates that the current position modification condition is being edited. It is not reflected in the program. This state is indicated when no position modification condition is set or when a valid position modification condition is edited.
- ENABLED indicates that the current position modification condition is reflected in the program.
- DISABLED indicates that the position modification condition reflected in the program has been canceled.

The result of ENABLED is reflected immediately if the program is being executed.

When the position modification condition is modified after ENABLED, changes made to the program are determined, and state EDIT is indicated.

Online position modification is set by using 1 UTILITIES Prog Adjust on the utility screen. Online position modification conditions include the following information:

Position modification condition list screen

UTILITIES Prog Adj			JOINT 10%
Program Adjust Schedule			5/10
#	Program	Lines	Status
1	SAMPLE1	22-29	ENABLED
2	SAMPLE1	39-49	ENABLED
3	SAMPLE3	10-14	DISABLED
4	SAMPLE4	123-456	EDIT
5	*****	0-0	*****
6	*****	0-0	*****
7	*****	0-0	*****
8	*****	0-0	*****
9	*****	0-0	*****
10	*****	0-0	*****
[TYPE] DETAIL			>
COPY CLR_ADJ CLR_ALL			>

Position modification condition detail screen

SAMPLE		LINE 0	
UTILITIES Prog Adj		JOINT 10%	
		1/14	
Current schedule: 5 Status:EDIT			
Program Name:			
1	SAMPLE2		
2	Starting line number:	1	
3	Ending line number:	30	
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
11	Linear/Circular speed:	0	mm/s
12	Joint speed:	0	%
13	Motion group:	ALL	
14	Adjust Y for:	ROBOT	
[TYPE] UNITS SCHED		[CHOICE]	>
COPY CLR_ADJ CLR_ALL		[CHOICE]	>

Table 6.7(a) Online Position Modification Settings

Item	Description
Program	Specifies the name of the target program for position modification.
Range	Specifies the range (the start and end lines) of the program lines to which position modifications are to be applied. NOTE The end line number must be greater than or equal to the start line number specified in item 2. When only one line is to be modified, the end line number must equal the start line number.
Offset relative to	User Modification is performed in reference to the user coordinate system. Tool Modification is performed in reference to the tool coordinate system.
Status	The position modification status indicates whether a specified position modification condition is reflected in the program. • EDIT : The position modification condition is being edited. • ENABLED : The position modification condition is reflected in the program. • DISABLED : The position modification condition is not reflected in the program.
X to R adjustment	Compensation values X to R indicate the position compensation amounts. Values (X, Y, Z) are in mm or inches, while values (W, P, R) are in degrees. The values specified here are included in the position data.
Linear/Circular speed Joint speed	These speed items replace the move speeds. Linear/Circular speed replaces the linear and circular movement speed with a specified speed. Joint speed replaces the axial movement speed with a specified speed.  CAUTION Once the move speed is rewritten, the original speed cannot be restored.
Motion group	Select an operation group to be subjected to modification.

Item	Description
Adjust Y for	This item is displayed only when an additional built-in traveling axis is set up as the seventh axis in group 1. The direction of the additional built-in axis is indicated in motion group. Specify the compensation target for the indicated direction. Robot: Modify only the position of the robot. Additional axis: Modify the position of the additional axis. All: Modify both the positions of the robot and the additional axis. If offset relative to is set to "Tool," only the robot can be selected.

[TYPE]	UNIT	SCHED	ENABLE	DISABLE	>
COPY	CLR_ADJ	CLR_ALL			>

Table 6.7(b) Function key for online position modification setting

Function key label	Description
UNIT	The position modification unit function changes the units of the position modification values (mm or inches).
SCHED	The schedule function is used to input the number of the position modification condition to be edited next.
ENABLE	ENABLED reflects the current position modification condition in a target program. The position data and move speeds are rewritten according to the position modification condition. This function key can be specified only when EDIT or DISABLED is indicated.
DISABLE	DISABLED cancels the current position modification condition reflected in a target program. The position data used before modification is restored. This function key can be specified only when ENABLED is indicated. The original move speed cannot be restored.
COPY	The position modification condition copy function copies a selected position modification condition into another condition number. After copying, EDIT is indicated as the modification status.
CLR_ADJ	The position modification condition erase function erases all the position modification and speed values set in a selected position modification condition. The program name and range are not erased. When erase is performed, the modified program is not restored to its original state.
CLR_ALL	This function key erases a selected position modification condition entirely including the program name and range. When erase is performed, the modified program is not restored to its original state.

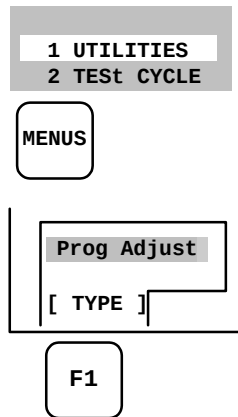
Procedure 6-17 Online position modification

Condition

- There is a program to be modified.

Step

- 1 Press the MENUS key to display the screen menu.
- 2 Select 1 UTILITIES.
- 3 Press F1, [TYPE] to display the screen selection menu.
- 4 Select Prog Adjust. then, the position modification condition list screen appears.



Position modification condition list screen

UTILITIES Prog Adj			JOINT 10%
Program Adjust Schedule			5/10
#	Program	Lines	Status
1	SAMPLE1	22-29	ENABLED
2	SAMPLE1	39-49	ENABLED
3	SAMPLE3	10-14	DISABLED
4	SAMPLE4	123-456	EDIT
5	*****	0-0	*****
6	*****	0-0	*****
7	*****	0-0	*****
8	*****	0-0	*****
9	*****	0-0	*****
10	*****	0-0	*****
[TYPE] DETAIL			>
COPY CLR_ADJ CLR_ALL			>

- 5 Position the cursor to the line number of a program to be modified. If the program to be modified is not indicated, select "*****".
- 6 Press F2, DETAIL. Then, the position modification condition detail screen appears. When "*****" is selected, EDIT is indicated as the status.

UTILITIES Prog Adj		
#	Program	Lines
1	SAMPLE1	22-29
2	SAMPLE1	39-49
3	SAMPLE3	10-14
4	Test-pro	123-456
5	*****	0-0
[TYPE] DETAIL		
F2		

Position modification condition detail screen

UTILITIES Prog Adj			JOINT 10%
			1/14
Current schedule: 5 Status:EDIT			
Program Name:			
1	SAMPLE2		
2	Starting line number:	0	
3	Ending line number:	0	
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/s
12	Joint speed:	10	%
13	Motion group:	All	
14	Adjust Y for:	ROBOT	
[TYPE] UNITS SCHED [CHOICE]			>

- 7 Set each items as desired.

NOTE

When only one program line is to be modified, enter the same value for both the start and end lines.

UTILITIES Prog Adj		JOINT 10%	
		1/14	
Current schedule: 5 Status:EDIT			
Program Name:			
1	SAMPLE2		
2	Starting line number:	1	
3	Ending line number:	30	
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
11	Linear/Circular speed:	0	mm/s
12	Joint speed:	10	%
13	Motion group:	All	
14	Adjust Y for:	ROBOT	
[TYPE]	UNITS	SCHED	[CHOICE] >
COPY	CLR_ADJ	CLR_ALL	[CHOICE] >

- 8 After completing the modification condition settings, press F4 ENABLE to reflect the position modifications in the target program. The result of ENABLE is reflected immediately if the program is being executed.

SCHED	ENABLE	>
F4		

NOTE

- 1 To modify a position modification condition after making it valid, cancel the condition once, then modify it.
- 2 When move instructions include a position register or incremental instruction, modifications are not reflected.

- 9 To cancel a set modification condition, press F5 DISABLE. When DISABLE is used, the current position modification condition must be valid.

SCHED	DISABLE	>
F5		

**CAUTION**

Once a move speed has been changed, the original speed cannot be restored even by pressing DISABLE.

- 10 To set the position modification condition of another condition number, press F3 SCHED.

SCHED	DISABLE	>
F3		

7 STATUS DISPLAY

The user can check various statuses of the robot with status display. Several types of screens are used for status display.

Contents of this chapter

- 7.1 LEDS ON THE TEACH PENDANT
- 7.2 USER SCREEN
- 7.3 REGISTERS
- 7.4 POSITION REGISTERS
- 7.5 PALLETIZING REGISTERS
- 7.6 STRING REGISTERS
- 7.7 CURRENT POSITION
- 7.8 SYSTEM VARIABLES
- 7.9 PROGRAM TIMER
- 7.10 SYSTEM TIMER
- 7.11 EXECUTION HISTORY
- 7.12 MEMORY USE STATUS DISPLAY
- 7.13 SAFETY SIGNALS
- 7.14 PROGRAM STATUS

7.1 LEDS ON THE TEACH PENDANT

The LEDs on the teach pendant indicate the following statuses:

Table 7.1 LEDs on the Teach Pendant

LED	Description
FAULT	This LED indicates that an alarm has been issued. When the alarm is released, this LED goes off.
HOLD	This LED goes on while the HOLD key on the teach pendant or operator's panel is pressed or while the peripheral I/O signal, *HOLD, is applied.
STEP	This LED goes on when the single step mode is set. This LED goes off when the continuous operation mode is set.
BUSY	This LED indicates that a program or other processing is being executed.
RUNNING	This LED indicates that a program is being executed.
JOINT	This LED goes on when the manual-feed coordinate system is a joint jog coordinate system.
XYZ	This LED goes on when the manual-feed coordinate system is a Cartesian jog coordinate system (jog coordinate system or cartesian coordinate system or user coordinate system).
TOOL	This LED goes on when the manual-feed coordinate system is a tool jog coordinate system.

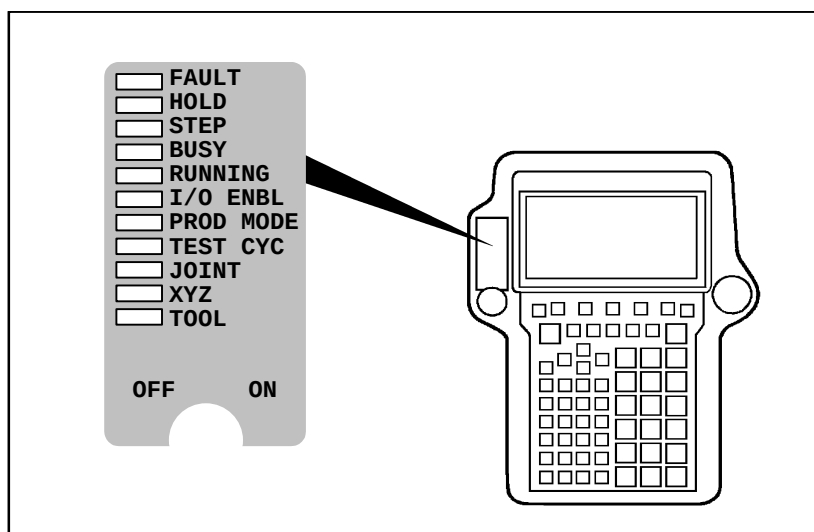


Fig. 7.1 LEDs on the Teach Pendant

7.2 USER SCREEN

A message instruction for the program being executed is displayed on this screen. (See Subsection 4.14.7.) When a message instruction is executed, the screen display automatically switches to the user screen.

Procedure 7-1 User screen display

Step

- 1 Press the MENUS key.
- 2 Select "9 USER."

NOTE

- 1 When a message instruction is not executed, nothing is displayed on this screen.
- 2 Even after the program is forcibly terminated, the message remains on the screen.

7.3 REGISTERS

A register is a variable for holding an integer or fraction. Two hundreds registers are provided. The register screen is used to display and set registers.

Procedure 7-2 Displaying register screen

Step

- 1 Press the MENUS key to display the screen menu.
- 2 Press "NEXT," then select "DATA."
Alternatively, instead of steps 1 and 2 above, the user can press the DATA key.
- 3 Press F1 "TYPE."
- 4 Select "Registers." The register screen appears.

DATA Registers		JOINT 30%
		1/200
R [1:	]	= 0
R [2:	]	= 0
R [3:	]	= 0
R [4:	]	= 0
R [5:	]	= 0
R [6:	]	= 0
Enter value		
[TYPE]		

**WARNING**

Registers are used in a program. Never change the value of a register before checking how the register is used in the system. Otherwise, the program can be adversely affected.

- 5 To enter a comment, use the following procedure:
 - a Move the cursor to a desired register number field, then press ENTER key.
 - b Select a comment input method.
 - c Press a desired function key, then enter a comment.
 - d Upon completion of input, press the ENTER key.
- 6 To change the value of a register, move the cursor to the register value field, then enter a desired value.

1	2	ENTER

DATA Registers		JOINT 30%
		1/200
R [1:	]	= 12
R [2:	]	= 0
R [3:	]	= 0
R [4:	]	= 0
R [5:	]	= 0
R [6:	]	= 0
Enter value		
[TYPE]		

Programming example

- 7 Registers are used in programs when the following are specified:
 - Register instruction (See Subsection 4.5.1)
 - Indirect specification of arguments (See Section 4.2)

SAMPLE4		JOINT 30%
		1/8
1: R[1]=0		
2: LBL[1]		
3: CALL PRG_A		
4: R[1]=R[1]+1		
5: IF R[1]<=10, JMP LBL[1]		
6: CALL PRG_B		
7: ABORT		
[End]		
[TYPE]		
		[EDCMD] >

Program A is repeated
11 times, program B is
executed, then program
execution terminates.

7.4 POSITION REGISTERS

A position register is a variable for holding position data. One hundreds position registers are provided. The position register screen is used to display and set registers.

Procedure 7-3 Position register setting

Step

- 1 Press the MENUS key to display the screen menu.
- 2 Press "0 NEXT," then select "3 DATA."
Alternatively, instead of steps 1 and 2 above, the user can press the DATA key.
- 3 Press F1, [TYPE] to display the screen change menu.
- 4 Select "Position Reg." The position register screen appears.

DATA Position Reg		JOINT	30%
			1/100
PR [1:		= *	
PR [2:		= *	
PR [3:		= *	
PR [4:		= *	
PR [5:		= *	
PR [6:		= *	
Press Enter			
[TYPE] MOVE_TO RECORD POSITION CLEAR			



WARNING

Position registers are used in a program. Never change the value of a position register before checking how the register is used in the system. Otherwise, the program can be adversely affected.

- 5 To enter a comment, use the following procedure:
 - a Move the cursor to a desired position register number field, then press ENTER key.
 - b Select a character input method.
 - c Press a desired function key, then enter a comment.
 - d Upon completion of input, press the ENTER key.
- 6 To change the value of a position register, move the cursor to the position register value field. Then, press F3 "RECORD" while holding down the SHIFT key.

DATA Position Reg		JOINT	30%
			1/100
PR [1:		= R	
PR [2:		= *	
PR [3:		= *	
PR [4:		= *	
PR [5:		= *	
PR [6:		= *	
Press Enter			
[TYPE] MOVE_TO RECORD POSITION CLEAR			

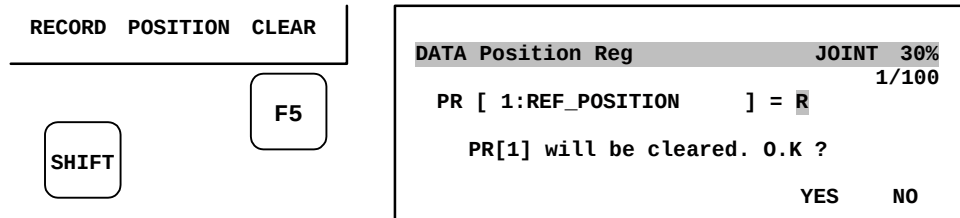
Diagram showing key presses: SHIFT, F3, and the menu bar [TYPE] MOVE_TO RECORD.

- "R" indicates that a position register already holds a taught value.
- An asterisk (*) indicates that it does not.

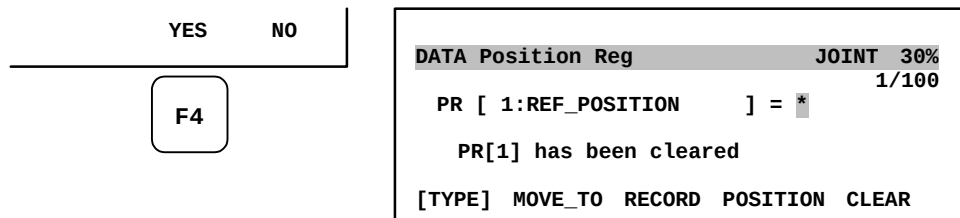
NOTE

In a multi-motion group system, teaching a position register records the position data for all axes regardless of the current motion group.

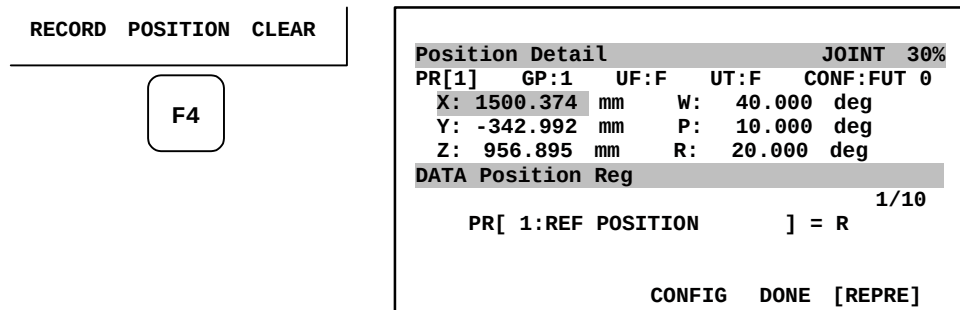
- 7 To delete position data loaded into a position register, press F5 “CLEAR” while holding down the SHIFT key.



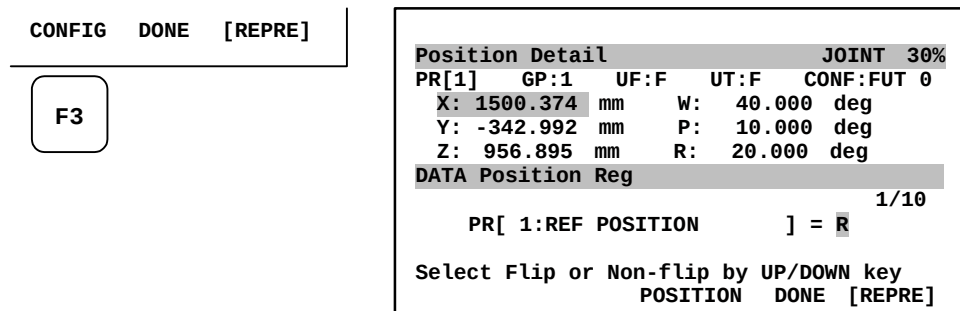
- 8 Select “YES.” The position data of the desired position register is cleared.



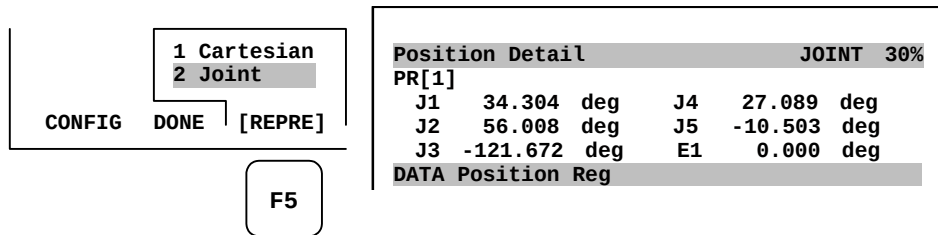
- 9 To find out the current values of position data, press F4 “POSITION.” The position detail data screen appears. To change a value, move the cursor to the desired field, then enter a new value.



- 10 To change the configuration, press F3 “CONFIG.” Move the cursor to a desired field, then change joint placement data using the ↓ and ↑ keys.



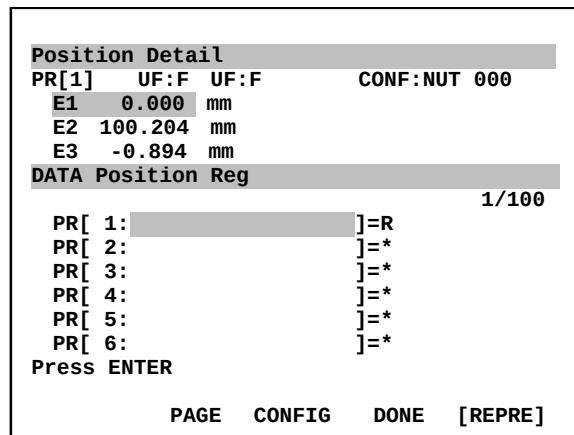
- 11 To change the storage form of the position data, press F5,[REPRE] and select the storage form.



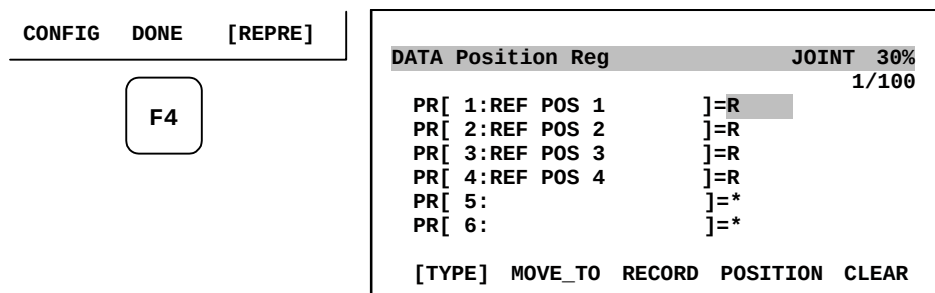
NOTE

JOINT display is valid when the robot is adjusted to the zero-degree position or when non-kinematic operation such as table operation control is executed.

- 12 To change the display to the additional axes (subgroup), press F2 PAGE.



- 13 Upon completion of setting, press F4 "DONE."



- 14 The position register can be used in the program as the following case:
- Position data of motion instruction(See Subsection 4.3.2)
 - Position register instruction and offset instruction, etc. (See Section 4.5 and Subsection 4.3.5)

Programming example

```

SAMPLE5                                JOINT 30%
                                      12/18
12: LBL[1]
13: OFFSET CONDITION PR[1]
14:L PR[2] 1000cm/min CNT100 Offset
15: PR[3,6]=R[10]
16:L PR[3] 1000mm/sec CNT100
17:L PR[4] 1000mm/sec CNT100 Offset
[End]
[TYPE]                                [EDCMD] >

```

7.5 PALLETIZING REGISTERS

The palletizing register screen displays the current values of the palletizing registers.

Procedure 7-4 Displaying a palletizing register screen

Step

- 1 Press the MENUS key to display the screen menu.
- 2 Press "0 NEXT," then select "3 DATA."
Alternatively, instead of steps 1 and 2 above, the user can press the DATA key.
- 3 Press F1 "TYPE".
- 4 Select "Pallet regis." The pallet register screen is displayed.

```

DATA: Pallet Regist                    JOINT 30%
                                      1/10
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]
Enter value
[TYPE]

```



WARNING

Palletizing registers are used in a program. Never change the value of a palletizing register before checking how the register is used in the system. Otherwise, the program can be adversely affected.

- 5 To enter a comment, use the following procedure:
 - a Place the cursor on the comment line, then press ENTER key.
 - b Select the way of naming the comment.
 - c Press a desired function key, then enter characters.
 - d Upon completion of input, press the ENTER key.
- 6 To change the value of a palletizing register, move the cursor to the palletizing register value field, then enter a new value.

JOINT 30% 1/10]=[1, 1, 1]		DATA: Pallet Regist		JOINT 30% 1/10	
2 ENTER		PL[1:PALLET	]=[2, 1, 1]		
		PL[2:	]=[1, 1, 1]		
		PL[3:	]=[1, 1, 1]		
		PL[4:	]=[1, 1, 1]		
		PL[5:	]=[1, 1, 1]		

7.6 STRING REGISTERS

NOTE

This item 7.6 is available only on 7DA5 series or later.

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.

Procedure 7-5

Step

- 1 Press the MENUS key to display the screen menu.
- 2 Press "0 NEXT," then select "3 DATA."
Alternatively, instead of steps 1 and 2 above, the user can press the DATA key.
- 3 Press F1, [TYPE] to display the screen change menu.
- 4 Select "String Reg." The string register screen appears.

DATA String Regist		JOINT 100%
SR[1:		=
SR[2:		=
SR[3:		=
SR[4:		=
SR[5:		=
[TYPE] DETAIL IMPORT		

- 5 To add a comment
 - a. Move the cursor to the string register number and press ENTER.
 - b. Select the way of naming the comment.
 - c. Press a desired function key, and then enter characters.
 - d. Upon completion of input, press the ENTER key.
- 6 To change the value
 - a. Move the cursor to the string register number on the right side of the = and press ENTER.
 - b. Select the way of naming the value.
 - c. Press a desired function key, and then enter characters.
 - d. Upon completion of input, press the ENTER key.
- 7 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.
- 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.

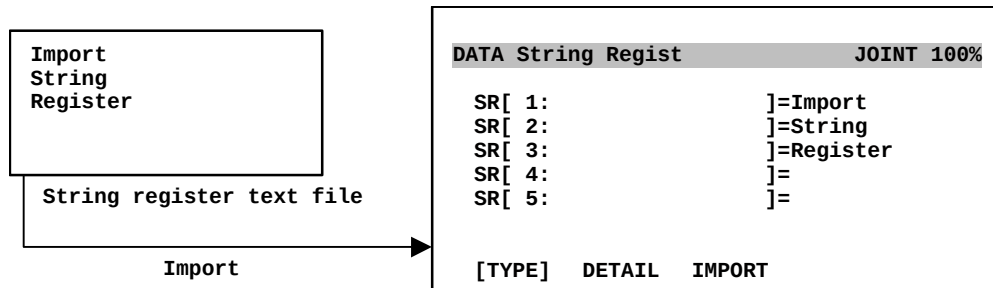
NOTE

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.

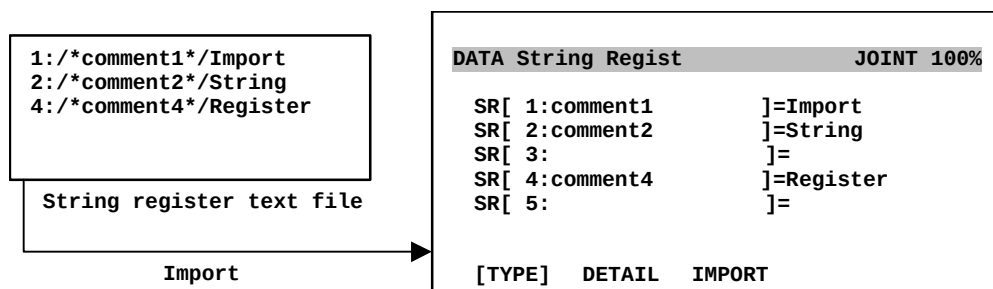
- The simple format containing only values for each string registers
The values will be imported starting with string register number 1 and increment to the next string register number on the end of the line.

Example:



- The full format containing string register number, comment, and values
A specific string register comment and value can be set by importing the string register text file written in the format such as “number: /*comment*/ value”.

Example:



7.7 CURRENT POSITION

The current position of the robot shows the location and the orientation of the robot in the work space. The current position can be represented in the cartesian frame and the joint frame.

Joint coordinates

Joint coordinates represent the current position by the angular displacement from the base side of each axis.

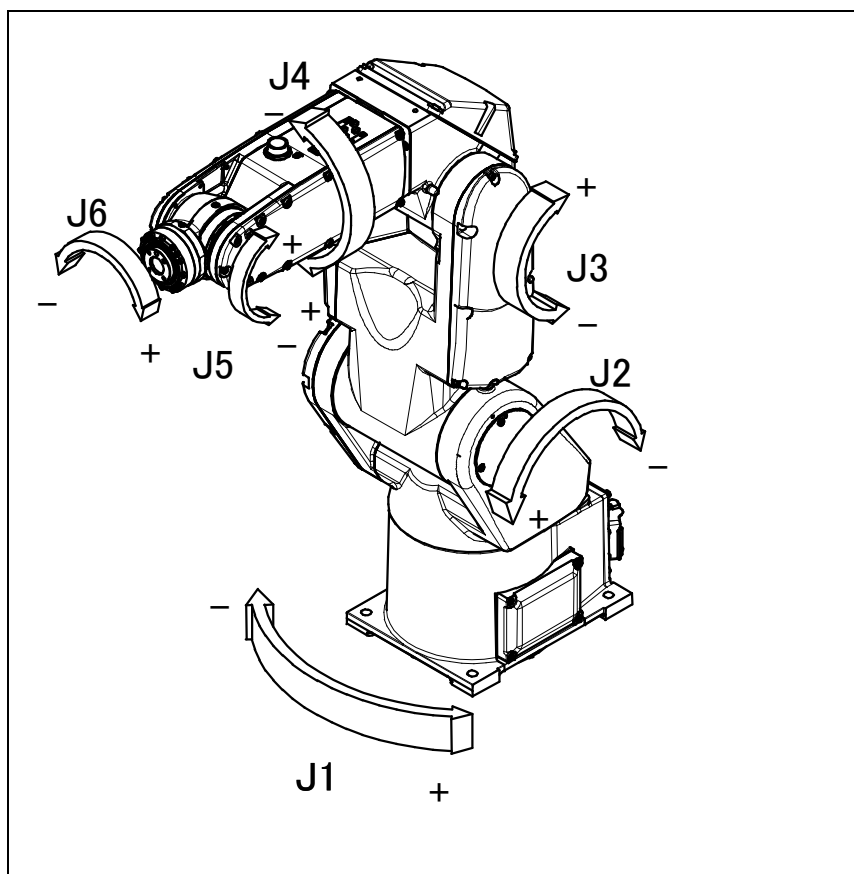


Fig. 7.7 (a) Joint Coordinate System

Displaying joint coordinates

POSITION			JOINT 30%		
Joint			Tool: 1		
J1:	0.000	J2:	0.000	J3:	0.000
J4:	0.000	J5:	0.000	J6:	0.000
E1:	*****	E2:	*****	E3:	*****
J2/J3 Interaction: 0.000					
[TYPE]	JNT	USER	WORLD		

NOTE

If the system has an additional axis, E1, E2 and E3 indicate the position data of the additional axis.

Displaying cartesian coordinates

The current position represented in cartesian coordinates is defined by the tool frame which is defined on the wrist to specify the location and orientation of the tool, and the cartesian frame which is fixed in the work space. Cartesian coordinates is represented by the world frame or the user frame.

4 STATUS
5 POSITION
6 SYSTEM

MENUS

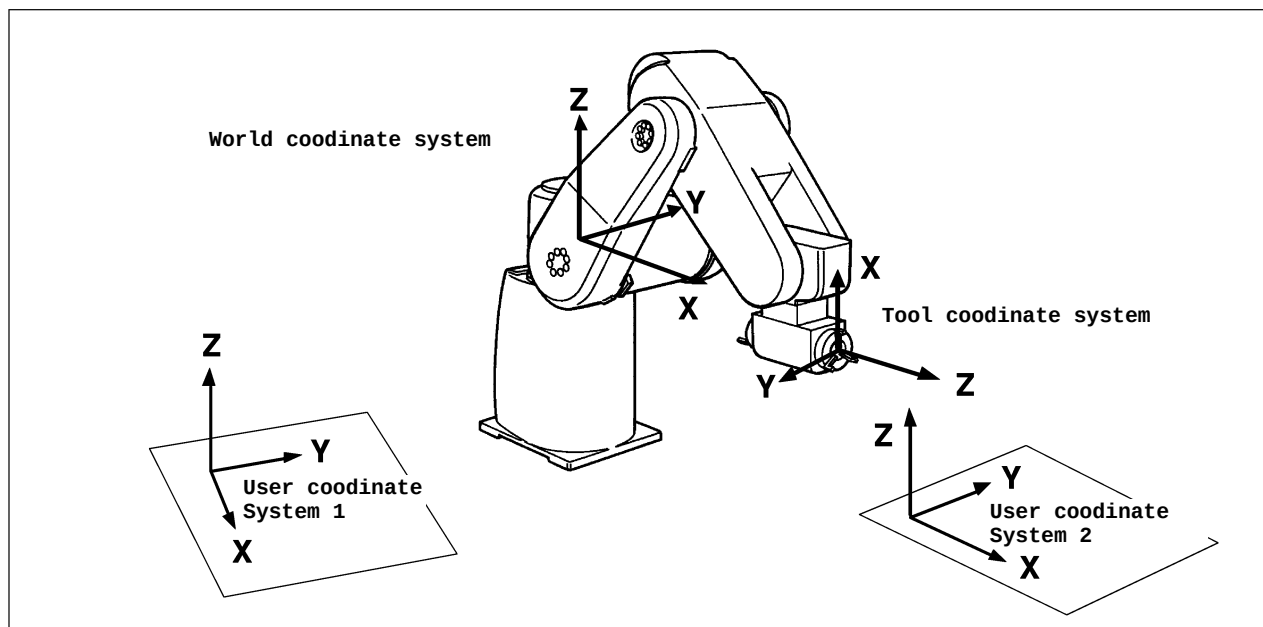


Fig. 7.7 (b) Cartesian coordinate system

Displaying world coordinate system

POSITIONJOINT 30%

WorldTool: 1

Configuration: FUT 0

x:1380.000y:-380.992z: 956.895

w: 40.000p: -12.676r: 20.000

E1: *****E2: *****E3: *****

[TYPE]JNTUSERWORLD

Display user coordinate system

POSITIONJOINT 30%

UserFrame: 0Tool: 1

Configuration: FUT 0

x:1500.374y:-342.992z: 956.895

w: 40.000p: 10.000r: 20.000

E1: *****E2: *****E3: *****

[TYPE]JNTUSERWORLD

Procedure 7-6 Displaying current position screen

Step

- 1 Press the MENUS key to display the screen menu.
- 2 Select NEXT, then select POSITION from the next menu.



- 3 The current position screen can be also displayed by pressing the POSN key.
 - To display joint coordinates, press F2 "JNT."
 - To display user coordinates, press F3 "USER."
 - To display world coordinates, press F4 "WORLD."

7.8 SYSTEM VARIABLES

All the system variables can be seen with the system variable screen. Settings of the system is stored in the system variables.

**WARNING**

The operation of the robot and controller is controlled with system variables. Only a person who knows details of the influence of changes in system variables should set system variables. If a person without detailed knowledge attempts to set the system variables, the robot and controller would malfunction.

Procedure 7-7 Displaying system variable screen

Step

- 1 Press the MENUS key. The screen select menu is displayed.
- 2 Select NEXT, then select SYSTEM.
- 3 Press F1,[TYPE].
- 4 Select Variables. The system variable screen is displayed.

SYSTEM Variables		JOINT 10%
		1/729
1	\$AP_MAXAX	536870912
2	\$AP_PLUGGED	4
3	\$AP_TOTALAX	16777216
4	\$AP_USENUM	[12] of Byte
5	\$AUTOINIT	2
6	\$BLT	19920216
7	\$CRT_DEFPROG	*uninit*
8	\$CSTOP	TRUE
9	\$DEFPULSE	4
10	\$DEVICE	'P3:'

[TYPE]

- 5 To change the settings of the system variables, move the cursor to the desired field and press the ENTER key after entering the value, or select the desired item from the function labels.
- 6 When one of the system variables has plural items which belong to this variable(hierarchical structure),move the cursor to the desired system variable and press the ENTER key. Then the list of items which belongs to this variable is displayed.

**WARNING**

Power should be turned on again to make a new setting valid. Otherwise, injury or property damage would occur.

SYSTEM Variables		JOINT 10%
		49/729
47	\$ORIENTTOL	10.000
48	\$OVRDSLCT	OVRDSLCT_T
49	\$PARAM_GROUP	MRR_GRP_T
50	\$PASSWORD	PASSWORD_T

[ENTER]

SYSTEM Variables		JOINT 10%
\$PARAM_GROUP		49/729
1	\$BELT_ENABLE	FALSE
2	\$CART_ACCEL1	192
3	\$CART_ACCEL2	0
4	\$CIRC_RATE	1
5	\$CONTAXISNUM	0
6	\$EXPENBL	TRUE

[TYPE]

- 7 To return to the upstairs layer, press the PREV key.

7.9 PROGRAM TIMER

A program timer is a timer for measuring the execution time from one line to another in a program. Ten program timers can be used as standard.

A program timer can be started and stopped by using a timer instruction (see Subsection 4.14.3). It also stops at forced termination and upon a halt.

There are two modes program timer, local timer and global timer.

- **Local Timer**
Local timer is stopped at forced termination and upon a halt. Timer count is restarted when program is resumed after a halt.
- **Global Timer**
Global Timer does not depend on state of program. Timer is not stopped at forced termination and upon a halt. Global timer can be measured time including stop time and execution time of some programs. Timer mode is set in the program timer detail screen.

Timer Flag

Timer flag output signal by timer value.

Timer count value < 0: OFF

Timer count value >=0: ON

Output signal port is set in the program timer detail screen. DO, RO and F can be used.

Example:

When DO[1] is set as timer flag, if `TIMER[1]<0`, DO[1] is OFF, if `TIMER[1]>=0`, DO[1] is ON.

NOTE

Global timer and timer flag are available only on 7DA6 series or later. Before that, program timer run as local timer.

The program timer detail screen displays the following information:

- Setting of Program timer mode (Local or Global)
- Program name and line number for which a timer was started most recently
- Program name and line number for which a timer was stopped most recently
- Setting of Timer Flag

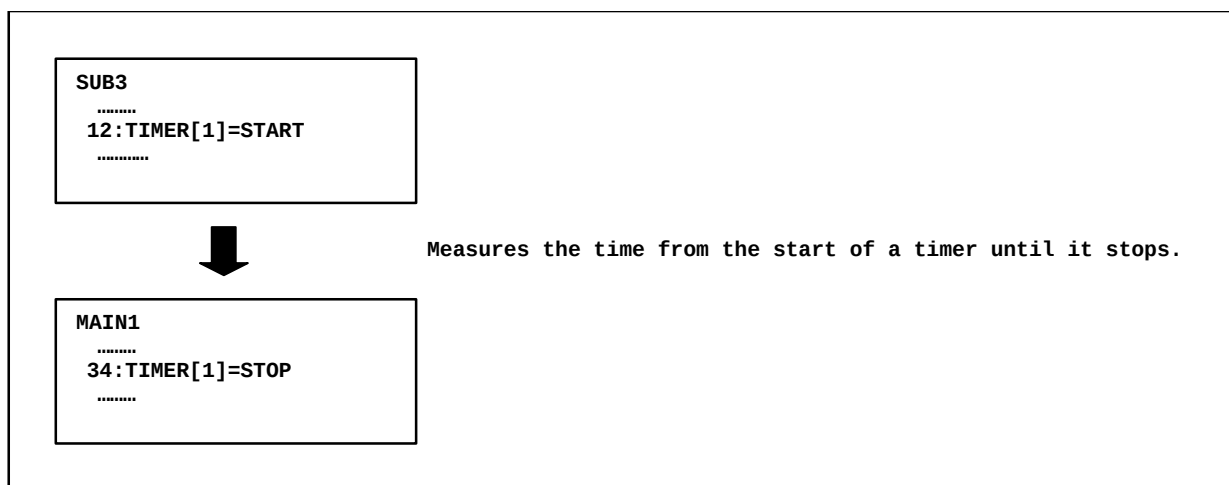


Fig. 7.9 Program Timer Measurement

Program timers are indicated by using 4 STATUS/Prg Timer on the program timer screen.

Procedure 7-8 Displaying program timers

Step

- 1 Press the MENUS key to display the screen menu.
- 2 Press 0 NEXT, and select 4, STATUS.
- 3 Press F1, [TYPE] to display the screen selection menu.
- 4 Select Prg Timer. Then, the program timer screen appears.

PRG TIMER LISTING			JOINT 10%
			1/10
TIMER	count	comment	
1	123.45	(s) [TIMER TEST]	
2 G	0.0	(s) [	
3	0.0	(s) [	
4	0.0	(s) [	
5	0.0	(s) [	
5	0.0	(s) [	
6	0.0	(s) [	
7	0.0	(s) [	
8	0.0	(s) [	
9	0.0	(s) [	
[TYPE] DETAIL			

- 5 If timer mode is set global timer, 'G' is displayed beside Timer number.
- 6 To display detail information, press F2, DETAIL. Then, the program timer detail screen appears.

PRG TIMER DETAIL			JOINT 10%
			1/3
TIMER[1]			
Comment	:	[TIMER TEST]	
Count	:	123.45(sec)	
Local/Global	:	LOCAL	
Start program	:		
[			]
line	:	0	
Stop program	:		
[			]
line	:	0	
timer flag	:	DO [0]	
[TYPE] LISTING			

- 7 To enter a comment, position the cursor to the comment field, and press the enter key. Select the input method, and enter characters using function keys.
- 8 As the start program, a program for which the timer was started most recently is indicated.
As the stop program, a program for which the timer was stopped most recently is indicated.
- 9 To set program timer mode, position the cursor to the Local/Global field, and press F4(GLOBAL) or F5(LOCAL). If timer mode is changed while running, timer is stopped.
- 10 To set output port of timer flag, position the cursor to the timer filed. To select kind of output signal, press F3(CHOICE) and select from DO, RO or F. To input port index, position the cursor to the index number and input number of index. If port index of timer flag is 0, timer flag is disabled.

NOTE

Timer flag setting is enabled after next timer start. If timer flag setting is changed while timer is running, timer flag setting is disabled until timer is stopped.

NOTE

Global timer and timer flag items in detail screen are displayed only on 7DA6 series or later.

7.10 SYSTEM TIMER

A system timer is a timer for indicating the system operation time. The times for four items are indicated. Four types of timers are provided for each operation group.

Table 7.10 System Timer Display

Item	Description
On Power time	Time during which the power to the controller is on
Servo on time	Time during which the system is ready for operation (servo on) after the release of an alarm.
Running time	Program execution time. The halt period is not included.
Waiting time	Time required to execute a wait instruction

To display the system timers, use 4, STATUS Sys Timer on the system timer screen.

Procedure 7-9 Displaying the system timer screen

Step

- 1 Press the MENUS key to display the screen menu.
- 2 Select 4, STATUS on the next page.
- 3 Press F1, [TYPE].
- 4 Select Sys Timer. Then, the system timer screen appears.

SYS TIMER		JOINT 10%
		1/4
GROUP:1		
Timer type	Total(h)	Lap(m)
On Power time:	0.2	0.0[OFF]
Servo on time:	0.2	0.0[OFF]
Running time:	0.0	0.0[OFF]
Waiting time:	0.0	0.0[OFF]
[TYPE] GROUP# ON/OFF RESET		

- 5 To switch between operation groups, press F2, GROUP#, and enter a group number.
- 6 To enable or disable lap time measurement, position the cursor to a desired item, and press F3, ON/OFF to switch the setting.
- 7 To reset the lap time, position the cursor to a desired item, and press F4, RESET.

7.11 EXECUTION HISTORY

The function of the program execution history records the execution history of the program which has been executed or which is being executed at the end, and enables you to see the execution history after the program is finished or paused.

For example, this function enables you to recognize the execution status of the program at power failure after the cold start is done in case that power supply is turned off for any causes while the program is executed.

NOTE

You can not see the execution history of the program which is been executed.

The following information can be referred with the execution history screen.

- Executed program name and line number(The status of the latest executed program is displayed at the first line.)
- Direction of execution
 - FWD : The line was executed by the forward execution.
 - BWD : The line was executed by the backward execution.
- Status of execution
 - Not exec : The line was read but the line has not been executed.
 - Paused : (The program was paused while executing the line)
 - Done : The execution of the line has been completed.
 - Aborted : The program has finished to be executed.

The maximum number of the execution history which can be recorded is 200. The number of record lines can be changed using the maximum number setting screen, selectable from the control start menu. When the maximum number of lines that can be recorded has been reached, subsequent history data recording is performed by automatically erasing the recorded data, starting from the oldest.

Note the following when you use this function:

- When a macro is executed by using the manual function, user key, etc except the program, the execution history of it is not be recorded. When the program assigned to be a macro is executed in the program edit screen, the assigned program name in place of the macro name is recorded as the execution history.
- When the KAREL program is executed, its execution history is not recorded.
- The execution history of the program automatically started at power on is not recorded.

Procedure 7-10 Displaying program execution history

Step

- 1 Press the MENUS key. The screen select menu is displayed.
- 2 Select STATUS from the next page.
- 3 Press F1,[TYPE].
- 4 Select Exec-hist. The execution history screen is displayed.

Execution history				JOINT 10%
Program name	Line.	Dir.	Stat.	
1 PNS0001	3	FWD	Done	
2 PNS0001	6	BWD	Paused	
3 PNS0001	7	FWD	Paused	
4 PNS0001	6	FWD	Done	
5 PNS0001	5	FWD	Aborted	
[TYPE] NEXT ALL_CLR CLEAR				

NOTE

If a single program has been executed, F2, NEXT TASK and F4, ALL CLEAR are not displayed on the execution history screen.

- 5 Only when the displayed status of a program is "Aborted", the execution history can be cleared by pressing SHIFT + F5,CLEAR.
- 6 When multitasking is used, pressing F2 NEXT displays the history of another task.
- 7 When multitasking is used, the execution history of all the tasks can be cleared by SHIFT + F4 ALL_CLR provided Abort is indicated for all the tasks.

7.12 MEMORY USE STATUS DISPLAY

This screen displays the use status and hardware configuration of the controller memory. The display includes the following information:

Table 7.12 (a) Memory Use Status Display(Pools)

Items	Descriptions
TPP	Displays the use of area to hold programs.
PERM	Displays the use of area to hold system variables and registers.
SYSTEM	Displays the use status for a part of the system software.
TEMP	Displays the use status of work area used by system software.

Table 7.12 (b) Memory Use Status Display(Hardware)

Items	Descriptions
F-ROM	Storage capacity of the F-ROM module used in controller
D-RAM	Storage capacity of the D-RAM (RAM) module used in controller
C-MOS	Storage capacity of the C-MOS (RAM) module used in controller

When the [STATUS memory] screen is selected, the following screen appears on the teach pendant. This screen indicates the information collected immediately before it appears.

A list screen displays the use status of program area, permanent area and temporary area.

Memory Status List Screen

STATUS Memory		JOINT	10%
	Total	Available	
Pools	-----	-----	
TPP	1000.0 KB	988.5 KB	
PERM	3002.0 KB	484.9 KB	
TEMP	21784.4 KB	7430.2 KB	
FR	31857.0 KB	12952.0 KB	
Description:			
TPP: Used by .TP, .MR, .JB, .PR			
PERM: Used by .VR, RD:, Options			
TEMP: Used by .PC, .VR, Options			
[TYPE]	DETAIL		HELP

A detailed screen displays use status of all the area mentioned above and displays the hardware information.

Memory Status Detailed screen

STATUS Memory		JOINT	10%
	Total	Free	Lrgst Free
Pools	-----	-----	-----
TPP	1000.0KB	988.5KB	988.4KB
PERM	3002.0KB	484.9KB	483.9KB
SYSTEM	6438.0KB	1.8KB	1.8KB
SHADOW	5760.6KB	5481.1KB	32.0KB
TEMP	21784.4KB	7430.2KB	6766.1KB
FR	31857.0KB	12952.0KB	
Hardware	-----	-----	-----
FROM	32.0 MB (T)	DRAM	32.0MB
SRAM	3.0 MB (CMOS)		
[TYPE]	BASIC		HELP

To move from a list screen to a detailed screen, press F2, DETAIL.

To move from a detailed screen to a list screen, press F2, BASIC.

Explanation of each area is displayed by pressing F5, HELP on both screens. To display the previous screen, press PREV key.

NOTE

This function indicates the use status of the memory. It does not change the use status.

7.13 SAFETY SIGNALS

The safety signal screen indicates the state of safety related signals.

To be specific, the screen indicates whether each safety signal is currently on. It is impossible to change the state of any safety signal on this screen.

Table 7.13 Safety Signals

Safety signal	Description
SOP E-Stop	This item indicates the state of the emergency stop button on the operator's panel. If the EMERGENCY STOP button is pressed, the state is indicated as "TRUE".
TP E-Stop	This item indicates the state of the emergency stop button on the teach pendant. If the EMERGENCY STOP button is pressed, the state is indicated as "TRUE".
Ext E-Stop	This item indicates the state of the external emergency stop signal. If the EMERGENCY STOP signal is asserted, the state is indicated as "TRUE".
Fence Open	This item indicates the state of the safety fence. If the safety fence is open, the state is indicated as "TRUE".
TP Deadman	This item indicates whether the DEADMAN switch on the teach pendant is grasped. If the teach pendant is operable, and the DEADMAN switch is grasped correctly, the state is indicated as "FALSE". If the DEADMAN switch is released or grasped tightly when the teach pendant is operable, the state is indicated as "TRUE". In this case, an alarm occurs, causing the servo power to be switched off.
TP Enable	This item indicates whether the teach pendant is operable. If the teach pendant is operable, the state is indicated as "TRUE".
Hand Broken	This item indicates the state of the hand safety joint. If the hand interferes with a workpiece or anything like this, and the safety joint is opened, the state is indicated as "TRUE". In this case, an alarm occurs, causing the servo power to be switched off.
Overtravel	This item indicates whether the current position of the robot is out of the operation range. If any robot articulation goes out of the operation range beyond the overtravel switch, the state is indicated as "TRUE". In this case, an alarm occurs, causing the servo power to be switched off.
Low Air Alarm	This item indicates the state of the air pressure. The abnormal air pressure signal is connected to the air pressure sensor. If the air pressure is not higher than the specified value, the state is indicated as "TRUE".
Belt Broken	This item indicates the state of the belt. If the belt broken robot digital input is asserted, the state is indicated as "TRUE".
SVOFF Input	This item indicates the state of the SVOFF(Servo Off Signal). If the SVOFF input signal connected to the panel board are open, the state is indicated as "TRUE".
Servo Disconnect	This item indicates the state of the Servo disconnect signal. If the Servo disconnect input signal connected to the panel board are open, the state is indicated as "TRUE".
Non Teacher Enb. Dev.	This item indicates the state of the NTED(Non Teacher Enabling Device) signal. If the NTED input signal connected to the panel board are open in the teach mode(T1,T2), the state is indicated as "TRUE".

NOTE

Refer to the Controller Maintenance Manual for more information on each safety signal.

Procedure 7-11 Displaying the safety signal screen

Step

- 1 Press the MENUS key to display the screen menu.
- 2 Select "0 NEXT", then select "4 STATUS".
- 3 Press F1, [TYPE] to display the screen switching menu.
- 4 Select Safety Signal. Then, You will see a screen similar to the following.

STATUS Safety		JOINT 10%
SIGNAL NAME	STATUS	1/13
1 SOP E-Stop:	TRUE	
2 TP E-Stop:	FALSE	
3 Ext E-Stop:	FALSE	
4 Fence Open:	FALSE	
5 TP Deadman:	TRUE	
6 TP Enable:	TRUE	
7 Hand Broken:	FALSE	
8 Overtravel:	FALSE	
9 Low Air Alarm:	FALSE	
10 Belt Broken:	FALSE	
11 SVOFF Input:	FALSE	
12 Servo Disconnect:	FALSE	
13 Non Teacher Enb. Dev.:	FALSE	
[TYPE]		

7.14 PROGRAM 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. Table 7.14 lists and describes each Program status item.

Use Procedure 7-12 to display the program status screens.

Procedure 7-12 Displaying the Program Status Screens

Steps

1. Press MENUS.
2. Select STATUS.
3. Press F1, [TYPE].
4. Select Prod. Status. The Prod. Status screen will be displayed. You will see a screen similar to the following.

STATUS Program		JOINT 10%
1 Task number:	7	
2 Task name:	RSR0001	
3 Program:	MAIN	
4 Routine:	MAIN	
Line number	6	
Status:	RUNNING	
[TYPE] PREV NEXT		

If you press F2[PREV], you will see previous task number status program screen.

If you press F2[NEXT], you will see next task number status program screen.

Table 7.14 Program Status or Production Status Items

ITEM	DESCRIPTION
Task number	This item displays number of task. If you change this item, overall screen is changed with this item.
Task name	This item displays name of main program that started execution.
Program	This item displays the name of the teach pendant or KAREL program that is currently being executed.
Routine	This item displays the name of the KAREL routine that is being executed, if there is a KAREL program currently being executed.
Line number	This item indicates the line of the teach pendant or KAREL program that is currently being executed.
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]

8 FILE INPUT/OUTPUT

This chapter describes file transfer to and from a communication device.

Contents of this chapter

- 8.1 FILE INPUT/OUTPUT UNITS
- 8.2 SETTING A COMMUNICATION PORT
- 8.3 FILES
- 8.4 SAVING FILES
- 8.5 LOADING FILES
- 8.6 PRINTING FILES
- 8.7 SUBDIRECTORIES
- 8.8 AUTOMATIC BACKUP
- 8.9 IMAGE BACKUP FUNCTION
- 8.10 ASCII UPLOAD FUNCTION
- 8.11 FILE MEMORY

8.1 FILE INPUT/OUTPUT UNITS

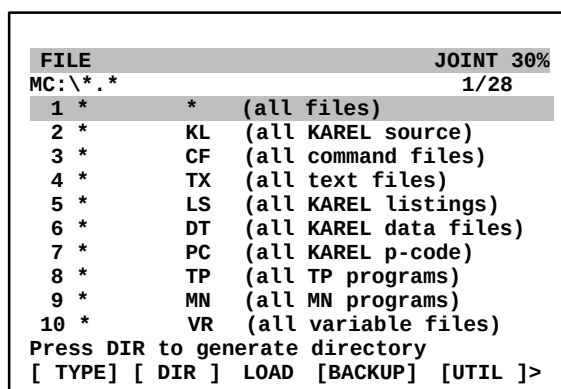
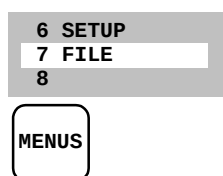
With the robot controller, the various file I/O devices can be used.

The standard setting specifies the use of memory cards. If using another file I/O device, perform the operation below to change the file I/O device setting.

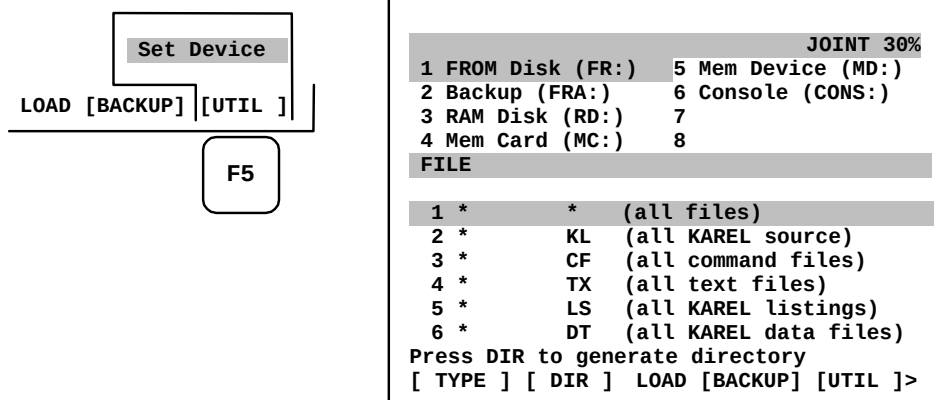
Procedure 8-1 Changing file I/O devices

Step

- 1 Press MENUS to display the screen menu.
- 2 Select FILE. The file screen appears.



- 3 Press F5 UTIL, and select Set Device. Then, the following screen appears:



- 4 Select a file I/O device to be used. An abbreviation for the currently selected file I/O device appears in the upper left part of the screen.

FILE	
MC:\	

Abbreviation	File I/O device
MC:	Memory card
FRA:	Area used for automatic backup of the F-ROM in the controller

File input/output units

The Robot controller allows the use of the following types of storage units to save programs and files.

- Memory card (MC:)
Flash ATA memory card. It is possible to use a Compact Flash card by attaching a PCMCIA adapter to it.
- Backup (FRA:)
Area to which files are saved with auto backup. It can retain information when the power is interrupted, with no backup battery.
- FROM disk (FR:)
Memory area which can retain information without backup battery. There are some important files for the system in the root directory of this storage device. It is possible to backup programs and save any file in this storage device. But, please do not save or delete files in the root directory. If you want to save file in this storage device, please create a subdirectory according to Procedure 8-14 and save the file in the created subdeirectory.
- RAM disk (RD:)
This is the storage device prepared for the special function. Please do not use this storage device.
- MF disk (MF:)
This is the storage device prepared for the special function. Please do not use this storage device.
- FTP (C1: to C8:)
Writes and reads files to and from a FTP server such as a PC connected via Ethernet. It is displayed only if FTP client settings have been made on the host communication screen.
- Memory device (MD:)
The memory device is capable of handling data on the memory of a controller, such as robot programs and KAREL programs, as files.
- Console (CONS:)
Device for maintenance only. It can reference the log file containing internal information.
- USB memory (UT1:)
USB memory mounted to the USB port on *iPendant*.
Software option "USB port on *iPendant*" (A05B-2500-J957) is required to use UT1:.

As standard, the settings are such that a memory card is used.

⚠ CAUTION

In the R-30iA Mate, the floppy disk (FLPY:) units below cannot be used.

- Floppy cassette adapter (A16B-0150-B001)
- Handy file (A16B-0159-B002)

(It cannot be used regardless of whether it is in FANUC or MS-DOS format.)

In the R-30iA Mate, the printer unit FANUC PRINTER (A86L-0001-0103) cannot be used. To print the content of a program, use the ASCII save function to output the content of the program in ASCII format to a file input/output unit, load it into a personal computer, etc., and print it.

Setting up a communication port

The Robot controller provides the communication ports below.

- Port 1 RS-232-C JD17 connector on the main CPU printed circuit board
- Port 2 RS-422 JD17 connector on the main CPU printed circuit board

With the standard settings, the port list screen displays ports 1 only. To enable port 2 (RS-422), perform a control start and change the system variable \$RS232_NPORT to 4 on the system variable setup screen. Port 2 is added to the port list screen so that a communication unit can be set up for port 2.

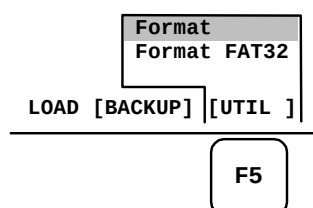
In the R-30iA Mate, the "Handy File", "FANUC Floppy", "Handy F MS-DOS", "Printer" items are not displayed as communication unit settings on the port setup screen.

Format

Some file I/O device can be formatted. Following is the procedure to format file I/O device.

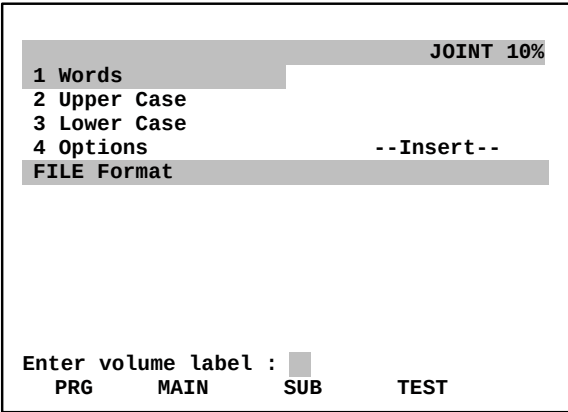
Procedure 8-2 Formatting file I/O device**Step**

- 1 Perform "Procedure 8-1 Changing file I/O devices" to select the desired file I/O device.
- 2 Press F5 UTIL and select "Format" or "Format FAT32". In case that "Format" is selected, the file I/O device is formatted by FAT16 format. In case that "Format FAT32" is selected, the file I/O device is formatted by FAT32 format. The following screen is displayed.



File Format		JOINT 10%
MC:*,*		1/28
Formatting MC:		
***** WARNING *****		
ANY DATA ON THE DISK WILL BE LOST!		
Format disk?		
YES	NO	

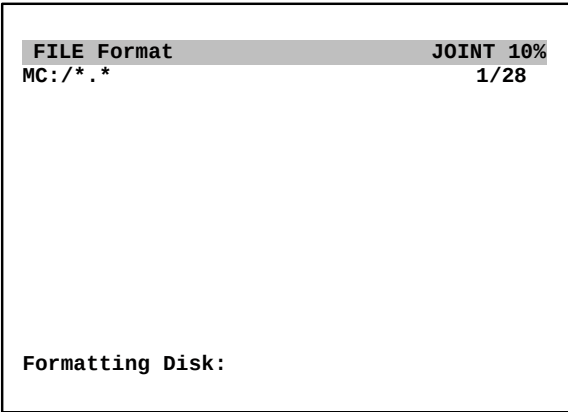
- 3 The confirmation message is displayed. Confirm whether the selected file I/O device is correct, and press F4 YES if file I/O device is correct.
- 4 The following screen is displayed.



CAUTION

All data in the file I/O device are lost by formatting. Please check the device carefully before formatting.

- 5 Enter the volume label using the function keys and the cursor keys. After the volume label is entered, the following screen is displayed, and the formatting is started.



8.1.1 Memory Card

The following memory card can be used.

Type	Recommended product
Compact flash memory card	CompactFlash memory card available from FANUC. A02B-0281-K601, A02B-0213-K211, A02B-0213-K212, A02B-0213-K213, A02B-0213-K214
PC card adapter	PC card adapter available from FANUC. A02B-0236-K150 (PC card size) A02B-0303-K150 (Half size)



A02B-0236-K150



A02B-0303-K150

Fig. 8.1.1(a) PC card adapter

⚠ CAUTION

- 1 Boot software must be newer than 7DA1(V7.10P)36 or 7DA3(V7.30P)17 to use 4GB(or more) Compact Flash(If boot software is 7DA4(V7.40P) or 7DA5(V7.50P), you can use 4GB(or more) CF in any version).
- 2 Boot software must be newer than 7DA5(V7.50P)03 to use 16GB(or more) CF.
- 3 If a memory card other than those recommended is used, a normal operation is not guaranteed, and a bad influence may occur on the controller.
- 4 Please do not power off or remove memory card when writing or reading a file for not to break files in memory card.
- 5 Please re-format memory card when alarm "FILE-064 internal DOS system error:xx" is occurred because files in memory card may be broken.
- 6 A02B-0303-K150 is the compact PC card adapter which size is half of PC card, but note that this PC card adapter cannot be inserted into the card slot on the notebook PC.

⚠ CAUTION

It is recommended that files on a memory card are backed up to storage device to protect the memory card contents against accidental loss.

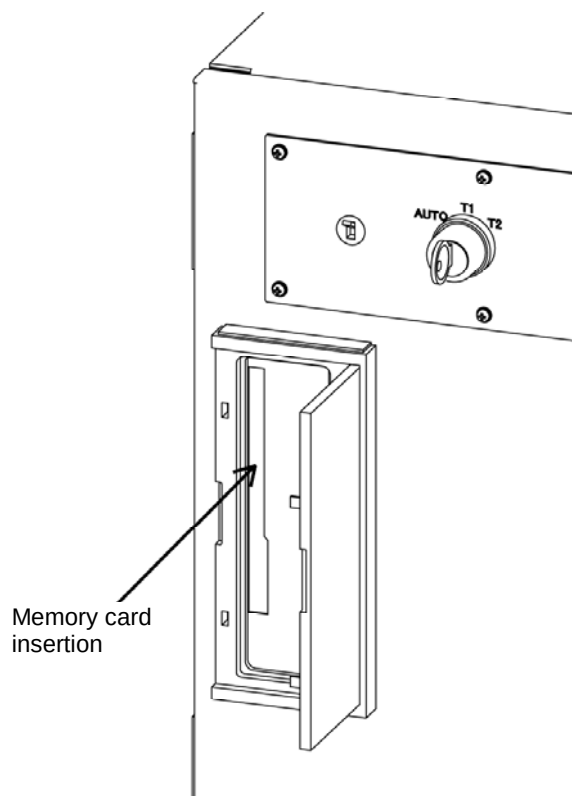


Fig. 8.1.1(b) Memory Card Insertion

When a memory card is to be used, select the memory card according to the description of changing the file I/O devices (see Section 8.1).

8.2 SETTING A COMMUNICATION PORT

The controller performs data transfer to and from external devices through communication ports by performing serial communication via the RS-232C or RS-422 interface.

The following communication ports are used. (Operator's panel/box; see Subsection 2.3.2.)

- Port 1: RS-232-C JD17 connector on the main CPU printed circuit board
- Port 2: RS-422 JD17 connector on the main CPU printed circuit board

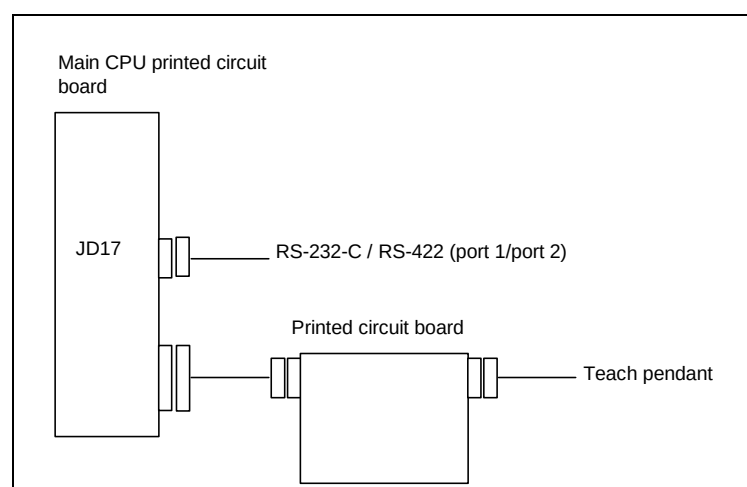


Fig. 8.2 Communication Ports

RS-422

The use of the RS-422 interface has the following advantage:

- While the RS-232-C standard supports a cable length of only about 10 to 20 m, the RS-422 standard allows a cable to be extended to about 50 m.
- RS-422 is less susceptible to noise than RS-232-C.

Application example

- When the communication cable must be routed over a long distance, use the RS-422 interface.

NOTE

The RS-422 interface uses electrical signals that are completely different from those of the RS-232-C interface. When the robot controller and a personal computer are connected via the RS-422 interface, a commercially available RS-422-to-RS-232-C converter may be required since personal computers do not generally have a RS-422 interface.

Communication ports are set by using [6 Setting; port setting] on the port setting screen.

Table 8.2 (a) Standard Communication Devices for Communication Ports

Communication port	Communication device
JD17 RS-232-C P2	Debug Console

Table 8.2 (b) Setting a Communication Port

Items	Descriptions
Device	This item specifies a communication device to communicate with the robot controller. The communication devices that can communicate with the robot controller are listed below: • Host Comm Used when the R-30iA is connected to the host computer to use the data transfer function. • NO Use • KCL/CRT • Debug Console • Factory Terminal • TP Demo Device • Current position • Development • CIMPLI CITY NOTE When the communication device is changed, other settings such as a baud rate are changed to the corresponding standard values. Later on, the user can change each setting as desired.
Speed (Baud rate)	Baud rate is the transmission rate and it is the number of codes which can be transmitted per second. Enter the transmission rate specified for the peripheral unit being used.
Parity bit	To detect an error in data transfer, this item sets a mode of vertical parity check, which adds one extra bit to each transferred character. - Odd : The number of 1's in each transferred character must be an odd number. - Even : The number of 1's in each transferred character must be an even number. - None : No parity check is made. Enter the parity check mode specified for the peripheral unit being used.
Stop bit	This item specifies the number of stop bits to be added at the end of the transferred characters, for data transfer synchronization. - 1 bit : One stop bit is added. - 1.5 bits : One and a half stop bits are added. - 2 bits : Two stop bits are added. Enter the number of stop bits specified for the peripheral unit being used.
Time-out value (sec)	This item sets a maximum time during which control over transfer with a communication device must be exercised. If no data transfer occurs for a specified period of time, the communication line is disconnected.

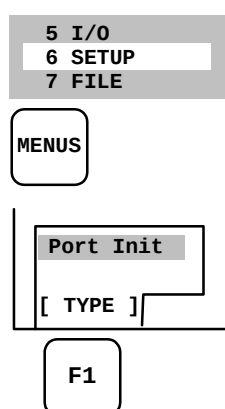
Table 8.2 (c) Standard Settings for Communication Devices

Device	Speed	Parity bit	Stop bit	Time-out value
Host Comm	9600	None	1 bit	None
Factory Terminal	9600	None	1 bit	None
KCL/CRT	9600	None	1 bit	None
TP Demo Device	9600	None	1 bit	None

Procedure 8-3 Setting a communication port

Step

- 1 Press the MENUS key to display the screen menu.
- 2 Select "6, SETUP."
- 3 Press F1, [TYPE] to display the screen change menu,
- 4 Select "Port Init." The port selection screen appears.



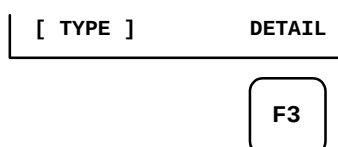
```

SETUP Port Init          JOINT 30%
                        1/1

Connector  Port  Comment
1 JD17 RS-232-C P2: [Maintenance Cons]

[ TYPE ]          DETAIL
  
```

- 5 Move the cursor to a desired connector port field, then press F3 "DETAIL." The port setting screen appears.



```

SETUP Port Init          JOINT 30%
JD17 RS-232-C P2:      1/5
1 Device                [Maintenance Cons]
2 Speed(Baud rate)      [9600]
3 Parity bit             [None]
4 Stop bit               [1bit ]
5 Time out value(sec)   [  0]

[ TYPE ]          LIST  [CHOICE]
  
```

- 6 To set a communication device, move the cursor to the “Device” field, then press F4, [CHOICE]. Select a desired communication device from the menu.

LIST	[CHOICE]
F4	

JOINT 30%	
1 Sensor	5 Maintenance Cons
2 Host Comm	6 Factory Terminal
3 No Use	7 TP Demo Device
4 KCL/CRT	8 -- NEXT --
SETUP Port Init	
1 Device	[Maintenance Cons]
2 Speed(Baud rate)	[9600]
3 Parity bit	[None]
4 Stop bit	[1bit]
5 Time out value(sec)	[0]
[TYPE]	LIST [CHOICE]

- 7 Select a communication device whose settings need to be changed. When the communication device is entered, the standard values are entered in the other setting fields.

1 Sensor
2 Host Comm
3 No Use
4 KCL/CRT
SETUP Port Init

ENTER

SETUP Port Init		JOINT 30%
JD17 RS-232-C P2:		1/5
1 Device	[Host Comm]	
2 Speed(Baud rate)	[9600]	
3 Parity bit	[None]	
4 Stop bit	[1bit]	
5 Time out value(sec)	[0]	
[TYPE]	LIST	[CHOICE]

The other setting fields can be changed field by field. When the “Device” field is changed to another communication device, the standard values for that device are entered in the other setting fields.

NOTE

To indicate that a port is not used, set "No Use" in the corresponding field of communication equipment.

- 8 Upon completion of setting, press F3 “LIST.” The port selection screen appears.

[TYPE]	LIST
F3	

SETUP Port Init		JOINT 30%
		1/1
Connector	Port	Comment
1 JD17 RS-232-C P2:	[Host Comm]	
[TYPE]	DETAIL	

NOTE

When setting the communications device, the error message, "The port was not initialized.", may be displayed and the settings of the port is returned to the previous settings. In this case, confirm the following.

- Has the communication device to be set already been set for another port?

- The same communication device cannot be set for more than one port.
- To use "Host Comm", software option, data transfer, is needed.
 - With the standard settings, the port list screen does not display RS-422 port 2. To enable RS-422 port 2, make a control start (see Appendix B.1.3, "Controlled Start".) and on the system variable screen, change the system variable \$RS232-NPORT from 3 to 4. This causes RS-422 port 2 to be added to the port list screen, allowing the user to set up the communication unit.

8.3 FILES

A file is a unit of data storage in the memory of the robot controller. The following types of file are used mainly:

- Program file (*.TP)
- Default Logic File (*.DF)
- System file (*.SV) Used to store the settings of the system.
- I/O Config Data File (*.IO) Used to store the settings of Input/Output configuration.
- Date file (*.VR) Used to store data such as register data

8.3.1 Program File

A program file (*.TP) contains a sequence of instructions for the robot. These instructions are called program instructions. Program instructions control robot operations, peripheral devices, and each application.

A program file is automatically stored in the memory of the controller. A directory of program files is displayed on the program list screen ("SELECT").

NOTE

The directory of program files is not displayed on the file screen. The file screen enables you to select the external memory device which includes the desired files and manipulate the files.

On the program list screen, operations such as copy, delete, and rename can be performed. (For program operations, see Section 5.5.)

- Registering a program (See Subsection 5.3.1.)
- Deleting a program (See Section 5.5.)
- Copying a program (See Section 5.5.)
- Changing program detail information (including the renaming of a program) (See Section 5.5.)

A program file also includes the information items listed below. These information items can be checked on the program list screen by pressing F5 [ATTR].

- Comment : The function of a program is summarized.
- Write protection : This prevents the program from being modified and deleted.
- Modification Date : Indicates the latest date when the program was modified.
- Program size : The size of the program is indicated in bytes.
- Copy source : The name of the source program from which the program was copied is indicated. When the program is an original program, this information item is blank.

8.3.2 Default Logic File

The default logic file (*.DF) includes the settings of the default logic instruction assigned to each function key (F1 to F4 key) in the program edit screen.

The default logic file is divided to the following kinds:

- DEF_MOTN0.DF Stores the settings of the default motion instructions.F1 key

The following three files store the settings of the default logic instruction assigned to each function key which is displayed in the next page.

- DF_LOGI1.DF F2 key
- DF_LOGI2.DF F3 key
- DF_LOGI3.DF F4 key

8.3.3 System File/Application File

A system file/application file (*.SV) contains a system control program for operating the application tool software, or contains data used with the system. The following types of system file are used:

- SYSVARS.SV : Used to store the settings of the system variables relative to the, reference points, joint operating area and brake control.
- SYSFRAME.SV : User frame and tool frame
- SYSSERVO.SV : Used to store servo parameter data
- SYSMAST.SV : Used to store mastering data
- SYSMACRO.SV : Used to store the settings of the macro command.
- FRAMEVAR.VR : Used to store the settings of the reference position which is used at setting the frame, comments, etc. Frame data is stored to SYSFRAME.SV.

8.3.4 Data File

Data file (*.VR,*.IO,*.DT) is the file which stores the data used by the system. The following kinds are in the data file:

- Data file (*.VR)
 - NUMREG.VR : Used to store the data of the register.
 - POSREG.VR : Used to store the data of the position register.
 - STRREG.VR : Used to store the data of the string register.
 - PALREG.VR : Used to store the data of the palletizing register. (Only when the palletizing option is installed.)
- I/O configuration data file (*.IO)
 - DIOCFGSV.IO : Used to store the settings of the I/O assignment.
- Robot setting data file (*.DT)

This file is used to store those settings that are made on the robot setting screen.
The file name varies depending on the model.

8.3.5 ASCII File

An ASCII file (*.LS) is a file of ASCII format. To read an ASCII file, the optional function for ASCII upload is required. The contents of an ASCII file can, however, be displayed and printed using a personal computer.

8.4 SAVING FILES

The function of saving files stores the data which exists in the RAM memory in the controller to the external storage device. The following screens on the teach pendant can be used to save the files.

- Program selection screen: A specified program is saved to a storage device as program files.
- File screen: The specified program file, system file, etc can be saved to a storage device. The following files can be saved: When a batched save operation is executed, program files, system files, and application files can all be saved at the same time.
 - Program file
 - System file
 - Application file

- Default logic file
- Application TP program file
- Error log file
- Diagnostic file
- Vision data file
- ASCII program file
- “5 SAVE” in the function menu: It is possible to preserve it on the storage device as program file and a system file, etc. of the program and the data, etc. displayed on the screen. The following files can be preserved:
 - Program file
 - System file
 - Data file
 - Application file
 - Default logic file

8.4.1 Saving with Program List Screen

Program list screen enables you to save the specified program as the program file.

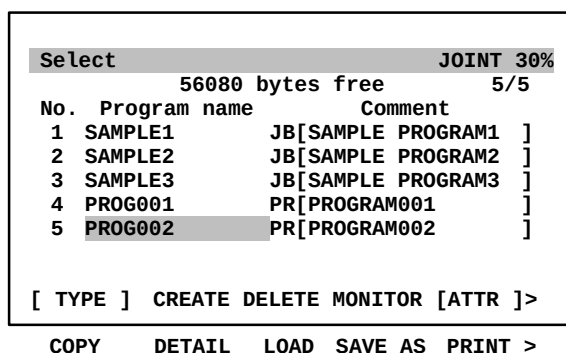
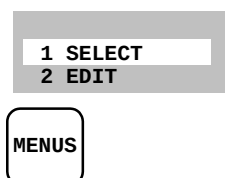
Procedure 8-4 Requirements for saving program files

Condition

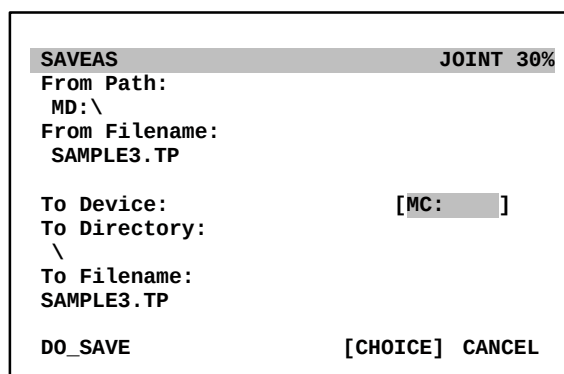
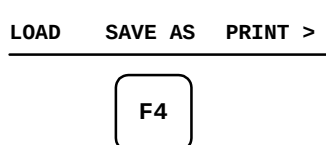
- The file input/output device is set correctly. (See Section 8.1.)

Step

- 1 Press the MENUS key to display the screen menu.
- 2 Select NEXT and then select “1 SELECT” on the next page. The program list screen appears.



- 3 Press NEXT,>, and press F4,SAVE on the next page. The program save screen appears.



- 4 Press F4 [CHOICE] and select the device to save the program. If you want to save the program to the subdirectory, enter the name of subdirectory to the item "To Directory". If you want to change the file name, enter new file name to the item "To Filename". Then, press F1 DO_SAVE to save the program.

NOTE

When you enter the subdirectory name to the item "To Directory", please enter "\ " to the end of subdirectory name.

If the designated subdirectory does not exist, when F1 DO_SAVE is pressed, the error "File not found" occurs and the program is not saved.

- 5 When a program having a same name as you want to save exists in the device, the following message is displayed. In order to overwrite the file, press F4 YES.

Overwrite?	YES	NO

8.4.2 Saving all the Program Files Using the File Screen

File screen enables you to save the program file or system file which is saved in the RAM memory in the device.

The following files can be saved by pressing F4, BACKUP:

Table 8.4.2 The saved files by each item in F4 BACKUP menu

Item in BACKUP menu	Saved files	Description
System files	System files (*.SV, *.VR)	Used to store the following files. - System variable file (SYSVARS.SV) - Servo parameter file (SYSSERVO.SV) - Mastering data file (SYSMAST.SV) - Macro data file (SYSMACRO.SV) - Frame setup file (FRAMEVAR.VR) - Frame data file (SYSFRAME.SV)
	Data files (*.IO, *.VR, *.DT)	Used to store the following files. - Register data file (*.VR) - I/O configuration data file (*.IO) - Robot setting data file (*.DT)
TP programs	Program files (*.TP)	Used to store all the programs file which has contents of programs.
	Default logic files (*.DF)	Used to store the settings of default logic instructions.
Application	Application files (*.SV)	Used to save the settings of an application. (Example: SYSSPT.SV)
Applic. TP	Application TP program files (*.TP, *.DF, *.MN)	Used to store the program for the specific application.
Error log	Error log files (*.LS)	Used to store the alarm log in ASCII format.
Diagnopsitc	Diagnostic files (*.DG)	Used to store the specific diagnostic file in ASCII format.
Vision data	Vision data files (*.VD)	Used to store the vision data, settig of camera and camera calibration data and so on. (* This function can be used in the software series 7DA5 or later.)
All of above	All of above files	Used to store all of aobve files. (The following files are not included.)

Item in BACKUP menu	Saved files	Description
Maintenance	Diagnostic log files	Used to store the internal data while some problems occur. For detail about this function, refer to 9.29 DIAGNOSTIC LOG GETTING FUNCTION. (*This function can be used in the following software.
ASCII program	ASCII file (*.LS)	Used to store all programs in ASCII format. (* This function can be used in the following software. Series 7DA1 version 45 or later Series 7DA3 version 35 or later Series 7DA4 version 19 or later Series 7DA5 version 09 or later)
Image	Image files (*.IMG)	Used to store the images of F-ROM and S-RAM memories of the controller.

To interrupt the saving, press the PREV key while saving.

FCTN

2 RESTORE/BACKUP

NOTE

At control start time, F4 is set to RESTOR instead of BACKUP. When RESTORE/BACKUP is selected from the auxiliary menu, BACKUP is displayed.

Procedure 8-5 Saving files using the file screen

Condition

- The file input/output device is set correctly. (See Section 8.1.)

Step

- 1 Press the MENUS key to display the screen menu.
- 2 Select "7 FILE." The file screen appears.

6 SETUP
7 FILE
8

MENUS

```

FILE                               JOINT 30%
MC:\*. *                           1/28
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)
Press DIR to generate directory
[ TYPE ] [ DIR ] LOAD [BACKUP] [UTIL ]>
DELETE COPY DISPLAY >

```

Saving program files

- 1 Press F4 "BACKUP", then select "TPE programs."

1 System files	2 TPE programs	3 Application
LOAD	BACKUP	[UTIL]>
F4		

FILE	JOINT 30%
	1/13
8 * TP (all TP programs)	
9 * MN (all MN programs)	
10 * VR (all variable files)	
11 * SV (all system files)	
Save MC:\SAMPLE1.TP ?	
EXIT	ALL YES NO

- F2, EXIT Ends saving program files
 - F3,ALL Saves all the program file and default logic instruction file.
 - F4 YES Saves the specified file (program, default logic instruction).
 - F5,NO Does not save the specified file (program, default logic instruction). After that, the system asks whether the next program file is to be saved.
- 2 Select the desired function key. In this case, program file (*.TP) is saved in the device.

EXIT	ALL
F3	

Saving MC:\SAMPLE1.TP, please wait...

- 3 When the file which has the same name as you specified already exists in the device, the following message is displayed.

MC:\SAMPLE.TP already exists
OVERWRITE SKIP CANCEL

- F3,OVERWRITE The specified file is overwritten and saved.
- F4,SKIP Does not save the specified file.
- F5,CANCEL Ends saving files.

Saving the system file.

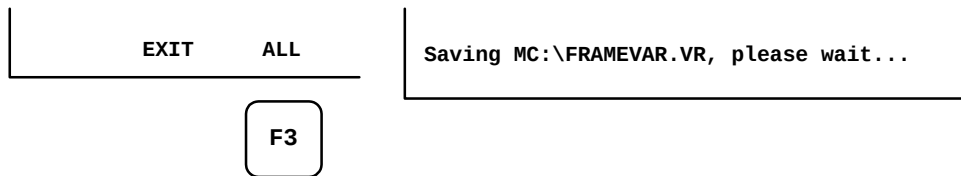
- 1 Press F4,SAVE and select System files.

1 System files	2 TPE programs	3 Application
LOAD	BACKUP	[UTIL]>
F4		

FILE	JOINT 30%
	1/13
8 * TP (all TP programs)	
9 * MN (all MN programs)	
10 * VR (all variable files)	
11 * SV (all system files)	
Save MC:\FRAMEVAR.VR ?	
EXIT	ALL YES NO

- F2, EXIT Ends saving system files
- F3,ALL Saves all of the system files and the data files.
- F4 YES Saves the specified file (system, data).
- F5,NO Does not save the specified file (system, data). After that, the system asks whether the next system file or data file is to be saved.

- 2 Select the desired function key. In this case, system file (*.SV) or data file (*.DT) is saved in the device.



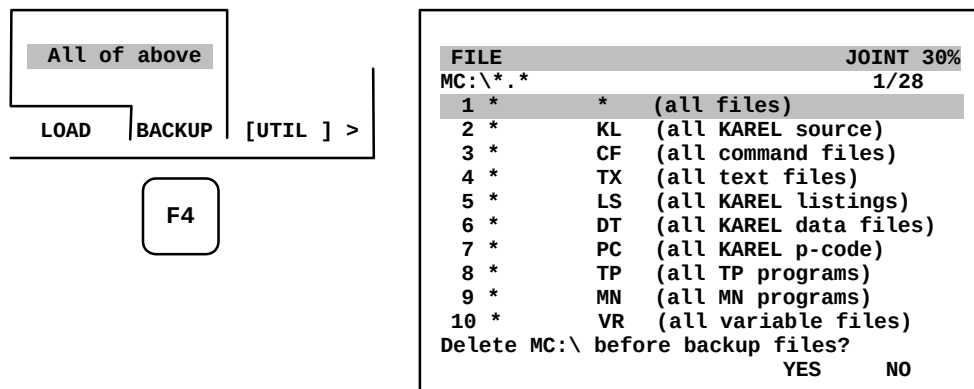
- 3 When the file having the same name as you want to save exists in the device, the following message is displayed.

MC:\FRAMEVAR.VR already exists
OVERWRITE SKIP CANCEL

- F3,OVERWRITE The specified file is saved by overwriting.
- F4,SKIP The specified file is not saved.
- F5,CANCEL Saving files is ended.

Batched save

- 1 Press F4 BACKUP, then select ALL of above.



NOTE

In case that the backup directory is the route directroy of the storage device which format type is FAT16, more than 240 files cannot be saved. In this case, please create new subdirectory to backup files. Please go to step 4.

- 2 When F4, YES is selected at Step1, the following confirmation message appears.

Delete MC:\ and backup all files?
YES NO

- When F4, YES is selected, all the files in the external memory unit are erased, then all the data is saved. Processing is interrupted using the PREV key. An interrupt occurs once the current file has been processed.

**CAUTION**

Before a batched save operation, all files in the external memory unit are erased.
Before executing a batched save operation, check the files in the external memory unit.

- 3 When F5, NO is selected at Step1, the following confirmation message appears.

No delete but backup all files to MC:\?	
YES	NO

- When F4, YES is selected, all the files in the external memory unit are kept and all the data is saved.

YES	NO

F4

In case that the files cannot be saved to the route directory of the storage device, perform the step 4 and 5.

- 4 Following screen is displayed.

Can't use root dir., create dir. backupxx?	
YES	NO

- When F4, YES is selected, following screen is displayed.

No delete but backup all files to MC:\backupxx\?	
YES	NO

- 5 When F4 YES is selected, all the data is saved to the subdirectory created in the step 4.

Maintenance data

Associated with maintenance data, please refer to diagnostic log.

8.4.3 Saving with a Function Menu

By selecting SAVE from a function menu, the data of a screen currently displayed can be saved into the device. The data of the following screens can be saved:

- Program edit screen Program file (*.TP)
- System variable screen System variable file (SYSVARS.SV)
- Positioning screen Mastering data file (SYSMAST.SV)
- Macro instruction setting screen Macro data file (SYSMACRO.SV)
- Frame setup screen Frame setup data file (FRAMEVAR.VR)
- Register screen Register data file (NUMREG.VR)
- Position register screen Position register data file (POSREG.VR)
- String register screen String register data file (STRREG.VR)
- Pallet register screen Pallet register data file (PALREG.VR)
- I/O screen I/O configuration data screen (DIOCFGSV.IO)
- Edit screen for each default logic instruction. Each default logic instruction. (*.DF)

Procedure 8-6 Saving with a function menu

Condition

- The file input/output device is set correctly. (See Section 8.1.)

Saving program files.

Step

- 1 Display the program edit screen or the program selection screen.

Select		JOINT 30%
49828 bytes free		1/5
No.	Program name	Comment
1	PROG001	PR[PROGRAM001]
2	PROG002	PR[PROGRAM002]
3	SAMPLE1	JB[SAMPLE PROGRAM1]
4	SAMPLE2	JB[SAMPLE PROGRAM2]
5	SAMPLE3	JB[SAMPLE PROGRAM3]
[TYPE] CREATE DELETE MONITOR [ATTR]>		

- 2 To display a function menu, press the FCTN key.

- 3 Select "2 SAVE."

A selected program file is saved.

```

1 QUICK/FULL MENUS
2 SAVE
3 PRINT SCREEN

```

FCTN

- 4 When the program having the same name as you want to save exists in the device, the file can not be saved.

File already exists

- 5 When the device is filled with the files, exchange the device and press F4,CONTINUE. All the data being saved at exchanging device is saved into the exchanged device.

Disk is full, change to empty disk
CONTINUE CANCEL

Saving other files.

Step

- 1 Display the screen you want to save.

DATA Registers		JOINT 30%
		1/200
R[1:	COUNTER1]=12	
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
Enter value		
[TYPE]		

- 2 Display the function menu by pressing the FCTN key.
- 3 Select "2 SAVE."
The contents of the screen being displayed is saved.

1 QUICK/FULL MENUS
2 SAVE
3 PRINT SCREEN



- 4 When the file having a same name exists in the device, the file is overwrite.
- 5 When the device is filled with the files, exchange the device and press F4,CONTINUE. All the data being saved at exchanging devices is saved into the exchanged device.

FLPY-005 Disk is full	
DATA Registers	JOINT 30%

8.4.4 File Manipulation

On the file screen, files saved on a device can be listed and a file can be copied or deleted.

Procedure 8-7 File manipulation

Condition

- The file input/output device is set correctly. (See Section 8.1.)

Step

- 1 Press the MENUS key. The screen menu is displayed.
- 2 Select 7 FILE. The file screen is displayed.

6 SETUP
7 FILE
8

MENUS

```

FILE                               JOINT 30%
MC:\*. *                           1/28
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)
Press DIR to generate directory
[ TYPE ] [ DIR ] LOAD [BACKUP] [UTIL ]>

DELETE COPY DISPLAY >

```

Displaying the list of files.

- 3 Press F2,[DIR].

```

Directory Subset                     JOINT 30%
1 *. *                             5 *.LS
2 *.KL                             6 *.DT
3 *.CF                             7 *.PC
4 *.TX                             8 -next page-
FILE

```

- 4 Select ”*. *”(all files).The list of the files being saved into the device is displayed.

```

FILE                               JOINT 30%
MC:\*. *                           1/19
1 PROGRAM1                         TP 768
2 PROGRAM2                         TP 384
3 PROGRAM3                         TP 768
4 SYSVARS                         SV 25600
5 SYSMACRO                        SV 324
6 NUMREG                          VR 708
7 DIOCFGSV                        IO 476
8 * * (all files)
[ TYPE ] [ DIR ] LOAD [BACKUP] [UTIL ]>

```



WARNING

Before a program set as a macro instruction is copied from a controller onto another controller, the macro setting screens of the two controllers should be compared. Be sure that the lists of the two controllers match. The program should be copied only when the lists match. Otherwise, an unpredictable result would occur that could injure personnel or damage equipment.

Deleting files

- 5 Select the file you want to delete and press F1,DELETE.

2	PROGRAM2	TP	384
3	PROGRAM3	TP	768
4	SYSVAR5	SV	25600

Delete MC:\PROGRAM3.TP? YES NO

DELETE COPY DISPLAY

F1

NOTE

Deleting a program from memory of the controller does not automatically delete the identical program from a device.

**CAUTION**

The operator should check that the current device has the file to be deleted. Otherwise, a wrong file can be deleted.

- 6 Press F4, YES. The file will be deleted.

FILE		JOINT 30%	
MC:*,*		3/19	
1	PROGRAM1	TP	768
2	PROGRAM2	TP	368
3	<Deleted>		
4	SYSVAR5	SV	25600
5	SYSMACRO	SV	324
Deleted file MC:\PROGRAM3.TP			
DELETE COPY DISPLAY [INSTALL] [VIEW]>			

8.4.5 ASCII save

ASCII save function saves the program which is saved in binary (internal) format to the external memory device in ASCII format. This ASCII format is like the output of the printer.

The programs, which are saved to the external memory device using this function, can be loaded to the personal computer and can be edited by it. In order to load the program which is saved to the external memory device in ASCII format directly into the robot controller, the software option is required.

File input/output device

The ASCII save function saves a file of ASCII format to a file input/output device selected according to Section 8.1.

Procedure 8-8 Execution of ASCII save function

Condition

- The file input/output device is set correctly. (See Section 8.1.)

Step

- 1 Press the MENU key to display the screen menu.
- 2 Select SELECT on the next page. The program list screen appears.

Select			JOINT 30%
49828 bytes free			3/5
No.	Program name	Comment	
1	SAMPLE1	JB[SAMPLE PROGRAM1]	
2	SAMPLE2	JB[SAMPLE PROGRAM2]	
3	SAMPLE3	JB[SAMPLE PROGRAM3]	
4	PROG001	PR[PROGRAM001]	
5	PROG002	PR[PROGRAM002]	
[TYPE] CREATE DELETE MONITOR [ATTR]>			

- 3 Press PRINT on the next page. The program print screen appears.

LOAD	SAVE	PRINT >
F5		
1 Words 2 Uppser Case 3 Lower Case 4 Options Select JOINT 30% --Insert-- --- Print Teach Pendant Program --- Program Name: SAMPLE3 Enter program name RSR PNS STYLE JOB TEST		

- 4 Enter the name of the program to be saved with the ASCII save function, then press ENTER.

Select --- Print Teach Pendant Program --- Program Name: SAMPLE3	ENTER

- 5 The specified program is saved with the ASCII save function. A file is saved with extension LS. In the same way, print data can be output as a file of ASCII format by print operation based on the function menu (Subsection 8.6.2).

8.5 LOADING FILES

Loading files is to load the files being saved in the file input/output device to the memory in the controller. The files can be loaded with the following screens on the teach pendant:

- Program selection screen -The specified program file is loaded from the device as the program.
- File screen -The specified program files and system files can be loaded. The following files can be loaded.
 - Program file (*.TP or *.MN)
 - Default logic instruction (*.DF)
 - System file (*.SV)
 - Data file (*.VR,*.IO)

NOTE

Selecting F4, RESTOR on the file screen in the control start (not control start 2) enables batched read. Files stored in an external memory unit are read in the following order:

- 1 Files having the same names as those saved when System files is selected
 - 2 Files having the same names as those saved when Application is selected
 - 3 *.TP, *.DF, and *.MN files in the external memory unit
- *.SV and *.VR files are automatically read by selecting Convert=YES.

**CAUTION**

If a program having the same name exists during a program read operation, the existing program is overwritten automatically.

8.5.1 Loading Using the Program List Screen

In the program list screen, the specified program file can be loaded from a device.

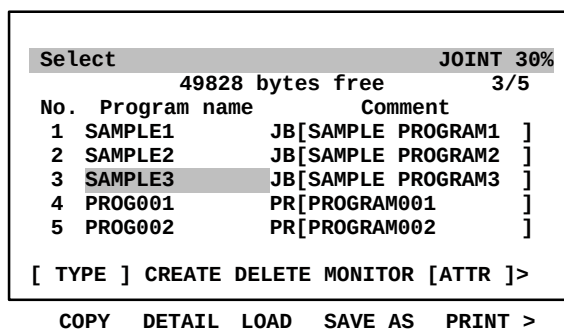
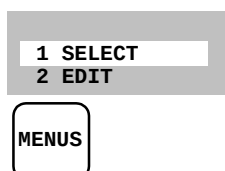
Procedure 8-9 Loading a program file using the program list screen

Condition

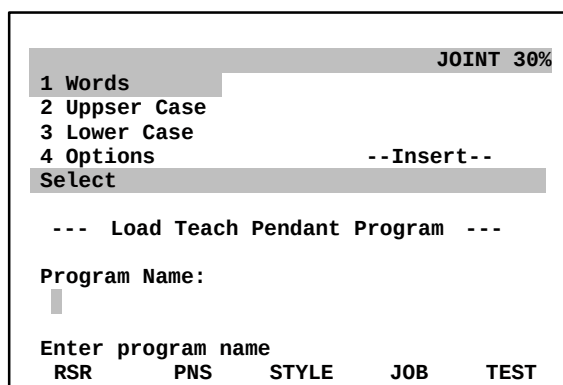
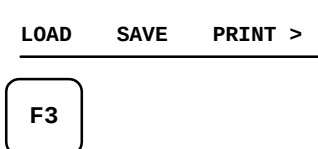
- The file input/output device is set correctly. (See Section 8.1.)

Step

- 1 Press MENUS key to display the screen menu.
- 2 Select "0 – NEXT –" and select "1 SELECT" from the next page.
Program list screen is displayed.



- 3 Press "NEXT",>, and press F3,LOAD, on the next page. Program load screen is displayed.



- 4 Enter the name of a program to be loaded, then press the ENTER key.

Program Name: PROG001	ENTER
Enter program name RSR PNS	

NOTE

Do not include a file extension in the program name.

A specified program is loaded from a device.

- 5 When the program having the same name as you want to load exists in the memory, the following message is displayed.

PROG001 already exists, select function OVERWRITE CANCEL

- OVERWRITE Loads the new file and overwrites it.

8.5.2 Loading a Specified Program File Using the File Screen

In the file screen, the specified file is loaded from the device to the memory.

The following files can be read:

- Program file (*.TP or *.MN) -Program file having contents of the program can be loaded.
- Default logic file (*.DF) -Default logic file having the settings of the default logic instruction can be loaded. The method of loading is the same as the program file.
- Data file (*.VR,*.IO) -The following data file can be loaded.
 - Register data file (NUMREG.VR)
 - Position register data file (POSREG.VR)
 - String register data file (STRREG.VR)
 - Palltizing register data file (PALREG.VR)
 - I/O config data file (DIOCFGSV.IO)
- System file (*.SV) -The following system files can be loaded. However, system files can be loaded only at the controlled start.(See Subsection B.1.3, "Controlled start")
 - System variable file (SYSVARS.SV)
 - Servo parameter file (SYSSERVO.SV)
 - Mastering data file (SYSMAST.SV)
 - Macro data file (SYSMACRO.SV)
 - Frame setup screen data file (FRAMEVAR.VR)
 - Frame data file (SYSFRAME.SV)

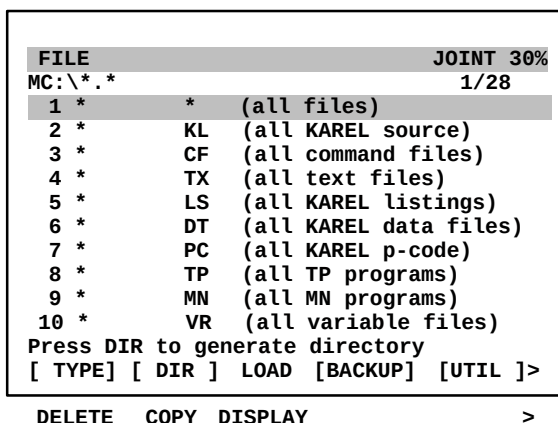
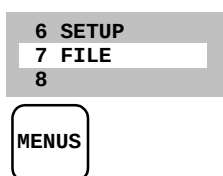
Procedure 8-10 Loading a program file using the file screen

Condition

- The file input/output device is set correctly. (See Section 8.1.)

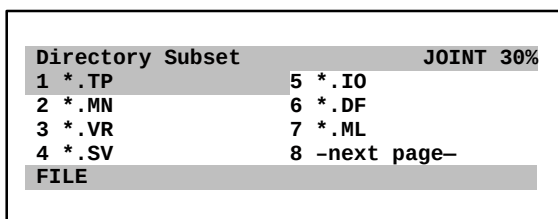
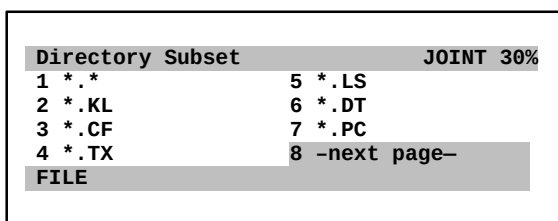
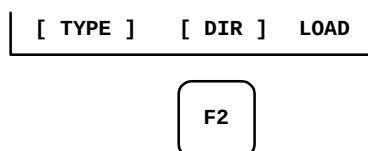
Step

- 1 Press the MENUS key to display the screen menu.
- 2 Select "7 FILE" to display the file screen.

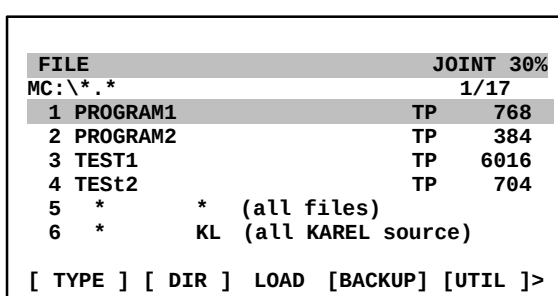
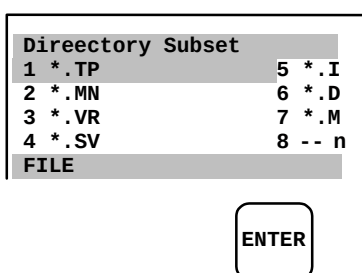


Loading a program file

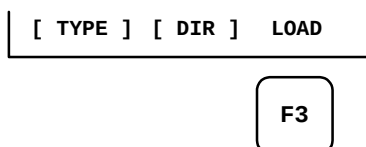
- 3 Press F2 "DIR."



- 4 Select "*.TP" (program file). The directory of program files stored on the device is displayed.



- 5 Move the cursor to the program file you want to load and press F3,LOAD.



Loading PROGRAM1.TP, Prev to exit.

Selected program is loaded from the device.

Loaded PROGRAM1.TP

- 6 If a program with the same name already exists in the RAM, the following indication is provided:

PROGRAM1.TP already exists
OVERWRITE SKIP CANCEL

- OVERWRITE Loads the new file and overwrites it.
- SKIP Skips to the next file.

- 7 If you want to load all program files, select "*.TP" and press F3,LOAD. When the PREV key is pressed, the operation is interrupted after the current a file is loaded.

FILE		JOINT 30%
		8/13
8 *	TP (all TP programs)	
9 *	MN (all MN programs)	
10 *	VR (all variable files)	
11 *	SV (all system files)	

Press DIR to generate directory
[TYPE] [DIR] LOAD [BACKUP] [UTIL]>

[TYPE] [DIR] LOAD

F3

Loading a data file

- 8 Press F2, DIR. Sub-menu is displayed.

[TYPE] [DIR] LOAD

F2

Directory Subset		JOINT 30%
1 *.TP	5 *.IO	
2 *.MN	6 *.DF	
3 *.VR	7 *.ML	
4 *.SV	8 -next page-	

FILE

- 9 Select "*.VR" (variable data file). The directory of variable data files stored on the device is displayed.

Directory Subset	
1 *.TP	
2 *.MN	
3 *.VR	
4 *.SV	
FILE	

ENTER

FILE		JOINT 30%
MC:*. *		1/15
1 NUMREG	VR	868
2 POSREG	VR	1024
3 *	(all files)	
4 *	KL (all KAREL source)	
5 *	CF (all command files)	
6 *	TX (all text files)	

[TYPE] [DIR] LOAD [BACKUP] [UTIL]>

- 10 Select a program file you want to load and press F3, LOAD.

[TYPE] [DIR] LOAD	Loading NUMREG.VR, Prev to exit.

F3

The specified program is loaded from the device. Loaded data is set as the current data.

Loaded NUMREG.VR

- 11 If you want to load all the file which has the same extension, Select "*.VR", "*.IO", etc and press F3,LOAD.

FILE	JOINT 30%
	10/13
8 * TP (all TP programs)	
9 * MN (all MN programs)	
10 * VR (all variable files)	
11 * SV (all system files)	

Press DIR to generate directory
 [TYPE] [DIR] LOAD [BACKUP] [UTIL]>

Loading system variable files

Condition

- Turn on the power by controlled start (See Subsection B.1.3, "Controlled start") The following simplified system starts.

SYSTEM Variables	CNTRL START MENU
	1/98
1 \$AP_MAXAX	536870912
2 \$AP_PLUGGED	4
3 \$AP_TOTALAX	16777216
4 \$AP_USENUM	[12] of Byte
5 \$AUTOINIT	2
6 \$BLT	19920216

[TYPE]

- 12 Press the MENUS key, then select "5 File." The file screen appears.

4 Variables
 5 File
 6 ALARM

MENUS

FILE	CTRL START MENU
MC:*. *	1/28
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)	

Press DIR to generate directory
 [TYPE] [DIR] LOAD [RESTOR] [UTIL]>

- 13 Press F2 “DIR” to display the submenu.

[TYPE]	[DIR]	LOAD

F2

Directory	Subset	CTRL	START	MENU
1	*.TP	5	*.IO	
2	*.MN	6	*.DF	
3	*.VR	7	*.ML	
4	*.SV	8	-next page-	
FILE				

- 14 Select “*.SV” (system variable data file). The list of the system files which are saved in the device is displayed.

FILE	CTRL	START	MENU
MC:*.*			1/17
1 SYSVARS	SV	78827	
2 SYSSERVO	SV	384	
3 SYSMAST	SV	6016	
4 SYSMACRO	SV	704	
5 *	*	(all files)	
6 *	KL	(all KAREL source)	

[TYPE] [DIR] LOAD [RESTOR] [UTIL]>

- 15 Select the file you want to load and press F3,LOAD.
When you press the PREV key while the system files are loaded by selecting “*.SV”, loading is kept on until the file being loaded at pressing the PREV key is finished to be loaded.

[TYPE]	[DIR]	LOAD

F3

- 16 When a system file is read, it is necessary to specify whether conversion is to be performed to maintain compatibility with the old system. Normally, select YES.

Convert ?	YES	NO

F4

- 17 Select “1 START (COLD)” from the function menu. The system is cold started.

1 START (COLD)
2

FCTN

Batched read

Step

- 1 Select a file screen in the control start (not control start 2).
- 2 Select F4 RESTOR.
- 3 A message asking the user for confirmation appears on the prompt line.

TEST	LINE 0		
FILE	CTRL START MENU		
MC:*,*			1/28
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)	
Restore from Memory card(OVRWRT)?			
	YES		NO

- 4 Select F4 YES. Then, the read operation starts.
Processing is interrupted using the PREV key. An interrupt occurs once the current file has been processed.

8.6 PRINTING FILES

Printing files is outputting the contents of a program, a data file, the contents of system variables, and so on to a file I/O device selected as described in the file I/O device selection section (→ 8.1), as ASCII format files. The image being displayed on the teach pendant screen can also be output (print screen).

Printing files can be executed by the following screens.

- Program selection screen: Can print the program files.
- "4 PRINT" on the second page of the FCTN menu: Can print the contents of the following screens:
 - Program edit screen: Program detail information and contents of program.
 - System variable screen: System variable data

Procedure 8-11 Printing files using program list screen

Condition

- The file I/O device is in an output enabled status.

Printing out a program file using the program list screen

Step

- 1 Press the MENUS key to display the screen menu.
- 2 Select "1 SELECT" on the next page.
The program list screen appears.

Select		JOINT 30%
56080 bytes free		5/5
No.	Program name	Comment
1	PROG001	PR[PROGRAM001]
2	PROG002	PR[PROGRAM002]
3	SAMPLE1	JB[SAMPLE PROGRAM1]
4	SAMPLE2	JB[SAMPLE PROGRAM2]
5	SAMPLE3	JB[SAMPLE PROGRAM3]
[TYPE] CREATE DELETE MONITOR [ATTR]>		
COPY DETAIL LOAD SAVE AS PRINT >		

- 3 Press F5 “PRINT” on the next page.
The program print screen appears.

LOAD	SAVE	PRINT >																														
F5																																
<table border="1"> <thead> <tr> <th colspan="2">1 Words</th> <th>JOINT 30%</th> </tr> </thead> <tbody> <tr> <td>2</td> <td>Uppser Case</td> <td></td> </tr> <tr> <td>3</td> <td>Lower Case</td> <td></td> </tr> <tr> <td>4</td> <td>Options</td> <td>--Insert--</td> </tr> <tr> <td colspan="2">Select</td> <td></td> </tr> <tr> <td colspan="3">--- Print Teach Pendant Program ---</td> </tr> <tr> <td colspan="3">Program Name:</td> </tr> <tr> <td colspan="3">SAMPLE3</td> </tr> <tr> <td colspan="3">Enter program name</td> </tr> <tr> <td>RSR</td> <td>PNS</td> <td>STYLE JOB TEST</td> </tr> </tbody> </table>			1 Words		JOINT 30%	2	Uppser Case		3	Lower Case		4	Options	--Insert--	Select			--- Print Teach Pendant Program ---			Program Name:			SAMPLE3			Enter program name			RSR	PNS	STYLE JOB TEST
1 Words		JOINT 30%																														
2	Uppser Case																															
3	Lower Case																															
4	Options	--Insert--																														
Select																																
--- Print Teach Pendant Program ---																																
Program Name:																																
SAMPLE3																																
Enter program name																																
RSR	PNS	STYLE JOB TEST																														

- 4 Enter the name of a program file to be printed out, then press the ENTER key.

Program Name:	SAMPLE3
Enter program name	ENTER
RSR	PNS

- 5 The specified program file is printed out. To stop printing, press the PREV key.

Procedure 8-12 Printing using the function menu

Program printing

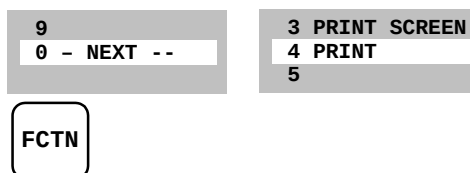
Condition

- The program edit screen is displayed.

SAMPLE1		JOINT 10%
		1/6
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[5] 100% FINE	
[End]		
POINT		TOUCHUP >

Step

- 1 Press the function key to display the function menu.
- 2 Press 0 NEXT, and select 4 PRINT.



- 3 The currently displayed program is printed.
To interrupt printing, press PREV key.

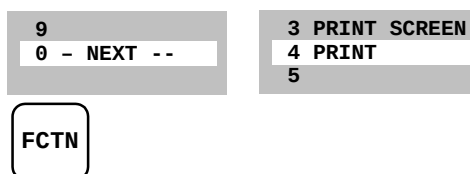
System variable printing**Condition**

- The system variable screen is displayed.

SYSTEM Variables		JOINT 10%
		1/98
1	\$AP_MAXAX	536870912
2	\$AP_PLUGGED	4
3	\$AP_TOTALAX	16777216
4	\$AP_USENUM	[12] of Byte
5	\$AUTOINIT	2
6	\$BLT	19920216
[TYPE]		

Step

- 1 Press the function key to display the function menu.
- 2 Press 0 - NEXT -, then select 4 PRINT.



- 3 A list of system variables is printed.
- 4 To print only lower level system variables, for example, to print the system variables in \$PARAM_GROUP, open the screen of the target level, and perform steps 1 and 2 above.

SYSTEM Variables		JOINT 10%
		3/236
47	\$ORIENTTOL	
48	\$OVRDSLCT	
49	\$PARAM_GROUP	
50	\$PASSWORD	
[TYPE]		

SYSTEM Variables		JOINT 10%
\$PARAM_GROUP[1]		3/236
1	\$BELT_ENABLE	FALSE
2	\$CART_ACCEL1	192
3	\$CART_ACCEL2	0
4	\$CIRC_RATE	1
5	\$CONTAXISNUM	0
6	\$EXP_ENBL	TRUE
[TYPE]		

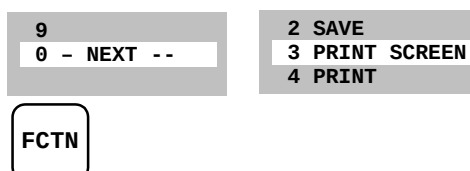
Procedure 8-13 Printing the displayed screen (print screen)

Condition

- The desired screen to be printed out is displayed.

Step

- 1 Press the FCTN key to display the function menu.
- 2 Press 0 –NEXT–, then select 3 PRINT SCREEN.



- 3 The displayed screen is printed out. "¥" is printed as the part of the reversed display on the teach pendant.
To stop printing, press the PREV key.

8.7 Subdirectories

Subdirectories can be used to organize programs or files on the file input/output device. Name of subdirectory can be specified at making. Subdirectory can be made into a subdirectory. Length of path name, for example MC:RC11\, should be less than 28 characters.

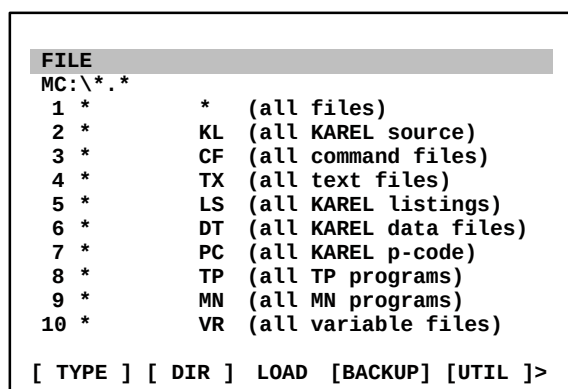
Procedure 8-14 Making Subdirectories

Condition

- The file input/output device is set correctly. (See Section 8.1.)

Step

- 1 Press the MENUS key to display the screen menu.
- 2 Select "7 FILE". The file screen appears as follows.



- 3 Press F5, [UTIL] and select Make DIR.
- 4 Input the subdirectory name followed from "Directory name:". Press ENTER to finish the input of subdirectory name.

```

1 Words
2 Upper Case
3 Lower Case
4 Options          -- Insert --
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)
Directory name: RC11
ABCDEF  GHIJKL  MNOPQR  STUVWX  YZ_@*

```

- 5 Subdirectory is created and current PATH is changed to this subdirectory automatically.

```

FILE
MC:\RC11\*. *
1 .. (Up one level) <DIR>
2 *      * (all files)
3 *      KL (all KAREL source)
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)

[ TYPE ] [ DIR ] LOAD [BACKUP] [UTIL]>

```

Procedure 8-15 Using Subdirectories

Condition

- The file input/output device is set correctly. (See Section 8.1.)

Step

- Press the MENUS key to display the screen menu.
- Select "7 FILE". The file screen appears.
- Press F2, [DIR], and select Directories or *.*. Directory entry is displayed with angle brackets.

```

FILE
MC: \*. *
1 RC11          <DIR>
2 RC12          <DIR>
3 *      * (all files)
4 *      KL (all KAREL source)
5 *      CF (all command files)
6 *      TX (all text files)
7 *      LS (all KAREL listings)
8 *      DT (all KAREL data files)
9 *      PC (all KAREL p-code)
10 *     TP (all TP programs)

[ TYPE ] [ DIR ] LOAD [BACKUP] [UTIL] >

```

- Move the cursor to a subdirectory entry with up or down arrow key then press ENTER to select this entry. Current PATH is changed to this subdirectory and the list of files included in this subdirectory is displayed automatically.

FILE			
MC: \RC11\			
1	..	(Up one level)	<DIR>
2	*	*	(all files)
3	*	KL	(all KAREL source)
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)
[TYPE] [DIR] LOAD [BACKUP] [UTIL]>			

- 5 Move the cursor to "(Up one level)" and press ENTER then PATH is moved to the one step upper subdirectory.

Procedure 8-16 Deleting Subdirectories

Condition

- The file input/output device is set correctly. (See Section 8.1.)

Step

- 1 Press the MENUS key to display the screen menu.
- 2 Select "7 FILE". The file screen appears.
- 3 Press F2, [DIR], and select Directories or *.*. Directory entry is displayed with angle brackets.
- 4 Move the cursor to the subdirectory that you want to delete. Press F1, [Delete] in the next page of Function key.
- 5 The message "Do you want to delete ?" is displayed. Then Press F4 [YES] , then selected subdirectory will be deleted. If any file exists in the selected subdirectory then the subdirectory can not be deleted. After all files are deleted in the subdirectory, try to delete the subdirectory again.

8.8 AUTOMATIC BACKUP

8.8.1 Overview of Automatic Backup

- Automatic Backup function performs the transaction of "all backup" in File menu automatically at the following timing.
 - The specified time (Up to 5 settings)
 - The specified DI is turned on.
 - Start up of the controller. (Interval can be specified.)
- The memory card (MC:) and the automatic backup area (FRA:) of F-ROM in the controller can be specified as a backup copy destination. The FRA: is specified by default.
- Automatic Backup function can manage many versions of backup in one device. Even if you backup the wrong programs or settings, you can load the previous version of backup. The number of versions to keep can be set from 1 up to 99. (Default is 2.)
- A storage device to be used for automatic backup need be previously initialized for automatic backup. Automatic backup will not be performed for any external storage device that has not been initialized for automatic backup.

Therefore, if an attempt is made to cause a backup copy to be automatically created on a memory card that has not be initialized for automatic backup, its content will not be lost.

The FRA: need not be initialized, since it is previously initialized.
- If the controller is turned off during automatic backup, or automatic backup is stopped immaturely, the latest backup copy is automatically restored into the system. No incomplete backup file is left in the storage unit, and the latest backup file can be read at any time.

NOTE

This function automatically saves all files. If the storage device used for automatic backup becomes faulty, the data saved in it may not be read. In case such an unforeseen accident takes place, it is necessary to save backups to another storage device such as a memory card as well.

8.8.2 Usable Memory Cards

NOTE

- 1 The required storage capacity is “(program size + 200 Kbytes) × (number of backup copies + 1).”
If the size of a program is 500 Kbytes, 13 backup copy versions of it can be made on a 10-Mbyte memory card.
- 2 If a memory card other than those recommended is used, a normal operation is not guaranteed, and a bad influence may occur on the controller.

8.8.3 Setting of Automatic Backup

MENU → “7 FILE” → F1([TYPE]) → “Auto Backup”. The following menu is displayed.

Auto Backup		1/15
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:	ENABLE	
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:	2	
15 Loadable version:	09/07/31 11:20	
[TYPE] INIT_DEV ENABLE DISABLE		

Automatic Backup works only when it is ENABLE.

Device and subdirectory to save.
Default of device is “Backup(FRA:)”
Default of subdirectory is blank.

Current status of the device displayed. *

Set the time to backup.
Up to 5 settings.
To clear setting, press F4 (CLEAR)

When the specified DI is turned on, backup is performed. (If index is 0, it is disabled.)

If it is ENABLE, backup is performed at start up.
The interval can be set.

The specified DO is turned on when backup is performed, or when error occurs at backup.
(8.8.4 Perform Automatic backup (Prior to V7.40) or
8.8.5 Perform Automatic backup (V7.40 or later))

Settings to manage versions of backup.
(8.8.6Version management)
(8.8.7Restore the backup)

Power-on time backup

If “Backup at Power up” is enabled, a backup copy is made when the power is turned on.

If the date of the latest backup copy in the storage device is within a period range (specified in “Interval”) from the current date, no backup copy is made at power-on time.

The period range is 7 days by default. If the default value is left unchanged, a backup copy is made at power-on time once every 7 days provided that “Backup at Power up” is enabled.

The unit of interval can be selected from “Day,” “Time,” and “Minute.”

If the “Interval” is reset to 0, a back-up copy is made every time the power is turned on.

Initializing of the storage device *

To use Memory Card for Automatic Backup, the Memory Card must be initialized for Automatic Backup. It is to protect to write to the other Memory Card. The status of device is displayed in "Status" line.

The FRA: need not be initialized, since it is previously initialized.

Ready for auto backup	Device is initialized for automatic backup
Device is not ready!	Device is not ready or device is not initialized for automatic backup

Device is initialized by the following operation.

- (1) If the device is not formatted, please format the device in file menu. (Refer to Procedure 8-2 Formatting file I/O device.)
- (2) Press F2 (INIT_DEV)
- (3) Message "Initialize the device for auto backup?" is displayed. Press F4 (YES).
- (4) Message "Enter number of versions to keep:" is displayed. Please enter the number (1 to 99) of versions to keep. Pressing only the enter key sets the number of backup copy versions to 2.

INIT_DEV deletes all files in the device, and create the special files and directories.

**CAUTION**

INIT_DEV does not format the device.

Please format the device in file menu (Procedure 8-2 Formatting file I/O device)

NOTE

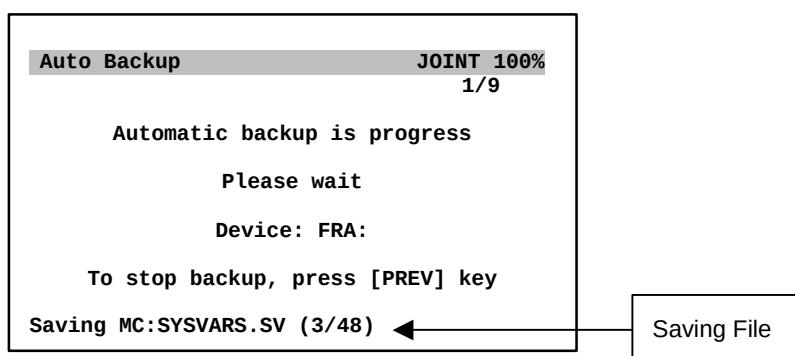
In case that the format type of the storage device is FAT16, up to 255 files can be saved to the route directory. Please pay attention to this point, when you select the storage device or the format type of the storage device. In case that you want to save more than 256 files, save the files to subdirectory or format the storage device by FAT32 format type.

8.8.4 Perform Automatic Backup (Prior to V7.40)

**CAUTION**

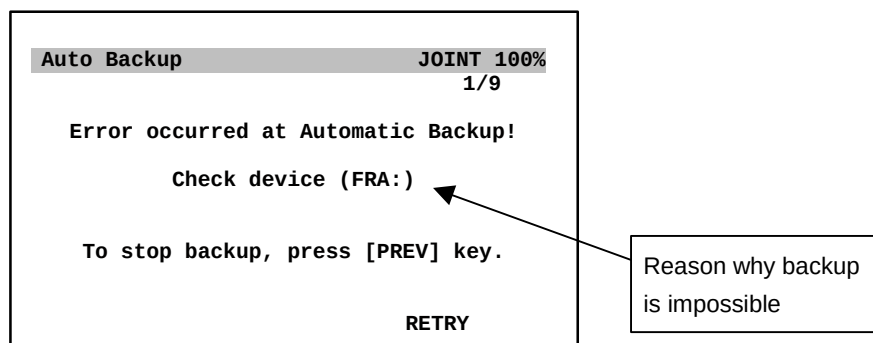
If you use the software V7.40 or later, refer the next section "8.7.5 Perform Automatic Backup"

When the specified condition is satisfied, automatic backup is performed.



- While automatic backup is performed, this menu is displayed. When automatic backup is completed, the previous menu is displayed.

- If you press PREV key, backup is cancelled and the previous menu is displayed. Any key except PREV is not accepted while automatic backup is performed.
- Even if you are using Teach Pendant, when automatic backup is performed, this menu is displayed and any key except PREV is not accepted. Please wait for Automatic backup is completed.
- If an attempt is made to perform automatic backup during program execution, it is performed while the program is running. It is also possible to start a program from the outside during backup.
- If the backup-in-progress signal is set, the specified signal becomes on while this menu is displayed.



- This menu appears if backup is impossible, for example, because no memory card has been inserted.
- In this case, the robot will not enter an alarm state. If a program is already running, it continues running. Also in this case, it is possible to start a program from the outside.
- By pressing F5(RETRY), backup is performed again.
- Pressing the PREV key resumes the previous menu.
- If a backup error signal is set, the specified signal becomes on while this menu is displayed.

**CAUTION**

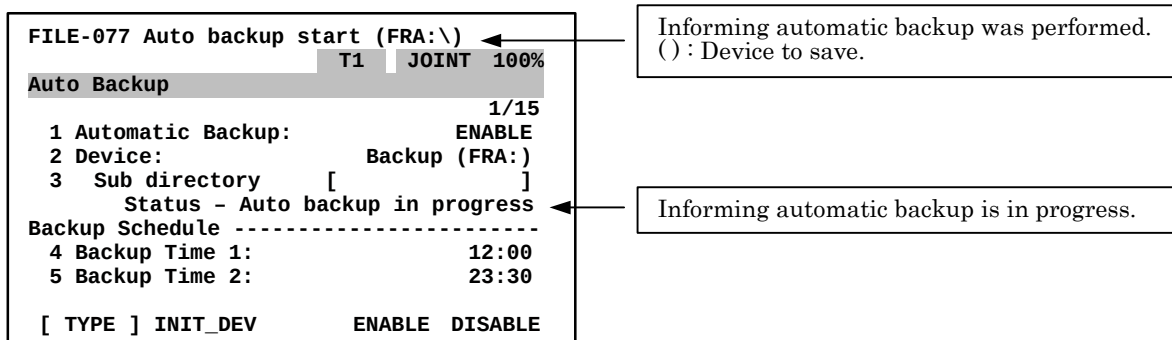
Please do not turn off the power of the controller while automatic backup function is performed. In case that the power of the controller is turned off while automatic backup function is performed, a normal operation is not guaranteed, and a bad influence may occur on the controller.

8.8.5 Perform Automatic Backup (V7.40 or later)

**CAUTION**

If you use the software prior to V7.40, refer the previous section “8.7.4 Perform Automatic Backup”

When the specified condition is satisfied, automatic backup is performed.



- When automatic backup is performed, the message “FILE-077 Auto backup start (FRA:\)” is displayed in the upper screen. And the device name to save is displayed in the case arc.
- While automatic backup is performed, “Auto backup in progress” is displayed on the “Status” line.
- When automatic backup is completed, the message “FILE-078 Auto backup complete” is displayed in the upper screen.
- When automatic backup is completed, “Ready for auto backup” is displayed on the “Status” line.
- If the backup-in-progress signal is set, the specified signal becomes on while this menu is displayed.

```

FILE-079 Error Auto backup (xxxx.xx) ←
FILE-055 MC not detected ←
Auto Backup JOINT 100%
1/15
1 Automatic Backup: ENABLE
2 Device: Backup (FRA:)
3 Sub directory [ ]
Status - Device is not ready!!
Backup Schedule -----
4 Backup Time 1: 12:00
5 Backup Time 2: 23:30
[ TYPE ] INIT_DEV ENABLE DISABLE

```

Informing automatic backup was performed. Showing the file name if automatic backup is failed to save a specific file.

Cause of the failure.

- The message “FILE-079 Error Auto backup (xxxx.xx)” appears in the upper screen if backup failed, for example, because no memory card has been inserted. And if backup failed at a specific file, the file name was shown in the case arc. Then, the cause of failure is displayed under the message.
- In this case, the robot will not enter an alarm state.
- If a backup error signal is set, the specified signal becomes on while automatic backup is in progress. The set signal reset when the next automatic backup is performed.



CAUTION

Please do not turn off the power of the controller while automatic backup function is performed. In case that the power of the controller is turned off while automatic backup function is performed, a normal operation is not guaranteed, and a bad influence may occur on the controller.

```

Alarm : Hist JOINT 100%
2/15
1 FILE-079 Error Auto backup
2 FILE-077 Auto backup start (MC:\)
3 SRVO-012 Power failure recovery
[ TYPE ] [ VIEW ] ACTIVE CLEAR DETAIL

```

User can check the history of messages about automatic backup.

- These messages displayed in the upper screen are overwritten by next warning or alarm messages. History of these messages about automatic backup can be checked in the alarm history screen.
[MENU] → ”4 ALARM” → (F3 “HIST”)

```

FILE-077 Auto backup start (FRA:\)
TEST001 LINE 0 T1 ABORTED
TEST001 JOINT 10%
4/4
1: J P[1] 100% FINE
2: L P[2] 2000mm/sec CNT100
3: L P[3] 2000mm/sec CNT100
[End]

```

Automatic backup is in progress.

User can continue present work.

- Unlike the normal backup process, automatic backup run in the background. So if automatic backup is performed, user continues the current work without regard to the automatic backup process.

- While automatic backup is in progress, it is impossible to perform normal backup (ref.8.4.2) to same device. If the operation is executed, normal backup is cancelled with the message “Backup in progress” in the bottom left of the screen. And automatic backup is continued.
However, differ in device, both of the two kinds of backup can be performed at the same time.
- Equally, while normal backup is in progress, it is impossible to perform automatic backup to same device. If the operation is executed, automatic backup is cancelled with the message “FILE-079 Error Auto backup (xxxx.xx)” and “FILE-081 Backup in progress” in the upper screen. And normal backup is continued.
However, differ in device, both of the two kinds of backup can be performed at the same time.

8.8.6 Version Management

Automatic Backup function can keep many backups in one device. The number of versions to keep is set at initializing the device. And you can change the number of versions to keep by the item “Maximum number of versions” anytime. The number of versions exceeds the specified number, the oldest version is deleted automatically.

If the device is FRA: If the size of a free storage area in F-ROM in the controller becomes smaller than 1 Mbyte, the oldest backup version is deleted automatically. In this case, the number of back versions actually held becomes smaller than “Maximum number of versions.” If the size of a free storage area in F-ROM is too small to hold an additional backup version, an error is detected during automatic backup execution.

If it is impossible to hold a specified number of backup versions on a memory card because of an insufficient storage capacity, an error is detected during automatic backup execution. Specify an appropriate number of backup versions by assuming the storage capacity required to hold one backup version is “program size + 200 Kbytes.”

If an error is detected because of an insufficient storage area during automatic backup, decrease the value specified in “Maximum number of versions.” This will causes an old backup version to be deleted, thus increasing a free area in the storage device.

Once a backup version is deleted by decreasing the value specified in “Maximum number of versions,” it cannot be restored by increasing the value.

Backup is 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 directory.

File menu can access the files only in root directory, so the latest version of backup can be loaded by file menu.

You can also load the older versions. (→ 8.8.7 Restore the backup)

When “all backup” is performed in file menu to the device that is initialized for Automatic Backup, the files are copied to the appropriate sub directory as same as automatic backup.

If the controller is turned off during backup, or backup is stopped prematurely, all backup files created during the current backup session are deleted, and the last backup version selected is restored to the root directory.

8.8.7 Restore the Backup

Backup files saved by Automatic Backup can be loaded by file menu. Pressing all of above on the file menu of the controlled start menu enables all files to be read simultaneously.

⚠ CAUTION

Before 7DA3, backup restore might be needed two times when backup is loaded after initial start. When max number of item is changed, this procedure is needed. (Example: max number of register or user alarm is changed by Program Limits screen.)

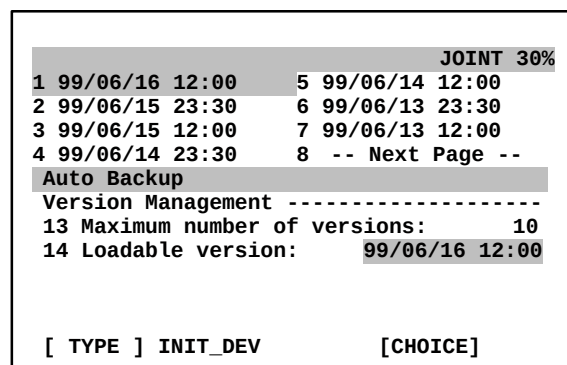
In this case, please load backup as following procedure.

1. Perform control start and load backup files by file menu.
2. Perform cold start.
3. Perform control start and load backup files again.

In 7DA4 or later, backup files don't need to load two times.

Usually the latest version of backup is in root directory and the version can be loaded by file menu. You can load the previous version by the following operation.

- (1) Press F4 (CHOICE) on the "Loadable version" item. The menu that contains the backup time of all versions in the device is displayed.



- (2) Please 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 root directory.
- (3) You can load the files of the selected version in file menu.
When controlled start is performed, pressing all of above on the file menu of the controlled start menu enables all backup files to be read simultaneously.

8.9 IMAGE BACKUP FUNCTION

Overview

By using the image backup function, it is possible to create images of the F-ROM and S-RAM memories of the controller. These images are saved as several files to a selected storage unit.

The image backup function can be used from the [FILE] menu when the system is in cold start mode. After [Image backup] is selected from the menu, actual backups are made the next time the controller is turned on.

To restore backups, press and hold down the [F1] and [F2] keys and turn on the controller. A menu appears from which previously saved images can be restored.

Storage units are a memory card (MC:) and a PC server (TFTP:) connected via Ethernet. To use a PC server as a storage unit, it is necessary to set up the Ethernet function of the controller correctly start up the TFTP server function on the PC server.

In case that both system software version and boot software version are 7DA5/12 or later, or 7DA7/04 or later, it is possible to backup or restore the image file to or from the subdirectory on memory card (MC:). Note that the procedure to backup or restore the image file is different from the procedure in other version.

Executing image backup

In case that the system software is series 7DA4 or prior to series 7DA4, or prior to version 12 in series 7DA5, or prior to version 04 in series 7DA7, please perform Procedure 8-17. In case that the system software series and version is later than above, please perform Procedure 8-18.

Procedure 8-17 Executing image backup (7DA4 or prior to 7DA4, Prior to 7DA5/12, Prior to 7DA7/04)



CAUTION

Image backup can be used only when the system is in cold start mode. It cannot be used during a control start.

Step

- 1 Press MENU key.
- 2 Select FILE item.
- 3 Press the F4 [BACKUP] key. The menu below appears.

```

1 System files
2 TP programs
3 Application
4 Applic.-TP
5 Error log
6 Diagnostic
7 All of above
8 Image backup
  
```

- 4 If the teach pendant is disabled when Image backup is selected, the prompt below appears.

```

Enable TP for image backup
  
```

Otherwise, the storage unit selection menu below appears.

```

1 Memory card (MC: )
2 Ethernet (TFTP:)
  
```

- 5 Image backup files are saved to the root directory of a selected unit. If the unit is MC: and it already contains *.IMG files, the confirmation message below appears.

```

Remove existing IMG files ?
      YES      NO
  
```



CAUTION

If the storage unit is TFTP:, *.IMG files are always overwritten.

- 6 Select YES, and a prompt appears, requesting that the power be turned off and back on.

```

Cycle power ?
      OK      Cancel
  
```

- 7 If OK is selected, the following message is displayed. Please turn off and on the power.

Please cycle power

- 8 As soon as the power is turned on, image backup is automatically started. A screen such as that shown below appears.

```

WRITING MC:\FROM00.IMG
WRITING MC:\FROM01.IMG
.....
WRITING MC:\SRAM02.IMG
DONE!

```

WARNING

- 1 During image backup, do not turn off the power.
- 2 If the storage unit is MC:, do not remove the memory card during image backup.

- 9 After image backup is completed or if an error occurs, the controller completes an ordinary start sequence. When the start is completed, either of the screens below appears.

Image backup completed successfully.
OK

Image backup failed.
OK

- 10 Press OK, and the file screen automatically appears.
- 11 If backup fails, alarm SYST-223 "Image backup failed(0x%x)" is posted to the alarm log. (0x%x) indicates the cause of the failure.

Procedure 8-18 Executing image backup (7DA5/12 or later, 7DA7/04 or later)

Step

- 1 Press MENUS key.
- 2 Select FILE item.
- 3 Press F5 [UTIL] and select the item "Set Device".
- 4 Select Mem Card (MC:) in the displayed menu.
- 5 Press F2 [DIR] and select the item "*. *".
- 6 Select the directory to save files in the list of directory so that the current directory becomes the directory to save files. It is possible to select the root directory as the directory to save files.

WARNING

Please use only alphanumeric character to enter the name of subdirectory to save image files. If non alphanumeric character is used, the name of subdirectory cannot be displayed in the list of directory to restore the image file.

- 7 Press F4 [BACKUP] key. The following menu is displayed.

```

1 System files
2 TP programs
3 Application
4 Applic.-TP
5 Error log
6 Diagnostic
7 All of above
8 Image backup

```

- 8 If the teach pendant is disabled when Image backup is selected, the prompt below appears.

```

Enable TP for image backup

```

Otherwise, the storage unit selection menu below appears.

```

1 Memory card (MC: )
2 Ethernet (TFTP:)

```

- 9 Image backup files are saved to the current directory of a selected storage device. If the unit is MC: and it already contains *.IMG files, the confirmation message below appears.

```

Remove existing IMG files ?
      YES      NO

```

**CAUTION**

If the storage unit is TFTP:, *.IMG files are always overwritten.

- 10 Select YES, and a prompt appears, requesting that the power be turned off and back on.

```

Cycle power ?
      OK      Cancel

```

- 11 If OK is selected, the following message is displayed. Please turn off and on the power.

```

Please cycle power

```

- 12 As soon as the power is turned on, image backup is automatically started. A screen such as that shown below appears.

```

WRITING MC:\FROM00.IMG
WRITING MC:\FROM01.IMG
.....
WRITING MC:\SRAM02.IMG
DONE!

```

**WARNING**

- 1 During image backup, do not turn off the power.
- 2 If the storage unit is MC:, do not remove the memory card during image backup.

- 13 After image backup is completed or if an error occurs, the controller completes an ordinary start sequence. When the start is completed, either of the screens below appears.

```
Image backup completed successfully.
OK
```

```
Image backup failed.
OK
```

- 14 Press OK, and the file screen automatically appears.
- 15 If backup fails, alarm SYST-223 "Image backup failed(0x%x)" is posted to the alarm log. (0x%x) indicates the cause of the failure.

Restoring image backups

In case that the system software is series 7DA4 or prior to series 7DA4, or prior to version 12 in series 7DA5, or prior to version 04 in series 7DA7, please perform Procedure 8-19. In case that the system software series and version is later than above, please perform Procedure 8-20.

Procedure 8-19 Restoring image backups (7DA4 or prior to 7DA4, Prior to 7DA5/12, Prior to 7DA7/04)

Step

- 1 Press and hold down both [F1] and [F2] keys and turn on the robot. The image restore screen appears.

```
***** RESTORE Controller Images *****
** Device selection menu ****
1. Memory card(MC:)
2. Ethernet(TFTP:)
Others. Cancel
Select :
```

- 2 If an item other than 1 and 2 is selected, the controller starts an ordinary start sequence. If item 1 or 2 is selected, the confirmation screen below appears.

```
***** 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] :
```

- 3 If 1 is entered, restoration starts. After completion, the controller starts an ordinary start sequence, using the restored images. If 2 is entered, image restoration is canceled.

Procedure 8-20 Restoring image backups (7DA5/12 or later, 7DA7/04 or later)**Step**

- 1 Press and hold down both [F1] and [F2] keys and turn on the robot. The image restore screen appears.

```

***** RESTORE Controller Images *****
** Device selection menu ****
1. Memory card(MC:)
2. Ethernet(TFTP:)
Others. Cancel
Select :

```

- 2 If an item other than 1 and 2 is selected, the controller starts an ordinary start sequence. If item 2 is selected, the procedure is same as Procedure 8-19. If item 1 is selected, the confirmation screen below appears.

```

*** BOOT MONITOR ***
Base version V7.70Pxx      [Release2]
*** Directory selection menu ***
Current Directory:
MC:\
1. OK(Current Directory)
2. Directory1
3. Directory2
4. Directory3
5. Directory4
6. Directory5
7. Directory6
8. Directory7
Select[0.NEXT.-1.PREV]:

```

- 3 Maximum 8 directories can be displayed at a time.(But, "OK (Current Directory)" is showed in No 1, "..(Up one level)" is showed in No2 if current directory is not root) Input 0."Next" and press Enter key, if you don't find destination directory. Next directories are displayed.

```

*** BOOT MONITOR ***
Base version V7.70Pxx      [Release2]
*** Directory selection menu ***
Current Directory:
MC:\
9. Directory8
10. Directory9
11. Directory10
12. Directory11

Select[0.NEXT, -1PREV]:

```

- 4 Select destination directory. Current directory is changed to selected sub directory and directory list including in the current directory are displayed as follows. (If the directory which name has non alphanumeric character can not be displayed in the list.)

```

*** BOOT MONITOR ***
Base version V7.70Pxx      [Release2]
*** Directory selection menu ***
Current Directory:
MC:\Directory10\
OK (Current Directory)
2.  ..(Up one level)
3.  Directory10_1
    Directory10_2
    Directory10_3

Select[0.NEXT, -1PREV]:

```

- 5 Select “1 OK (Current Directory)” if current directory is destination sub directory. Following display is displayed if number of image files corresponds with memory module size.

```

*** BOOT MONITOR ***
Base version V7.70Pxx      [Release2]
***** RESTORE Controller Images *****

Current memory module size
FROM: xxMb  SRAM: xMb

Image files are detected in
MC:\Directory10\
FROM: xxfiles  SRAM: xfiles
Correspond with module size.

Restore image files ? [Y=1/N=else]:

```

- 6 Select 1[Y] if you are sure for selected path. Image files are restored from selected path with following display.

```

Reading FROM00.IMG ... Done
Reading FROM01.IMG ... Done
Reading FROM02.IMG ... Done
Reading FROM03.IMG ... Done
Reading FROM04.IMG ... Done
Reading FROM05.IMG ... Done
Reading FROM06.IMG ... Done
Reading FROM07.IMG ... Done
Reading FROM08.IMG ... Done
Reading FROM09.IMG ... Done
Reading FROM10.IMG ... Done
Reading FROM11.IMG ... Done

```

8.10 ASCII UPLOAD FUNCTION

8.10.1 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. This is optional function.

- 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 sold on the open market.

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 8.9.4. 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.

8.10.2 Loading an ASCII Teach Pendant Program from the Teach Pendant

Use procedure 8-21 when you load an ASCII teach pendant program.

NOTE

The program being edited on the teach pendant cannot be loaded.

Procedure 8-21 Loading an ASCII Teach Pendant Program

Condition

- The memory card must be connected properly.
- The program you want is on the memory card.
- The program features are compatible with 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	(TP programs)
10	*	MN	(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:*.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	(TP programs)
10	*	MN	(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.

8.10.3 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 8.10.3 (a) and perform the operation in Table 8.10.3.(b)

In Procedure 8-22 you will be viewing Alarm History log.

Table 8.10.3 (a) Viewing ASCII Upload Error Screen Items.

ITEM	DESCRIPTION
ASBN Alarms	The facility code ASBN alarms are generated by the ASCII Upload option.

Table 8.10.3 (b) 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.
DETAIL	Press this key to access detailed information on a particular ASCII Upload error.

Procedure 8-22 Viewing ASCII Upload Errors**Conditions**

- A .LS file was loaded and it failed.

Steps

- 1 Press **MENUS** and select **ALARM**. 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
23- JAN-09 12:34
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.

8.10.4 Example ASCII File

This section details the syntax of an ASCII teach pendant program.

ASCII file 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

See an example of ASCII file in the following.

/PROG TEST1	-----	Program Data
/ATTR	-----	
OWNER = MNEDITOR;		File Attribute
COMMENT = "";		Data
PROG_SIZE = 598;		
CREATE = DATE xx-05-01 TIME 22:23:54;		
MODIFIED = DATE xx-05-01 TIME 22:24:02;		
FILE_NAME = ;		
VERSION = 0;		
LINE_COUNT = 2;		
MEMORY_SIZE = 958;		
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	-----	Application
CYCLE_REFERENCE = 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;		Data
CYCLE_TARGET = 0.00 ;		
/MN	-----	Instructions
1:J P[1] 100% FINE ;		
2:J P[2] 100% FINE ;		
/POS	-----	Start of
P[1]{		Position Data
GP1:		
UF : 0, UT : 1,		
X = 1584.56 mm,	Y = 299.91 mm,	Z = 300.3 mm,
W = -120.000 deg,	P = 0.000 deg,	R = -115.21 deg
};		
P[2]{		
GP1:		
UF : 0, UT : 1,		
X = 1584.56 mm,	Y = 299.91 mm,	Z = 300.3 mm,
W = -120.000 deg,	P = 0.000 deg,	R = -115.21 deg
};		
/END		

8.11 FILE MEMORY

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:"," RD:","MC:" and "UD1:". Use Procedure 8-23 to check file memory.

Procedure 8-23 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		JOINT 10%	
Total(KB): Free(KB): Type/Status			
Device	-----		
FR:	31857	18620	FAT16
MC:	499672	239048	FAT16
RD:	59	53	FAT16
UD1:	499672	331213	FAT16
[TYPE]		REFRESH	HELP

The data in this screen is uapdated to the latest data by pressing F4 REFRESH.

Table 8.11 FILE Memory Screen Items

ITEM	DESCRIPTION
Total (KB)	This item displays the amount of total memory available for each devices.
Free(KB)	This item displays the amount of free memory available for each devices.
Type/Status	This item displays the format type for each devices.

9

UTILITY

This chapter explains special functions of the robot controller.

Contents of this chapter

- 9.1 MACRO INSTRUCTION
- 9.2 SHIFT FUNCTIONS
- 9.3 COORDINATE SYSTEM CHANGE SHIFT FUNCTIONS
- 9.4 SOFT FLOAT FUNCTION
- 9.5 CONTINUOUS ROTATION FUNCTION
- 9.6 POSITION REGISTER LOOK-AHEAD EXECUTION FUNCTION
- 9.7 OPERATION GROUP DO OUTPUT FUNCTION
- 9.8 TIME BEFORE FUNCTION
- 9.9 DISTANCE BEFORE FUNCTION
- 9.10 STATE MONITORING FUNCTION
- 9.11 AUTOMATIC ERROR RECOVERY FUNCTION
- 9.12 REMOTE TCP FUNCTION
- 9.13 HIGH-SENSITIVITY COLLISION DETECTION
- 9.14 LOAD SETTING
- 9.15 LOAD ESTIMATION
- 9.16 COLLISION DETECTION FOR AUXILIARY AXIS
- 9.17 GRAVITY COMPENSATION
- 9.18 PASSWORD FUNCTION
- 9.19 MIXED LOGIC INSTRUCTION
- 9.20 PMC MONITOR FUNCTION
- 9.21 PMC EDIT FUNCTION
- 9.22 OPERATION LOG BOOK (OPTION)
- 9.23 ORIGINAL PATH RESUME
- 9.24 PROGRAM TOOLBOX
- 9.25 ADVANCED CONSTANT PATH
- 9.26 HIGH PERFORMANCE SOFT FLOAT FUNCTION
- 9.27 MULTI TASKING FUNCTION
- 9.28 ERROR SEVERITY TABLE
- 9.29 DIAGNOSTIC LOG
- 9.30 AUTOMATIC ERROR RECOVERY FUNCTION FOR MULTI-TASK
- 9.31 PATH SWITCHING FUNCTION
- 9.32 AUXILIARY AXIS SERVO OFF (LOCAL STOP) FUNCTION
- 9.33 STITCH FUNCTION
- 9.34 DUAL MOTOR DRIVE
- 9.35 MULTI ROBOT CONTROL FUNCTION
- 9.36 ROBOT ISOLATION FUNCTION
- 9.37 MULTI UOP INTERFACE FUNCTION
- 9.38 ERROR CODE OUTPUT FUNCTION
- 9.39 DATA MONITOR FUNCTION
- 9.40 BRAKE CHECK FUNCTION
- 9.41 PANEL WIZARD
- 9.42 ENHANCED MIRROR IMAGE
- 9.43 KAREL USE SUPPORT FUNCTION
- 9.44 KAREL PROGRAM EXECUTION HISTORY RECORD
- 9.45 TORQUE LIMIT FUNCTION
- 9.46 TCP SPEED OUTPUT INSTRUCTION
- 9.47 TP DRAM/FILE STORAGE FUNCTION

- 9.48 CYCLE TIME TRACKING
- 9.49 ROBOT TOOLS OF ROBOT HOMEPAGE
- 9.50 LINE REMARK INSTRUCTION
- 9.51 FINISHING FUNCTION PACKAGE
- 9.52 FOR/ENDFOR STATEMENT
- 9.53 MATH FUNCTION INSTRUCTION
- 9.54 GROUP MASK EXCHANGE

9.1 MACRO INSTRUCTION

A macro instruction is a function for registering a program consisting of a sequence of instructions as one instruction, and calling such a set of instructions for execution as required.

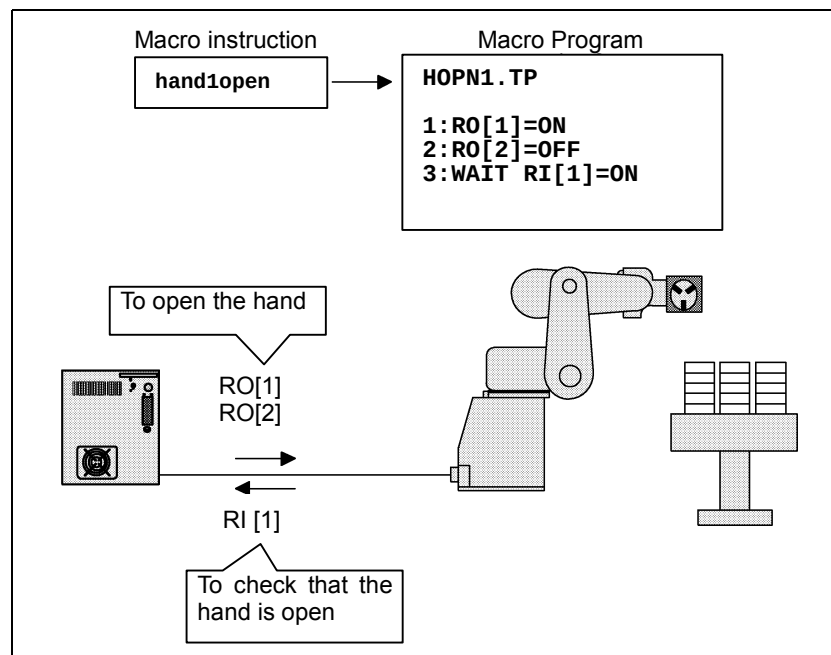


Fig. 9.1 Macro Instructions

A macro instruction has the following capabilities:

- A macro instruction, when taught in a program, can be started as a program instruction.
- A macro instruction can be started using the manual operation screen on the teach pendant.
- A macro instruction can be started using a user key on the teach pendant.
- A macro instruction can be started using the user button on the operator's panel. (The operation box can not be used because it does not have a user key.)
- You can start the macro command using DI, RI or UI.

Existing programs can be registered as macro instructions. Up to 20 macro instructions can be registered. A macro instruction can be used according to the following procedure:

- 1 Create a program to be executed as a macro instruction.
- 2 Register the created macro program as a macro instruction and determine from which device the macro instruction is to be called.
- 3 Execute the macro instruction.

The macro instruction setting screen [6 SETUP. Macro] is used for setting a macro instruction.

9.1.1 Setting Macro Instructions

The setting of a macro instruction involves the following items:

- Macro program

- Name of a macro instruction
- Assignment of a device used to start the macro instruction

Macro program

A macro program is a program started by a macro instruction. A macro program can be taught and played back (when played back as a program) in the same way as an ordinary program, except for the following restrictions:

- The subtype of a program, when registered as a macro program, is changed to MR (macro). When the registration of the macro program is canceled, the subtype returns to the original one. (For information about the subtype s, see Subsection 4.1.3.)
- A macro program registered as a macro instruction cannot be deleted.
- A program not including a motion (group) can be started even when the motion enabled state is not set (even when an alarm is issued). For group mask setting, the program information screen is used. (See Subsection 5.3.1.)
- The macro command not having the motion instruction should be made as the program which does not contain the motion group.

Name of a macro instruction

The name of a macro instruction is used to call the macro program from within a program. A macro instruction name must consist of an alphanumeric character string not longer than 16 characters. The macro command can be started while robot is moving.



CAUTION

Please do not use the parentheses “(” and “)” in the name of a macro instruction.
Example: handopen(hand1)

Assignment of a device

A macro instruction must be assigned to a key, screen item, etc. so it can be called. The item to which a macro instruction is assigned is called a device. The following devices are available:

- Items on the manual operation screen on the teach pendant (MF)
- User keys on the teach pendant (UK and SU)
- User buttons and other buttons on the operator's panel (not provided on the operator's box)
- DI, RI, UI

NOTE

If a macro instruction is allocated to a key switch on the teach pendant, the function previously allocated to the key becomes unavailable.



CAUTION

The operator should check that no macro instructions are allocated to user keys of the teach pendant.
If some instructions are allocated, a trouble would occur during execution.

Macro instructions can be assigned to the following devices:

- MF[1] to MF[99] : Items on the manual operation screen
- UK[1] to UK[7] : User keys 1 to 7 on the teach pendant
- SU[1] to SU[7] : User keys 1 to 7 + SHIFT key on the teach pendant
- SP[4] to SP[5] : User button 1 to 2 on the teach pendant
- DI[1] to DI[999] : DI 1 to 999
- RI[1] to RI[24] : RI 1 to 24
- UI[7] HOME signal

NOTE

- 1 The total number of the assign to the DI and RI is up to 5.
- 2 The allocation of macros to UI signals other than the HOME signal can be enabled with system variable \$MACRUOPENBL.
- 3 The number which can be actually used is only logical number allocated to the input signal line.

The macro instruction setting screen [6 SETUP. Macro] is used for setting a macro instruction.

**WARNING**

Before a program set as a macro instruction is copied from a controller onto another controller, the macro setting screens of the two controllers should be compared. It should be ensured that the lists of the two controllers match. The program should be copied only when the lists match. Otherwise, an unpredictable result would be produced.

Procedure 9-1 Setting macro instructions

Condition

- A macro program is created.

HOPN1	JOINT 30 %
	2/6
1: RO[1]=ON	
2: RO[2]=OFF	
3: WAIT RI[1]=ON	
[END]	
POINT	TOUCHUP>

- Macro program detail information is set.

NOTE

- 1 For greater convenience, a group mask can be set for a program not including motion instructions.
- 2 If the program to be modified contains a motion instruction, the group mask cannot be set.

Program detail	JOINT 30 %
	1/7
Program Name	
1 HOPN1	
2 Sub Type:	[None]
3 Comment:	[Open HAND1]
4 Group Mask:	[*,*,*,*,*,*,*]
5 Write protect:	[OFF]
6 Ignore pause:	[OFF]
7 Stack size:	[300]
END	PREV NEXT

Changing the motion group (setting a group mask)

Step

- 1 The program information screen is used to change the group mask.
- 2 Press the MENUS key to display the screen menu.
- 3 Select "1 SELECT" on the next page. The program list screen appears.

- 4 Press F2 “DETAIL” on the next page. The program information screen appears.
- 5 Move the cursor to group 1 of “Group Mask.” Press F5 “*” to set (*,*,*,*,*,*,*).

1 *

Program detail
JOINT 30 %

4/7

4 Group Mask: [*,*,*,*,*,*,*]

END PREV NEXT
1 *

F5

NOTE

If a motion instruction is already taught in a program to be modified, no group mask can be set.

Setting a macro instruction

Step

- 1 Press the MENUS key to display the screen menu.
- 2 Select “6 SETUP.”
- 3 Press F1 “TYPE” to display the screen change menu.
- 4 Select “Macro.” The macro instruction setting screen appears.

Macro Command		JOINT 30 %	
	Instruction name	Program	Assign
1	[hand]	[]	-- []
2	[hand]	[]	-- []
3	[hand]	[]	-- []
4	[hand]	[]	-- []
5	[hand]	[]	-- []

[TYPE]
CLEAR

- 5 For macro instruction input, press the ENTER key to display the character string input screen, then enter characters using an F key.

Macro Command
JOINT 30 %

Instruction name

1 [] []
2 [] []

ENTER

Macro Command
JOINT 30 %

Instruction name

1 [] []
2 [] []

ABCDEF
GHIJKL
NMOP

F2

JOINT 30 %

1 Words
2 Upper Case
3 Lower Case
4 Options ---Insert---

Macro Command
JOINT 30 %

Instruction name

1 [hand] [] []
2 [] [] []

abcdef ghijkl mnopqr stuvwx yz-@*.>

Upon completion of input, press the ENTER key.

Macro Command		JOINT 30 %	
Instruction name		Program	Assign
1	[handlopen]	[]	-- []
2	[]		

[TYPE] CLEAR [CHOICE]

ENTER

NOTE

No duplicate macro instruction definition is allowed.

- 6 For macro program input, press F4 [CHOICE] to display a list of programs, then choose a program from the list. When the macro program name is entered without the macro name, the program name will be used as the macro name.

[CHOICE]		JOINT 30 %	
1	PROGRAM1	5	SAMPLE1
2	PROGRAM2	6	SAMPLE2
3	HOPN1	7	
4	HCLS1	8	---next page---

F4

Macro Command		JOINT 30 %	
Instruction name		Program	Assign
1	[handlopen]	[HOPN1]	-- []

[TYPE] CLEAR [CHOICE]

[CHOICE]		JOINT 30 %	
1	PROGRAM1	5	
2	PROGRAM2	6	
3	HOPN1		
4	HCLS1		

ENTER

Macro Command		JOINT 30 %	
Instruction name		Program	Assign
1	[handlopen]	[HOPN1]	-- []

[TYPE] CLEAR [CHOICE]

- 7 For device assignment, press F4 "[CHOICE]" to display a list of device, then choose a device from the list.

[CHOICE]		JOINT 30 %	
1	--	5	SP
2	UK	6	DI
3	SU	7	RI
4	MF	8	---NEXT---

F4

Macro Command		JOINT 30 %	
Instruction name		Program	Assign
1	[handlopen]	[HOPN1]	-- []

[TYPE] CLEAR [CHOICE]

[CHOICE]		JOINT 30 %	
1	--	5	
2	UK	6	
3	SU		
4	MF		

ENTER

Macro Command		JOINT 30 %	
Instruction name		Program	Assign
1	[handlopen]	[HOPN1]	MF []

[TYPE] CLEAR [CHOICE]

- 8 Enter a desired device number.

JOINT 30 %			
name	Program	Assign	
1 [HOPN1	]	MF	[]

1 ENTER

Macro Command JOINT 30 %			
Instruction name	Program	Assign	
1 [handlopen	]	MF	[1]

[TYPE] CLEAR

**CAUTION**

After all macro instructions are set, the setting information should be saved in external storage in case the information needs to be re-loaded. Otherwise, the current setting information would be lost when it is changed.

- 9 For macro instruction deletion, move the cursor to a desired field, then press F2“CLEAR” while holding down the SHIFT key.

Macro Command			
Instruction name	Program	Assign	
1 []	[HOPN1	]	MF [1]
2 [hand1close	]	MF	[2]
3 []	[]	--	[]
4 []	[]	--	[]

[TYPE] CLEAR

F2

Macro Command JOINT 30 %			
Instruction name	Program	Assign	
1 []	[HOPN1	]	MF [1]
2 [hand1close	]	MF	[2]
3 []	[]	--	[]
4 []	[]	--	[]

[TYPE] CLEAR

- 10 “Clear OK?” appears.
- To delete the macro instruction, press F4 “YES.”
 - To cancel deletion of the macro instruction, press F5 “NO.”

YES	NO
F4	

Clear OK?	
YES	NO

9.1.2 Executing Macro Instructions

A macro instruction can be executed by:

- Selecting an item on the manual operation screen on the teach pendant (with the SHIFT key held down)
- Pressing user keys on the teach pendant (without pressing the SHIFT key)
- Pressing user keys on the teach pendant (with the SHIFT key held down)
- DI, RI, UI
- Calling the macro instruction from the program

When a macro instruction is started, the macro program is executed in the same way as an ordinary program is executed, except for the following restrictions:

- The single step mode is disabled. The continuous operation mode is always used.
- The macro program is always aborted without the pausing status.
- The macro program is always executed starting from the first line.

When a macro program includes a motion instruction (uses a motion group), the motion enabled state must be set to execute the macro instruction. When no motion group is used, the motion enabled state need not be set.

The motion enabled state is set when:

- ENBL is on.

- SYSRDY output is on. (Servo power supply is on)

Table 9.1.2 Macro Instruction Execution Conditions

		Without a motion group	With a motion group
MF [1 to 99] SU [1 to 7]	TP enabled	Executable(*1)	Executable
UK [1 to 7]		Executable	-
SP [4 to 5] DI [1 to 999] RI [1 to 5] UI [1 to 8]	TP disenabled	Executable	Executable

NOTE (*1)

Even when the teach pendant is disabled, a macro instruction that does not possess a motion group can be executed from an MF or SU by setting system variable \$MACRTPDSBEXE = TRUE.

- *) It is possible to supply an argument in a macro instruction call in a program and use it in a macro program. For details, see Subsection 4.7.5, "Arguments."

Procedure 9-2 Executing a macro instruction using the teach pendant (manual operation screen)**Condition**

- The teach pendant is enabled.

NOTE

Even when the teach pendant is disabled, a macro instruction that does not possess a motion group can be executed from an MF or SU by setting system variable \$MACRTPDSBEXE = TRUE.

- A device from MF[1] to MF[99] is set using the macro instruction setting screen.

Macro Command		JOINT 30 %	
Instruction name	Program	Assign	
1 [hand1open]	[HOPN1]	MF	[1]
2 [hand1close]	[HCLS1]	MF	[2]
3 []	[]	--	[]
4 []	[]	--	[]
[TYPE]			

Step

- 1 Press the MENUS key to display the screen menu.
- 2 Select "3 MANUAL FCTNS."
- 3 Press F1, [TYPE] to display the screen change menu.
- 4 Select "Macros." The manual operation screen appears.

MANUAL MACROS		JOINT 30 %	
Instruction		1/3	
1 [hand1open]			
2 [hand1close]			
[TYPE]		EXEC	

**WARNING**

The macro program is started in the next step, causing the robot to make a motion. Before executing the operation, the operator should check that no persons and no unnecessary equipment are in the work area. Otherwise, injury or property damage could occur.

- 5 To start a desired macro instruction, press F3 “EXEC” while holding down the SHIFT key. The macro program is started.



Hold down the SHIFT key until the execution of the macro program is completed.

NOTE

When the macro program contains a motion group, hold down the shift key until execution of the macro program terminates. If the shift key is released while the macro is being executed, the macro program is stopped. When the macro program does not contain a motion group, program execution continues even if the shift key is released.

**CAUTION**

If the SHIFT key is released during execution, the macro program is terminated forcibly. Note that when execution is interrupted and F3 “EXEC” is pressed again, the macro program is executed from the first line again.

Procedure 9-3 Executing a macro instruction using the teach pendant (using a user key)

Condition

- The teach pendant is enabled.

NOTE

Even when the teach pendant is disabled, a macro instruction that does not possess a motion group can be executed from an MF or SU by setting system variable \$MACRTPDSBEXE = TRUE.

- A device from UK[1] to UK[7] or SU[1] to SU[7] is set on the macro instruction setting screen.

Macro Command		JOINT 30 %	
Instruction name	Program	Assign	
1 [hand1open]	[HOPN1]	] SU	[1]
2 [hand1close]	] [HCLS1	] SU	[2]
3 [	] [	] --	[]
4 [	] [	] --	[]
[TYPE] CLEAR			

Step

- 1 To start a macro instruction on the teach pendant, use the assigned user key on the teach pendant.

**WARNING**

The macro program is started in the next step, causing the robot to make a motion. Before executing the operation, the operator should check that no persons and no unnecessary equipment is in the work area. Otherwise, injury or property damage would occur.

- 2 When a user key from UK[1] to UK[7] is assigned to the macro instruction, press the assigned user key to start the macro instruction.

NOTE

A macro instruction that possesses a motion group cannot be executed using a device from UK[1] to UK[7]. A device from SU[1] to SU[7] must be assigned to such a macro instruction.

- 3 When a device from SU[1] to SU[7] is assigned to the macro instruction, press the user key while holding down the SHIFT key.

NOTE

When the macro program contains a motion group, hold down the shift key until execution of the macro program terminates. If the shift key is released while the macro is being executed, the macro program is stopped. When the macro program does not contain a motion group, program execution continues even if the shift key is released.

**CAUTION**

If the SHIFT key is released during execution, the macro program is terminated forcibly. Note that when execution is interrupted and F3 "EXEC" is pressed again, the macro program is executed from the first line again.

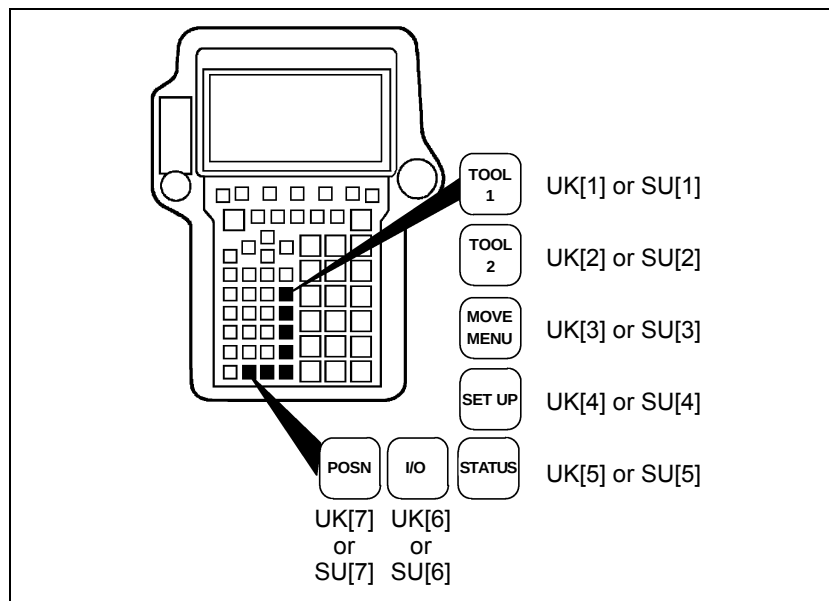


Fig. 9.1.2 User Keys on Teach Pendant

**CAUTION**

When a key on the teach pendant is assigned to a macro instruction, it becomes that macro instructions device, and the key can no longer be used for its original function.

Procedure 9-4 Execution of macro command using DI,RI and UI

Condition

- The teach pendant must be disabled.
- DI[1] to DI[99], RI[1] to RI[24] or UI[7] is specified as the device in the macro instruction setting screen.

Macro Command		JOINT 30 %	
Instruction name	Program	Assign	
1 [RETURN TO REFPOS]	[REFPOS]	UI	[7]
2 [WORK1 CLAMP]	[CLAMP1]	DI	[2]
3 [PROCESSING PREP]	[PREP]	RI	[3]
4 [	[	--	[]
[TYPE] CLEAR			

Step

- 1 To start the macro command using DI, RI or UI, input the digital signal from the external device or directly input these signals in the I/O screen on the teach pendant.
- 2 When DI or RI or UI which is set in the macro instruction setting screen is input, the macro command which is assigned to the signal will be started.

NOTE

Moreover, \$MACROUOPENBL can be changed in the system variable screen displayed at controlled start.

9.2 SHIFT FUNCTIONS

The shift functions shift the specified positions for the operation instructions within a certain range of a previously taught program to other locations.

The shift functions perform the following:

- Shift the position data for the operation instructions within the entire range or within a certain range of an existing program.
- Insert the shift results into a new or existing program.
- Repeat the same shift on another program.

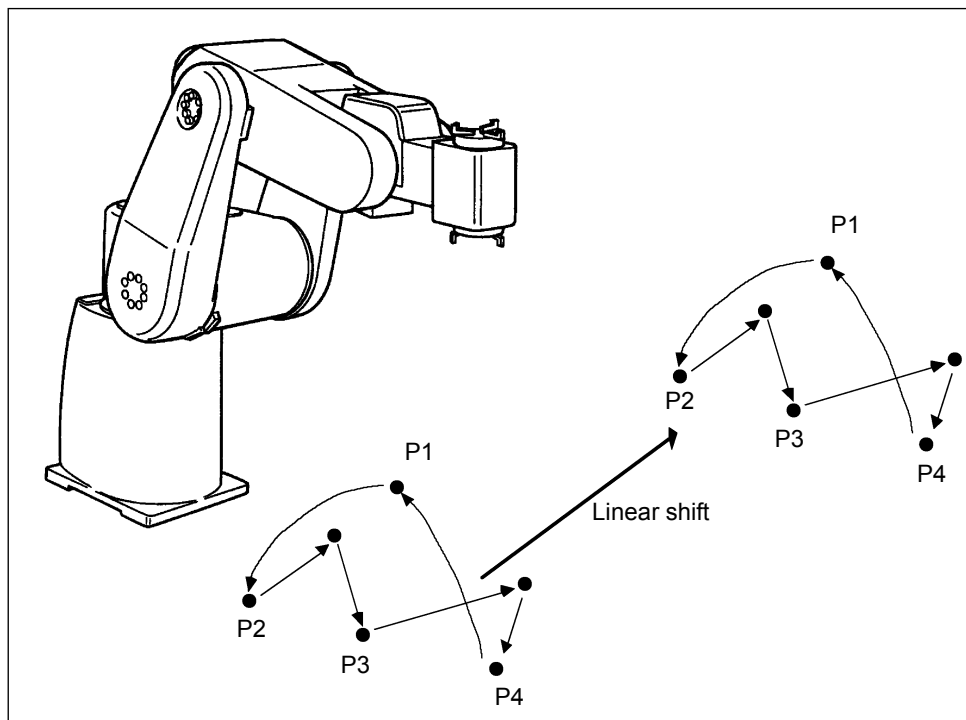


Fig. 9.2 Shift

The following rules apply to converted position data:

Rules governing position data:

- Position data having Cartesian coordinates is converted to Cartesian coordinates. Position data with joint coordinates is converted to joint coordinates.
- If converted joint coordinate position data falls outside the joint operating area, it is stored as unspecified. Converted Cartesian coordinate position data is stored as is even if it falls outside the joint operating area.
- Position data in the position registers is not converted.
- The position data with joint coordinates for operation instructions involving incremental instructions is stored as unspecified.

Rules governing the Cartesian coordinate system number (UT, UF) in position data having Cartesian coordinates:

- The Cartesian coordinate system number is not changed due to conversion.
- During conversion (on the shift information input screen), a user coordinate system number (UF) of 0 is used. Position data is converted to data in the Cartesian coordinate system with a UF of 0 (world coordinate system) and displayed.

Rules governing the configuration (joint placement and turn number) of position data having Cartesian coordinates:

- The configuration is not changed as a result of the conversion.
- For the turn number, if the conversion causes rotation about the wrist axis by 180° or greater, the turn number for the axis is optimized, and a message appears so that the user can determine whether to accept it.

The following shift functions are available:

- Program shift : Performs a 3-dimensional linear shift or linear rotation shift.
- Mirror shift : Performs a 3-dimensional symmetrical shift about a specified mirror plane.
- Angle-input shift : Performs a rotation shift about a specified rotation axis.

9.2.1 Program Shift Function

The program shift function performs a linear shift or linear rotation shift on the specified positions for the operation instructions within a certain range of a previously taught program.

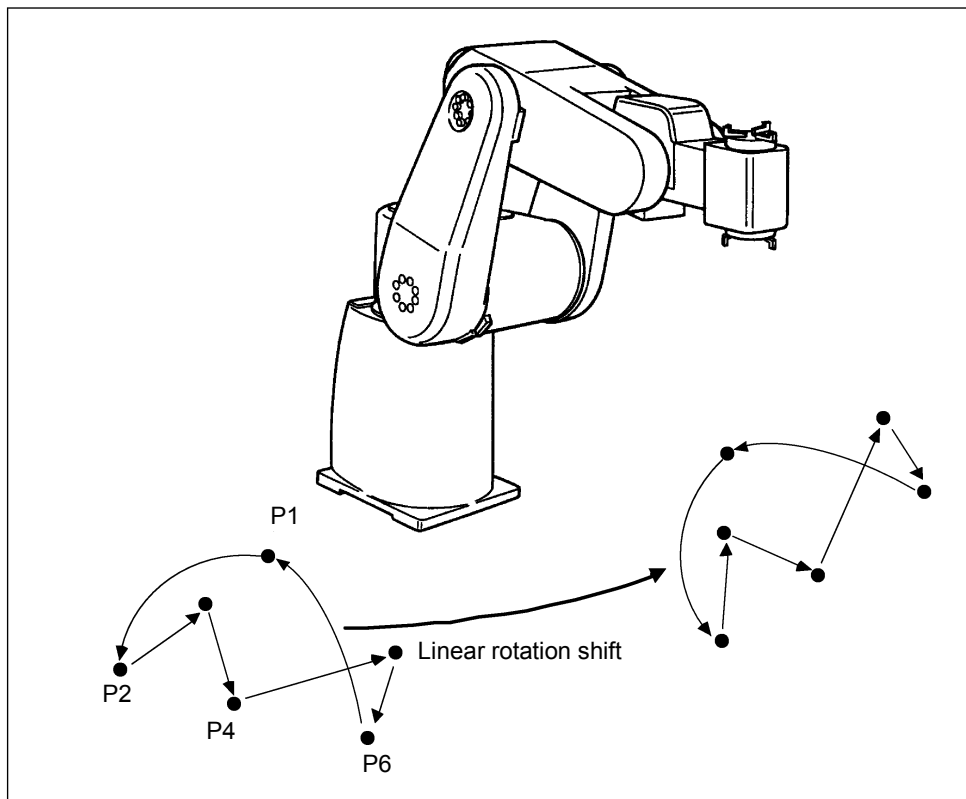


Fig. 9.2.1 (a) Linear Rotation Shift

The program shift function requires the following setup:

Program name setting

Program name setting specifies the name of the source program, the range of lines on which the shift is to be performed, as well as the name of the program into which the shift results are to be inserted and the line at which they are to be inserted.

Shift information input

Shift information input specifies the direction and amount of the program shift function. Two types of shift are supported: linear shift and linear rotation shift. The shift direction and amount can be specified in either of two ways: representative point specification and direct specification.

- In representative point specification, the user indicates (specifies) representative source and destination points to determine the shift direction and amount.
For a linear shift, one source point (P1) and one destination point (Q1) must be indicated (specified).

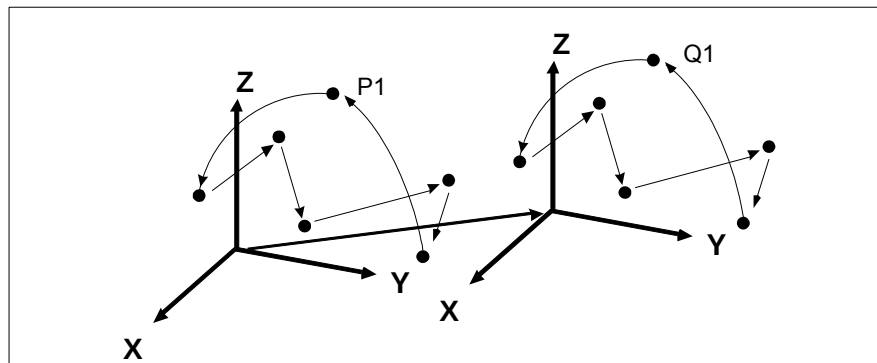


Fig. 9.2.1 (b) Specifying a Linear Shift

For a linear rotation shift, three source points (P1, P2, and P3) and three destination points (Q1, Q2, and Q3) must be indicated (specified).

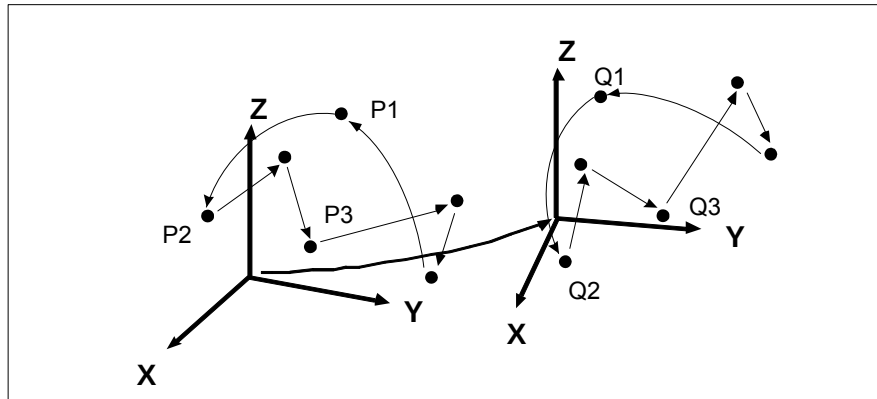


Fig. 9.2.1 (c) Specifying a Linear Rotation Shift

- In direct specification, the user directly specifies the direction and amount (X, Y, Z) of linear shift. In direct specification, linear rotation shift cannot be specified.

To execute the program shift function, use the program shift screen PROGRAM SHIFT. The figure below shows how to navigate through the program shift screen.

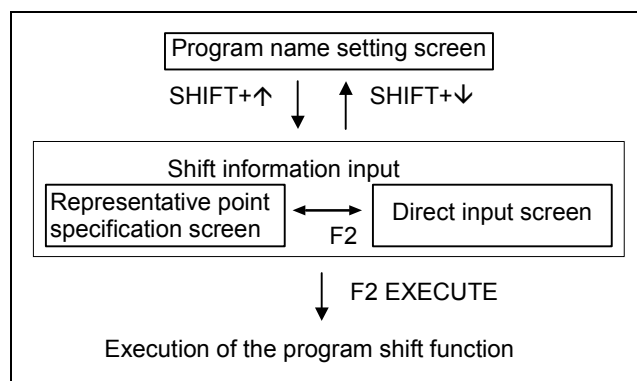


Fig. 9.2.1 (d) Program Shift Screen

The program name input screen contains the following items:

Table 9.2.1 (a) Contents of the Program Name Input Screen

Item	Description
Original Program	Specifies the name of the source program.
RANGE	Specifies the type of the desired range of the source program. • WHOLE = Performs shift on the entire program. • PART = Performs shift on part of the program.
Start line	Specifies the start line of the desired range of the source program. If WHOLE is set to all, this item cannot be specified.
End line	Specifies the end line of the desired range of the source program. If WHOLE is set to all, this item cannot be specified.
New Program	Specifies the program into which the shift results are to be inserted. If a new program name is specified, a new program is created with that name. If the name of an existing program is specified, the results are inserted into that program.
Insert line	Specifies the line at which the shift results are to be inserted, if insertion of the results are to be into an existing program is specified. If the program is a new one, this item cannot be specified.

The representative point specification screen contains the following items:

Table 9.2.1 (b) Contents of the Representative Point Specification Screen

Item	Description
Position data	Indicates the position of the point where the cursor is currently located. The position is always represented by coordinates in the world coordinate system.
Rotation	Specifies whether rotation is to be performed.
Source position	Specifies the position of a representative source point.
Destination position	Specifies the position of a representative destination point.
REFER	F4 REFER allows the use of a position variable or position register in the source program as the position of a representative point.

Procedure 9-5 Executing the program shift function

Condition

- The program on which the shift is to be performed exists.

TEST2	JOINT 30 %
	1/6
1:J P[1] 100% FINE	
2:J P[2] 70% CNT50	
3:L P[3] 100cm/mic CNT30	
4:L P[4] 500m/sec FINE	
5:J P[1] 100% FINE	
[END]	
POINT	TOUCHUP >

Step

- 1 Press the MENUS key. The screen menu appears.
- 2 Select 1, UTILITIES.
- 3 Press F1, [TYPE]. The screen switching menu appears.
- 4 Select Program shift. The program name input screen appears.

1 UTILITIES
2 TEST CYCLE

MENUS

Program shift
TYPE

F1

PROGRAM SHIFT	JOINT 10 %
Program	1/6
1 Original program:	[Test1]
2 Range:	WHOLE
3 Start line: (not used)	***
4 End line: (not used)	***
5 New program:	[Test1]
6 Insert line: (not used)	***
Use shifted up,down arrows for next page	
[TYPE]	>
CLEAR	>

- 5 Specify the necessary items.

- 6 After specifying the items, go to the next screen with SHIFT + ↓. The representative point specification screen appears. To return to the previous screen, use SHIFT + ↑.

SHIFT



PROGRAM SHIFT		JOINT 30 %	
Shift amount/Teach			
Position data P1			
X	*****	Y	*****
		Z	*****
1 Rotation			OFF
2 Source position			P1
3 Destination position			Q1
[TYPE]	EXECUTE	ON	OFF >

- 7 For a shift with rotation, set "Rotation" to ON.

PROGRAM SHIFT		JOINT 10 %	
Shift amount/Teach			
Position data			
X	*****	Y	*****
		Z	*****
1 Rotation			ON
2 Source position			P1
			P2
			P3
3 Destination position			Q1
			Q2
			Q3
[TYPE]	EXECUTE	ON	OFF >

- 8 Specify representative source and destination points.

REFER RECORD>

SHIFT F5

PROGRAM SHIFT		JOINT 10 %	
Shift amount/Teach			
Position data Q1			
X	1234.4	Y	100.0
		Z	120.0
1 Rotation			ON
2 Source position		P1	Recorded
		P2	Recorded
		P3	Recorded
3 Destination position		Q1	Recorded
		Q2	Recorded
		Q3	Recorded
[TYPE]	EXECUTE	REFER	RECORD>

- 9 For reference point input, press F4 REFER. Select F4 P[] or F5 PR[] to enter arguments.

REFER	RECORD>	3 Destination position				Q1	Recorded
					Q2	P[5]	
					Q3		
F4		[TYPE]	EXECUTE		P[]	PR[] >	
	P[] PR[] >						
F4							

- 10 After setting shift information, press F2, EXECUTE and then F4, YES. The conversion results are written into the program.

[TYPE]	EXECUTE	TEST2		JOINT 30 %
				1/6
F2		1:J P[1] 100% FINE		
		2:J P[2] 70% CNT50		
		3:L P[3] 100cm/mic CNT30		
		4:L P[4] 500m/sec FINE		
		5:J P[1] 100% FINE		
		[END]		
Execute transform?		POINT		
YES NO		TOUCHUP >		
F4				

- 11 The direct input screen appears with F2, DIRECT on the next page. Specify the shift amount directly.

PROGRAM SHIFT		JOINT 10 %
Shift amount/Direct entry		
1 X (mm)		1888.92
2 Y (mm)		239.87
3 Z (mm)		50.52
[TYPE]	EXECUTE	>
CLEAR	TEACH	>

NOTE

Set the shift amount using coordinates in the world coordinate system.

- 12 After setting the shift amount, press F2, EXECUTE to execute the shift.
 13 If the turn number is changed due to the shift, the user is notified and asked which to select.

Select P[3]:J5 angle (183°)	
183° -177° uninit	QUIT >

- 14 F1 indicates the axial angle associated with the changed turn number.
 F2 indicates the axial angle associated with the original turn number.
 F3 uninit causes the data to become unspecified data.
 F5 QUIT interrupts the conversion.

- 15 To erase all the shift information, press F1, CLEAR on the next page. Then, the currently selected program is specified as the source program.

CLEAR TEACH

F1

9.2.2 Mirror Shift Function

The mirror shift function shifts the specified positions for the operation instructions in a certain range of an already taught program symmetrically about a plane.

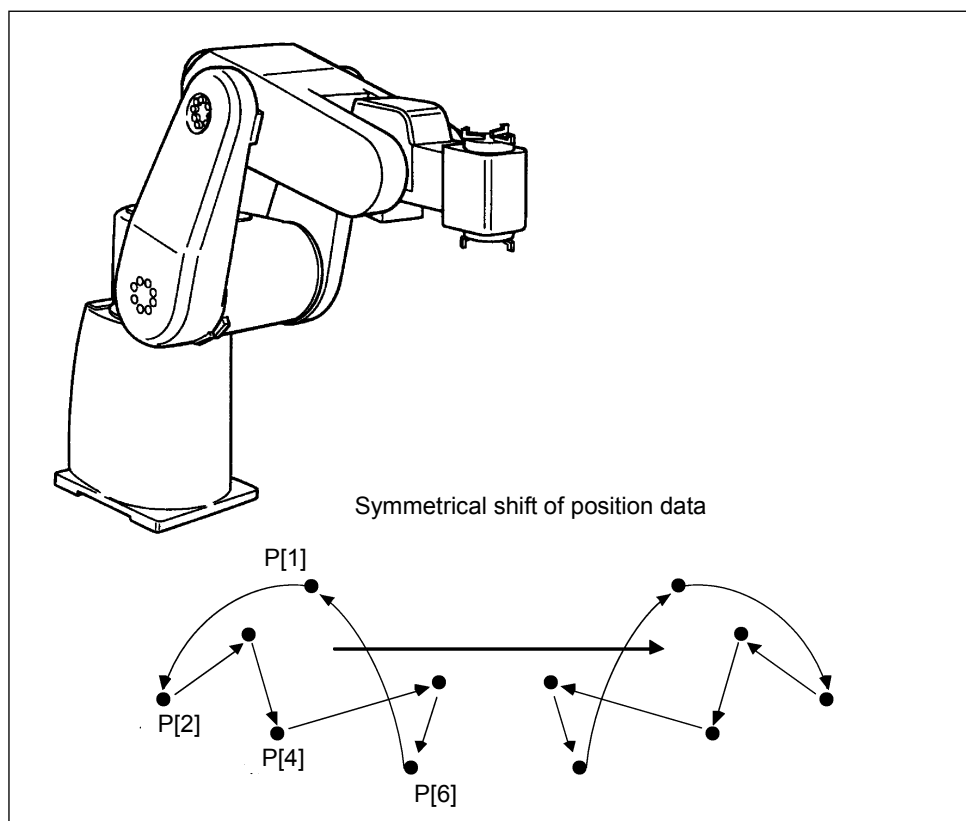


Fig. 9.2.2 (a) Mirror Shift Function

Theoretically, the mirror shift function converts the attitude of the tool from right-handed coordinates to left-handed coordinates. In reality, however, the attitude is returned to the right-handed coordinate system by inverting the Y-axis because no left-handed coordinates exist.

The mirror shift function, therefore, performs conversion most naturally when the plane of symmetry is parallel to the XZ plane of the tool coordinate system.

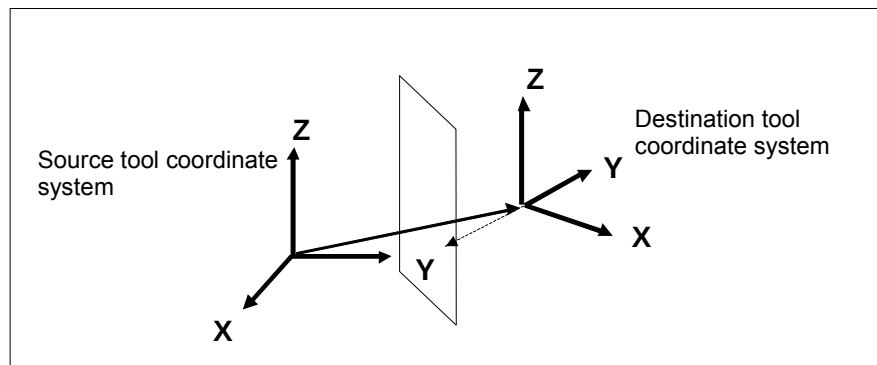


Fig. 9.2.2 (b) Conversion from One Tool Coordinate System to Another with the Mirror Shift Function

⚠ CAUTION

- 1 The tool coordinate system must be established accurately. The mirror shift function requires that the Z-axis match the tool direction.
- 2 The tool center point (TCP) must be set accurately to ensure correct operation with the points resulting from a symmetrical shift. Otherwise, the points resulting from the shift will contain offset values.

The mirror shift function requires the following setup:

Program name setting

Program name setting specifies the name of the source program, the range of lines on which the shift is to be performed, as well as the name of the program into which the shift results are to be inserted and the line at which they are to be inserted.

Shift information input

Shift information input specifies the direction and amount of the mirror shift. Two types of shift are supported: symmetrical shift and symmetrical rotation shift.

- In representative point specification, the user indicates (specifies) representative source and destination points to determine the shift direction and amount.
For a symmetrical shift, one source point (P1) and one destination points (Q1), two points in total, must be indicated (specified).

For a symmetrical rotation shift, three source points (P1, P2, and P3) and three destination points (Q1, Q2, and Q3), six points in total, must be indicated (specified).

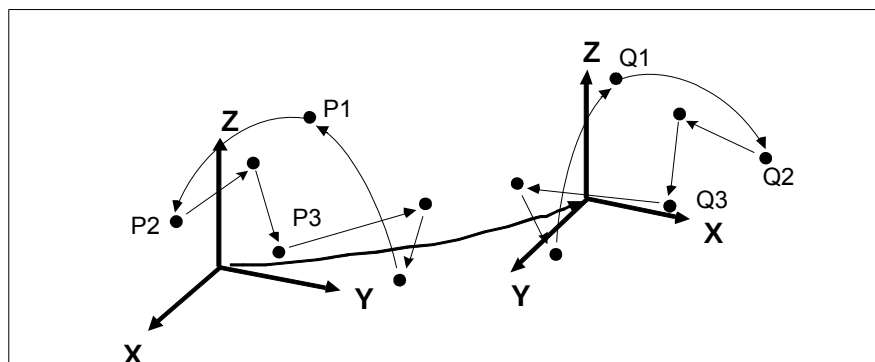


Fig. 9.2.2 (c) Specifying the Mirror Shift Function

To execute the mirror shift function, use the mirror screen MIRROR IMAGE SHIFT. The explanation of the program shift screen also applies to the mirror screen.

Procedure 9-6 Executing the mirror shift function

Condition

- The program on which the shift is to be performed exists.

TEST2			JOINT 30 %
			1/6
1:J	P[1]	100% FINE	
2:J	P[2]	70% CNT50	
3:L	P[3]	1000cm/min CNT30	
4:L	P[4]	500m/sec FINE	
5:J	P[1]	100% FINE	
[END]			
POINT			TOUCHUP >

Step

- 1 Press the MENUS key. The screen menu appears.
- 2 Select 1, UTILITIES.
- 3 Press F1, [TYPE]. The screen switching menu appears.
- 4 Select Mirror Image. The program name input screen appears.

1 UTILITIES
2 TEST CYCLE

MENUS

Mirror Image
TYPE

F1

MIRROR IMAGE SHIFT	JOINT 10 %
Program	1/6
1 Original program:	[Test1]
2 Range:	WHOLE
3 Start line: (not used)	***
4 End line: (not used)	***
5 New program:	[Test1]
6 Insert line: (not used)	***

Use shifted up,down arrows for next page

[TYPE] >

CLEAR >

NOTE

The program selected last with the list screen is automatically selected as the source program.

- 5 Specify the necessary items.
- 6 After specifying the items, go to the next screen with SHIFT + ↓. The representative point specification screen appears. To return to the previous screen, use SHIFT + ↑.

SHIFT

MIRROR IMAGE SHIFT	JOINT 10 %
Shift amount/Teach	
Position data	
X ***** Y ***** Z *****	
1 Rotation	OFF
2 Source position	P1
3 Destination position	Q1
[TYPE] EXECUTE	ON OFF >
CLEAR DIRECT	>

- 7 For shift with rotation, set "Rotation" to Yes.

MIRROR IMAGE SHIFT			JOINT 10 %
Shift amount/Teach			
Position data			
X	*****	Y	*****
Z	*****		
1 Rotation			ON
2 Source position			P1
			P2
			P3
3 Destination position			Q1
			Q2
			Q3
[TYPE]	EXECUTE	ON	OFF >

- 8 Specify representative source and destination points.
For details, see the explanation of the program shift function.
- 9 After setting the shift amount, press F2 EXECUTE to execute the shift.

[TYPE]	EXECUTE
F2	
Execute transform?	
YES	NO
F4	

**WARNING**

Avoid moving the robot to a position that is not correctly shifted. Check the shift results before moving the robot. Otherwise, serious problems can occur.

- 10 To erase all shift information, press F1 CLEAR on the next page.

CLEAR	TEACH
F1	

9.2.3 Angle-input Shift Function

The angle-input shift function allows the user to perform a program shift by directly entering three or four representative points and an angular displacement. It also allows the user to perform multiple shifts at equal intervals on the same circumference at one time by specifying the iteration.

If many locations on the same circumference are subject to the same machining, such as the holes on a car wheel, this function allows the user to create position data for all the locations to be machined by specifying only a single location.

The angle-input shift function requires the following setup:

Program name setting

Program name setting specifies the name of the source program, the range of lines on which the shift is to be performed, as well as the name of the program into which the shift results are to be inserted and the line at which they are to be inserted.

Shift information input

Shift information input specifies the representative points for determining the rotation axis for the angle-input shift function and sets the angular displacement and shift iteration. The representative points can be specified in either of two ways: one in which the rotation axis is specified and one in which it is not specified.

- If the rotation axis is not specified, three representative points (P1, P2, and P3) on the same circumference must be specified. With these three points, the rotation plane and axis are automatically calculated. The intersection of the rotation plane and axis (rotation center) is set as representative point P0.

Rotation center P0, which is set automatically, can be changed directly later. From the second conversion on, the position of the rotation center can be compensated for by enabling the rotation axis.

- If the rotation axis is specified, a point on the rotation axis must be specified for representative point P0 and any three points on the rotation plane must be specified for representative points P1, P2, and P3. (P1, P2, and P3 need not be on the same circumference.) The rotation plane is determined with representative points P1, P2, and P3. The axis that is vertical to the rotation plane and which passes through representative point P0 is determined as the rotation axis.

In either way, the more distant the representative points P1, P2, and P3, the more precise the conversion. The direction of rotation is regarded as being positive when the rotation is from representative point P1 to P2.

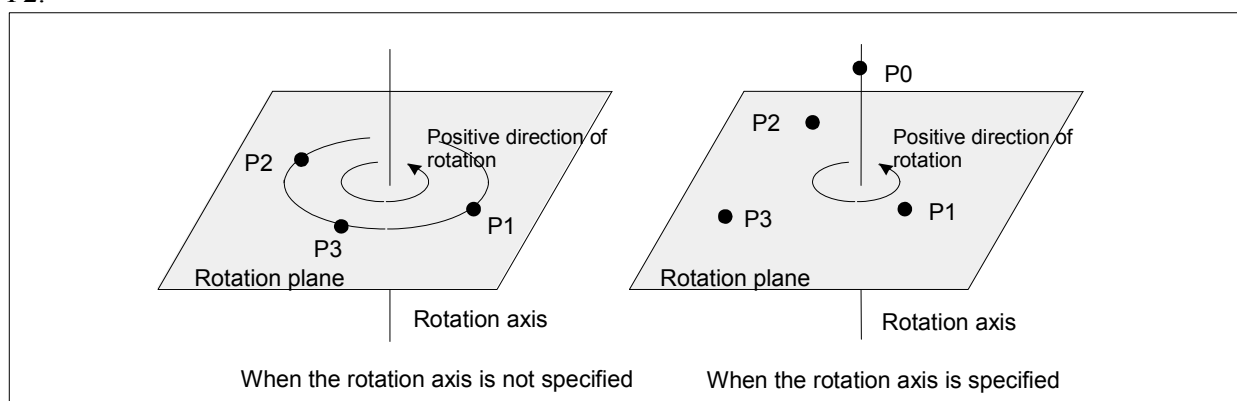


Fig. 9.2.3 (a) Specifying the Angle-Input Shift Function

To execute the angle-input shift function, use the angle-input shift screen ANGLE ENTRY SHIFT. The figure below shows how to navigate through the angle-input shift screen.

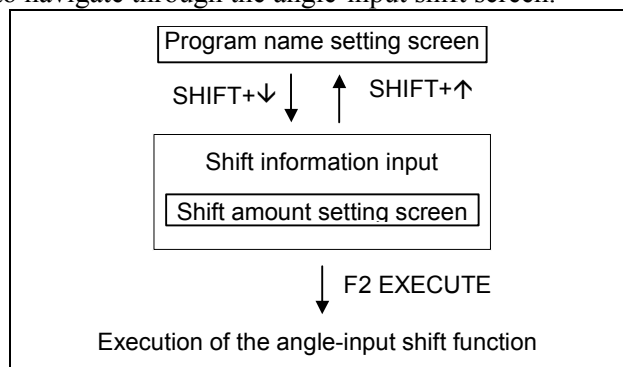


Fig. 9.2.3 (b) Angle-Input Shift Screen

The items on the program name setting screen are the same as those on corresponding screen for the program shift function.

The shift amount setting screen contains the following items:

Table 9.2.3 Contents of the Shift Amount Setting Screen

Item	Description
Rotation plane	Specifies the positions of the representative points for determining the rotation plane. If the rotation axis is not specified, these points must be on the same circumference so that the rotation center can be calculated. If the rotation axis is specified, the representative points need not necessarily be on the same circumference. The positions must be specified with coordinates in the world coordinate system.
Rotation axis enable	Specifies how the rotation axis is to be determined from the representative points. The representative points must be specified differently depending on the setting made for this item.
Rotation axis	Specifies the position of representative point P0 for determining the rotation axis. This item is available only when Rotation axis enable is set to TRUE. Only representative point P0 can be specified directly with position data (numeric values) in any coordinate system. To specify P0 directly, position the cursor to this item and press the Enter key. The rotation axis direct specification screen appears.
Angle	Specifies the angular displacement (in degrees) by which the shift is to be performed with the rotation axis and plane determined with the representative points. Enter an signed real number directly. (The plus sign need not be entered.) The direction of rotation is regarded as being positive when the rotation is from representative point P1 to P2.
Repeating times	Specifies the conversion iteration. If the locations to be machined are arranged at equal intervals on the same circumference, specifying the iteration allows the user to machine all the locations by specifying a single location. If the iteration is 2 or greater, a comment line is automatically inserted at the beginning of the program resulting from the shift. Consider the following example: Source program: Program A 1:J P[1] 100% FINE 2:L P[2] 1500mm/sec FINE If conversion is performed with the "angular displacement" set to 20°, "iteration" set to 3, and "destination program" set to program B, program B will be as follows: 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 in program B is as follows: P[1]: Position resulting from rotating P[1] in program A by 20° P[2]: Position resulting from rotating P[2] in program A by 20° P[3]: Position resulting from rotating P[1] in program A by 40° P[4]: Position resulting from rotating P[2] in program A by 40° P[5]: Position resulting from rotating P[1] in program A by 60° P[6]: Position resulting from rotating P[2] in program A by 60°
REFER	F4, REFER allows the use of a position variable or position register in the source program as the position of a representative point.

Procedure 9-7 Executing the angle-input shift function

Condition

- The program on which the shift is to be performed exists.

Step

- 1 Press MENU. The screen menu appears.

- 2 Select 1, UTILITIES.
- 3 Press F1, [TYPE]. The screen switching menu appears.
- 4 Select Angle entry. The program name input screen appears.

1 UTILITIES 2 TEST CYCLE MENUS Angle entry TYPE F1	ANGLE ENTRY SHIFT JOINT 10 % Program 1/6 1 Original program: [Test1] 2 Range: WHOLE 3 Start line: (not used) **** 4 End line: (not used) **** 5 New program: [Test1] 6 Insert line: (not used) **** Use shifted up,down arrows for next page [TYPE] > CLEAR >

- 5 Specify the necessary items.
- 6 After specifying the items, go to the next screen with SHIFT + ↓. The shift amount setting screen appears. To return to the previous screen, use SHIFT + ↑.

SHIFT ↓	ANGLE ENTRY SHIFT JOINT 10 % Shift amount Position data of P1 X:***** Y:***** Z:***** 1 Rotation Plane P1: 2 P2: 3 P3: 4 Rotation axis enable: FALSE 5 Rotation axis P0:Not used 6 Angle (deg): 0.00 7 Repeating times: 1 [TYPE] EXECUTE REFER RECORD>

- 7 For shift with the rotation axis specified, set "Rotation axis enable" to TRUE. If required, specify "Repeating times".
- 8 Specify the representative points.

REFER RECORD> SHIFT F5	1 Rotation Plane P1:Recorded 2 P2: 3 P3:

- 9 For reference point input, press F4, REFER. Select F4 [P] or F5 [PR] to enter arguments.

REFER RECORD> F4 P[] PR[] > F4	1 Rotation Plane P1:Recorded 2 P2:P[5] 3 P3:

- 10 Enter the angular displacement.

- 11 After setting the shift information, press F2 EXECUTE to execute the shift.

[TYPE] EXECUTE

F2

- 12 If the turn number is changed due to the conversion, the user is notified and prompted to make a selection.

Execute transform?
YES NO

Repeat3: Select P[1]:J6 angle (183°)
183° -177° uninit QUIT >

F4

- 13 F1 indicates the axial angle associated with the changed (optimized) turn number.
F2 indicates the axial angle associated with the original turn number.
F3 uninit causes the data to become unspecified data.
F5 QUIT interrupts the conversion.
Select one of the above keys.
- 14 To directly enter the position data for representative point P0, position the cursor to the P0 line and press the Enter key. The rotation axis direct specification screen appears.

ANGLE ENTRY SHIFT		JOINT 10 %
Shift amount		
Rotation center axis direct entry		
1 Frame	USER FRAME1	
2 X:		0.00
3 Y:		0.00
4 Z:		0.00
[TYPE] EXECUTE		[CHOICE] >
CLEAR		>

- 15 To specify the position of representative point P0 with numeric values in any coordinate system, position the cursor to line Frame and press F4, [CHOICE]. From the menu that appears, select the desired coordinate system.

[CHOICE] >

F4

- 16 Provide the other necessary shift information has been set, press F2, EXECUTE to execute the shift.

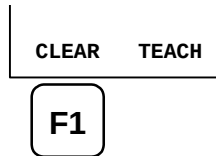
[TYPE] EXECUTE

F2

Execute transform?
YES NO

F4

- 17 To erase all the shift information, press F1, CLEAR on the next page.



9.3 COORDINATE SYSTEM CHANGE SHIFT FUNCTIONS

The coordinate system change shift functions change the tool coordinate system (tool) or user coordinate system for the operation instructions within a certain range of an already taught program, and convert the position data so that the TCP is located at the same position, considering the shift amount resulting from the change from the old to the new coordinate system.

NOTE

The coordinate system change shift functions allow the user to specify that the position data not be converted.

Coordinate system change shift functions

The coordinate system change shift functions perform the following:

- Change the tool coordinate system or user coordinate system number in the position data (Cartesian coordinates) for the operation instructions within the entire range or within a certain range of an existing program.
- If the position data is joint coordinates, convert the coordinates considering the shift amount resulting from the tool change or user coordinate system change.
- Insert the shift results into a new or existing program.
- Execute the same shift on another program.

Position data conversion

The following rules apply to converted position data:

Rules for positions and attitudes:

- Position data with Cartesian coordinates is converted to Cartesian coordinates. Position data with joint coordinates is converted to joint coordinates.
- If converted joint coordinate position data falls outside the joint operating area, it is stored as unspecified. Converted Cartesian coordinate position data is stored as is even if it falls outside the joint operating area.
- Position data in the position registers is not converted.
- Position data with joint coordinates for operation instructions involving incremental instructions is stored as unspecified.

Rules for the configuration (joint placement and turn number) of position data with Cartesian coordinates:

- The configuration is not changed due to conversion.
- For the turn number, if the conversion causes rotation about the wrist axis by 180° or more, the turn number for the axis is optimized, and a message appears so that the user can decide whether to accept it.

For the tool change shift functions, select the desired position data conversion method from the following:

- TCP fixed: The original position of the tool end point is preserved in the converted data.

For example, TCP fixed is useful if the previously used hand was damaged and replaced by a new one. By setting the tool coordinate system number of the old hand for Old UTOOL number and the tool coordinate system number of the new hand for New UTOOL number and using a tool change shift function with TCP fixed, the TCP of the new tool is moved to the original specified point correctly.

- Robot fixed: The original attitude of the robot (joint positions) is preserved in the converted data.

For example, Robot fixed is useful if the program was taught in a tool coordinate system different from that used by the actually mounted hand and the correct tool coordinates are set later. By setting the tool number used when the program was taught for Old UTOOL number and the correct tool coordinate system number for New UTOOL number, and using a tool change shift function with Robot fixed, the program can operate in the correct tool coordinate system, with the same positions as the originals.

The coordinate change shift functions allow the user to specify whether to convert position data.

- Perform conversion: Position data is converted so that the TCP is located at the same position.
- Do not perform conversion: Position data is not converted even if the coordinate system number is changed.

Types of coordinate system change shift functions

The following coordinate system change shift functions are supported:

- Tool change shift function: Changes the tool coordinate system number in the position data.
- Coordinate change shift function: Changes the user coordinate system number in the position data.

To execute the coordinate system change shift functions, use the change shift screen TOOL OFFSET (UFRAME OFFSET). The figure below shows how to navigate through the change shift screen.

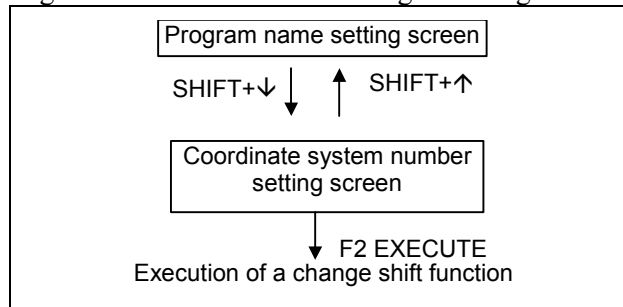


Fig. 9.3 Coordinate System Shift Screen

Procedure 9-8 Executing the tool change shift function

Condition

- The program on which the shift is to be performed exists.

TEST1			JOINT 30 %
			1/6
1:J	P[1]	100% FINE	
2:J	P[2]	70% CNT50	
3:L	P[3]	1000cm/min CNT30	
4:L	P[4]	500m/sec FINE	
5:J	P[1]	100% FINE	
[END]			
POINT			TOUCHUP >

Step

- 1 Press MENUS. The screen menu appears.
- 2 Select 1 UTILITIES.
- 3 Press F1 TYPE. The screen switching menu appears.
- 4 Select Tool offset. The program name input screen appears.

1 UTILITIES	JOINT 10 %
2 TEST CYCLE	

MENUS

Tool offset

TYPE

F1

TOOL OFFSET		JOINT 10 %
Program		1/6
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		
[TYPE]		>
CLEAR		>

5 Specify the necessary items.

6 After specifying the items, go to the next screen with SHIFT + ↓. The coordinate system number setting screen appears. To return to the previous screen, use SHIFT + ↑.

SHIFT

↓

TOOL OFFSET		JOINT 10 %
UTOOL number		1/6
1 Old UTOOL number		1
2 New UTOOL number		2
3 Convert type		TCP fixed
[TYPE]	EXECUTE	>
CLEAR		>

7 Enter the current and new tool coordinate system numbers.

To set F as the new tool coordinate system number, enter 15.

8 Press F2 EXECUTE to execute the shift.

[TYPE] EXECUTE

F2

9 If the turn number is changed as a result of the conversion, the user is notified and prompted to make a selection.

Select P[3]:J5 angle (183°)			
183°	-177°	uninit	QUIT >

10 F1 indicates the axial angle associated with the optimized turn number.

F2 indicates the axial angle associated with the original turn number.

F3 uninit causes the data to become unspecified data.

F5 QUIT interrupts the conversion.

- 11 To erase all the shift information, press F->">" and press F1 1 CLEAR on the next page.

CLEAR

F1



CAUTION

When the tool change shift function is performed, the tool coordinate system number selected by the system is changed to the new tool number.

Procedure 9-9 Executing the coordinate change shift function

Condition

- The program on which the shift is to be performed exists.

```

TEST1                                JOINT 30 %
                                     1/6
1:J P[1] 100% FINE
2:J P[2] 70% CNT50
3:L P[3] 1000cm/min CNT30
4:L P[4] 500m/sec FINE
5:J P[1] 100% FINE
[END]
POINT                                TOUCHUP >

```

Step

- Press MENUS. The screen menu appears.
- Select 1, UTILITIES.
- Press F1, [TYPE]. The screen switching menu appears.
- Select Frame offset. The program name input screen appears.

```

1 UTILITIES
2 TEST CYCLE

MENUS

Frame offset
TYPE

F1

```

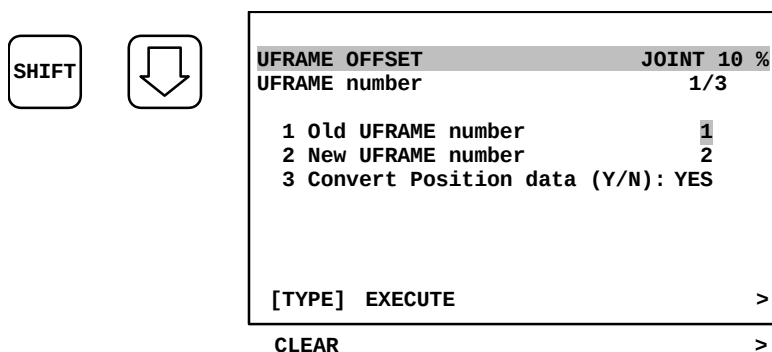
```

UFRAME OFFSET                        JOINT 10 %
Program                             1/6
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
[TYPE] >
CLEAR >

```

- Specify the necessary items.
- After specifying the items, go to the next screen with SHIFT + ↓. The coordinate system number setting screen appears. To return to the previous screen, use SHIFT + ↑.



- 7 Enter the current and new user coordinate system numbers.
To set F as the new user coordinate system number, enter 15.
- 8 Press F2, EXECUTE to execute the shift.

[TYPE] EXECUTE

F2

- 9 If the turn number is changed as a result of the conversion, the user is notified and prompted to make a selection.

Select P[3]:J5 angle (183°)
183° -177° uninit QUIT >

- 10 F1 indicates the axial angle associated with the optimized turn number.
F2 indicates the axial angle associated with the original turn number.
F3 uninit causes the data to become unspecified data.
F5 QUIT interrupts the conversion.
- 11 To erase all the shift information, press F->">" and then press F1 CLEAR on the next page.

CLEAR

F1



CAUTION

When the coordinate change shift function is executed, the user coordinate system number selected by the system is changed to the specified new user coordinate system number.

9.4 SOFT FLOAT FUNCTION

Usually, the robot moves accurately toward the goal specified using the teach pendant (taught point). When the robot is used to mount workpieces on a machine tool, variances in workpiece precision may result in a shift in the workpiece position relative to the tool, thus possibly causing interference between the workpiece and tool.

A soft float function has been added which is effective in mounting workpieces with variances in precision onto a machine tool.

The soft float function is also very effective if the synchronization speed is unstable as in the extraction of workpieces in sync with hydraulic extrusion, and if workpieces that the robot cannot grip accurately, such as rough-machined workpieces, are to be handled.

Function

The soft float function works as follows:

- Two types of soft float are supported: joint soft float for specifying the softness related to the direction of rotation of each arm of the robot, and Cartesian soft float for specifying the softness on the Cartesian axes.
- The function is enabled/disabled using an instruction in the program. Its conditions are also specified using the instruction.
- "Servo flexibility" can be specified for each axis. The term servo flexibility indicates how strongly the axis resists external forces. It is specified between 0% and 100%. A servo flexibility of 100% corresponds to being the most flexible. The servo flexibility is specified using a condition table that contains a set of data for one group (for nine axes).
- If an external force above a certain level (so high as to overcome a static frictional force) is applied to a robot, the axis of the robot is pressed and moved.
- An external force applied to a robot may prevent it from reaching the taught point. The distance between the taught point and the point the robot can reach is nearly proportional to the magnitude of the external force.
- If static load is applied to a robot, the robot controls force to maintain its attitude even if the soft float function is enabled.

The detailed descriptions of the soft float function follow.

Program instruction

The following three program instructions related to the soft float function are supported.

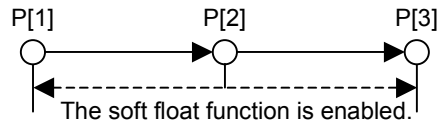
- **SOFTFLOAT[n]**
The soft float function is enabled using condition n.
* The setting of soft float condition is explained in "Condition setting menu".
- **SOFTFLOAT END**
The soft float function is disabled.
- **FOLLOW UP**
When an external force is removed from a robot, it usually tries to go back to the taught point. However, this instruction causes the robot to assume that the current position is the taught point, and prevents it from going back to the taught point.

Soft float function effective range

The SOFTFLOAT[n] instruction can be used in two modes; in one mode it is used solely in a program and in the other mode it is used as an auxiliary motion instruction after a motion statement. The range in which the soft float function is effective for robot operation is determined according to which mode this instruction is used.

- **Sole instruction**
The soft float function is enabled after the end of the motion specified on the line preceding the solely specified SOFTFLOAT[n] instruction.
In the following example, the soft float function is enabled after the motion specified on line 1 ends, and disabled by SOFTFLOAT END on line 5.

```
1: J P[1] 100% FINE
2: SOFTFLOAT[1]
3: L P[2] 100mm/sec FINE
4: L P[3] 100mm/sec FINE
```

5: SOFTFLOAT END**- Auxiliary motion instruction**

The soft float function becomes enabled during execution of a motion statement attached with a SOFTFLOAT [n] instruction.

The point at which the soft float function becomes enabled is determined by a soft float condition "Exec Start Ratio."

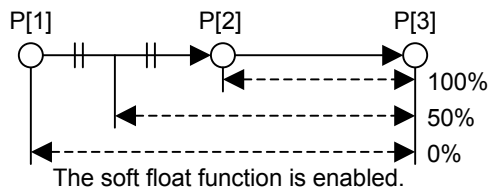
Auxiliary motion instruction is specified as the ratio (from 0% to 100% in 1% steps) of a distance to be traveled before the robot reaches the taught point corresponding to a motion statement attached with a SOFTFLOAT[n].

In the following example, the soft float function is effective between P[1] taught using a motion statement on line 1 and P[2] taught using a motion statement on line 2 attached with the SOFTFLOAT[n] instruction.

```

1: J P[1] 100% FINE
2: L P[2] 100mm/sec FINE SOFTFLOAT[1]
3: L P[3] 100mm/sec FINE
4: SOFTFLOAT END

```

**NOTE**

The auxiliary motion instruction is not supported by Cartesian soft float.

- Condition setting menu

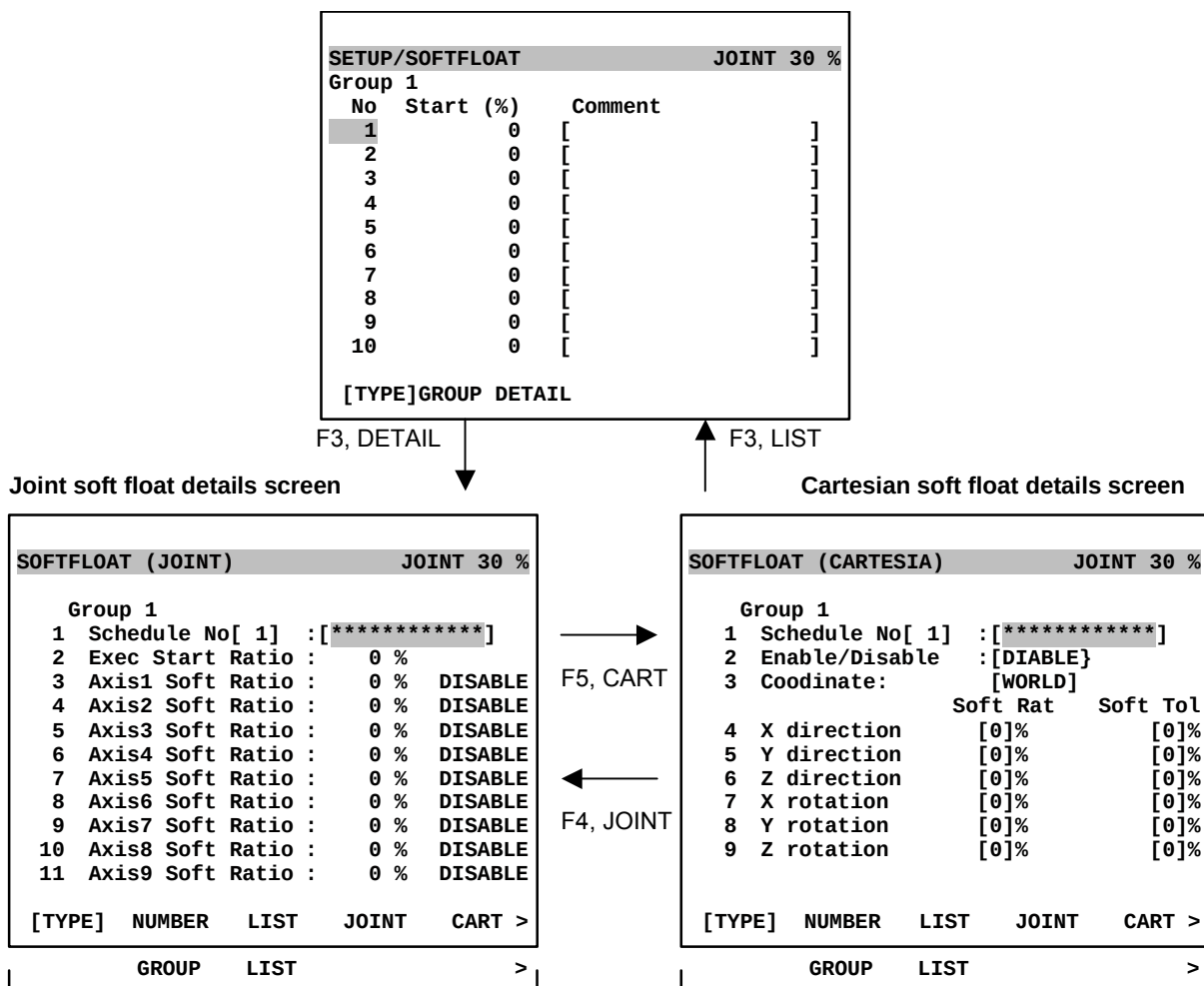
The soft float conditions are specified on the [SETUP.Softfloat] menu, which consists of the following two menus.

- List menu
- Detail menu

A function key is used to select either menu.

- Pressing the F3 (DETAIL) key on the list menu selects the detail menu.
- Pressing the F3 (LIST) key on the detail menu selects the list menu.

Up to 10 conditions can usually be specified for the soft float function.



The following data can be specified on the detail menu.

Table 9.4 Setting items of Soft float detail menu

Items	Descriptions
Comment	Soft float condition number. By default, ten numbers can be set. Pressing the input key with the cursor on line 1 enables entering a comment. The comment text can be specified in the same way as on other menus.
Soft float start ratio	Line 2 specifies the point where the soft float function is enabled if the SOFTFLOAT [n] is used as an auxiliary motion instruction. See "Soft float function effective range" for the soft float start ratio.

Items	Descriptions
Servo flexibility	Servo flexibility for each axis can be specified on line 3 and the subsequent lines. The servo flexibility indicates how strongly the axis resist external forces. It is specified between 0% and 100%. A flexibility of 100% corresponds to being the most flexible. Whether the soft float function is enabled/disabled can be specified for each axis on line 3 and the subsequent lines. Setting the cursor at the rightmost end (enabled/disabled setting position) of each line causes the F4 (ENABLE) and F5 (DISABLE) keys to appear. Use these keys to specify whether to enable/disable the soft float function. NOTE Pressing the F2 (NUMBER) key selects another page of the detail menu for other conditions. <pre> 10 Axis8 Soft Ratio : 0 % DISABLE 11 Axis9 Soft Ratio : 0 % DISABLE [TYPE] NUMBER LIST > ↓ 10 Axis8 Soft Ratio : 0 % DISABLE 11 Axis9 Soft Ratio : 0 % DISABLE [TYPE] NUMBER LIST ENABLE DISABLE> </pre>
Enable/Disable	When this item is set to DISABLE, soft float cannot be executed.
Coordinate	Select one of WORLD, USER, and TOOL. NOTE If the remote TP is used, USER indicates the coordinate system on the remote TCP.
X direction	Set the softness on or around the X-, Y-, and Z-axes. If Soft Rat increases, the spring constant decreases, allowing the robot to move with less force. If Soft Tol increases, the maximum force and moment applied by the robot in that direction decreases, allowing the robot to move with less force. The difference between Soft Rat and Soft Tol is illustrated below.

Operation area limitation

Operation area during Cartesian softfloat is able to be limited.

If the position error between TCP of motion command and actual TCP in the frame which is set in softfloat schedule exceeds the threshold, alarm occurs and robot stops.

For safety, following system variables have to be set to avoid interference before the program which executes Cartesian softfloat runs.

To reflect new value of these system variables, power OFF/ON is necessary.

- \$param_group[g].\$cb_mass = 1.0 :enable limitation
- \$param_group[g].\$cb_ix = Threshold in X direction [mm]
(0.0 occurs alarm)

- \$param_group[g].\$cb_iz = Threshold in Z direction [mm]
(0.0 occurs alarm)
- \$param_group[g].\$cb_iz = Threshold in Z direction [mm]
(0.0 occurs alarm)

Cautions/restrictions

When using the soft float function, observe the following cautions/ restrictions.

- Restrictions imposed when the soft float function is enabled
 - It is not guaranteed that the robot always follows the taught path.
 - The taught route changes according to override.
 - The required operation time may be prolonged compared with normal operation.
- 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 servo.
 - Jog feed is performed with the program at pause
 - The program is restarted after the cursor is moved manually with the program at pause.
 - Backward execution is performed.
 - Power is applied
- If the program is caused to pause, then restarted, the states of the soft float function (such as enabled/disabled and the soft float start ratio) are set to the conditions which exist before the program is caused to pause, except for the cases in the above operation, where the soft float function is disabled.
- The soft float function cannot be enabled by any method other than the SOFTFLOAT instruction.
- When the soft float function is enabled, the robot moves in the CNT 0 mode (no position check is made) even if FINE has been specified as motion statement positioning mode.
- When the soft float function is enabled, if an external force causes the robot to move beyond a certain distance, the following servo alarms occur.
 - 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 alarm occurs:
[INTP-216 (program name, line number) Invalid value for group number]
- The range of motion with the soft float function enabled should be minimized. A weight balance may vary depending on the soft float ratio and travel distance, thus shifting the vertical axis upward or downward.

The range of motion with an auxiliary motion instruction issued should also be minimized for the same reason. In addition, the speed of motion should be kept low.

- 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 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.
- This function cannot be used with arc tools.
- Force from outside should not be added when softfloat starts, when softfloat ends and when follow up is doing.

If softfloat starts with not only gravity force but also other force from outside, robot will move above or below, and "Stop/Move error excess" or "Softfloat time out" alarm will occur.

In such cases, please modify TP program by changing position data or by adding wait instruction as softfloat starts and ends without other force.

NOTE**Follow-up**

With the soft float function, external forces are applied to the robot so that it operates at positions slightly different from those specified. When the external force is removed after the completion of the operation, the robot usually attempts to move back to a specified point abruptly. Follow-up prevents this abrupt movement.

9.5 CONTINUOUS ROTATION FUNCTION

The continuous rotation function allows continuous and limitless rotation about the final axis or an additional rotation axis of the robot in one direction.

NOTE

For example, the "final axis" refers to the J6 axis of a robot having six axes.

For example, this function is useful for rotating those devices that require continuous rotation, such as conveyers, pumps, and grinders, about a robot axis or additional rotation axis.

To specify the items for this function, such as disable/enable, use the SETUP Continuous T screen (new).

The start and stop of continuous rotation are directed from a program.

Before this function can be used, the setup necessary for continuous rotation must be performed.

Only a single continuous rotation axis can be allocated for each operation group. The axis must satisfy the following conditions:

- Final axis of the robot
- Final axis of the built-in additional rotation axes
- Any of the normal additional rotation axes
- Final axis of the independent additional axes

The continuous rotation axis must satisfy the following mechanical conditions:

- The mechanism must allow continuous operation (must be free of obstacles such as stoppers).
- The gear reduction ratio (value of Numerator of Gear Ratio/Denominator of Gear Ratio on the setting screen, the speed of the motor required for one rotation about the axis) must be 4000 or less.

To use this function, an option (continuous rotation function) is required.

Function

When this function is enabled, the axis allocated as a continuous rotation axis allows limitless rotation. The angle on the axis is, therefore, represented by a relative degree within $+180^\circ$, not by an absolute one. For example, the figure below shows rotation from 0° to 200° in the positive direction. The angle on the axis after the rotation is -160° , not 200° .

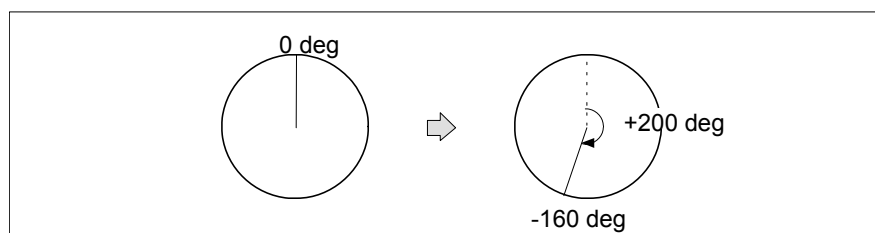


Fig. 9.5 (a) Angle on the Continuous Rotation Axis

When this function is enabled but continuous rotation is not performed (see the next page for an explanation of how to use continuous rotation), rotation is performed about the continuous rotation axis from the current

angle to the target angle in whichever direction incurs the least amount of motion. (Usually, the direction of rotation about the axis is determined with the relationship between the current and target angles.) This "shorter-way operation" is effective in reducing the cycle time.

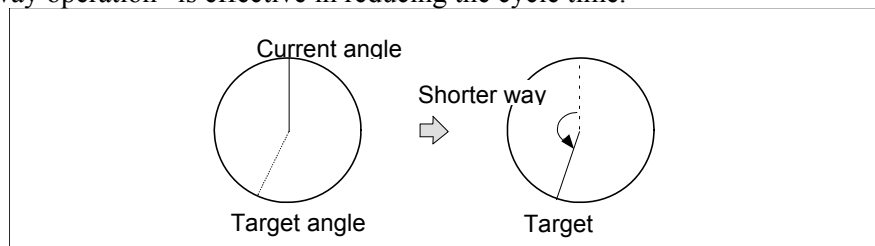


Fig. 9.5 (b) Shorter-Way Operation

Setup

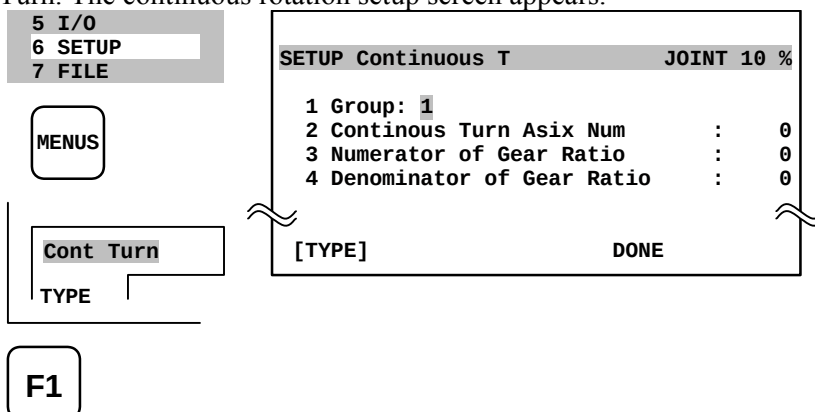
To use the function,

- Perform setup on the SETUP Continuous T screen and
- Specify the start/stop of continuous rotation with the operation add instruction, "continuous rotation speed instruction."

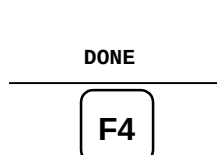
Procedure 9-10 Setting up the continuous rotation function

Step

- 1 Press MENU. The screen menu appears.
- 2 Select SETUP.
- 3 Press F1, [TYPE]. The screen switching menu appears.
- 4 Select Cont Turn. The continuous rotation setup screen appears.



- 5 Specify the necessary items using the numeric and other keys.
 - To disable the continuous rotation function, set "0" for Continuous Turn Axis Num.
 - The maximum value for Numerator of Gear Ratio and Denominator of Gear Ratio is 32766.
 - Set the operation group number for Group. If a different number (number of the operation group to be viewed) is entered in this field, the other settings are changed to those of the operation group.
- 6 After specifying the items, press F4 DONE. The following message appears.



- 7 Turn off the power, then turn it back on with a cold start.
The items on the continuous rotation setup screen are described below.

Table 9.5 Contents of the Continuous Rotation Setup Screen

Item	Description
Group	Set the operation group number.
Continuous Turn Axis Num	Set the number of the continuous rotation axis. If "0" is set, this function is disabled for the operation group.
Numerator of Gear Ratio Denominator of Gear Ratio	Set the gear reduction ratio for the continuous rotation axis set for the above item. A value from 0 to 32766 can be set for each item. The items must, however, satisfy the following: $\text{Numerator of Gear Ratio} \div \text{Denominator of Gear Ratio} \leq 4000$

Using the function

After setting up the continuous rotation axis, specify the start point of continuous rotation using the operation add instruction, "continuous rotation speed instruction."

The following "continuous rotation speed instruction" is supported. The "continuous rotation speed instruction" must be specified as an operation add instruction.

* The specification method is the same as that for other operation add instructions, and is therefore omitted. (→ Subsection 5.3.4, "Teaching an Supplementary Motion Instruction".)

- Continuous rotation speed instruction CTV

* where $i = -100$ to 100 , which is the ratio of the rotation axis speed to the maximum axis speed (%)

Starting continuous rotation

Continuous rotation is started as soon as an operation statement with a continuous rotation speed instruction added is started.

Stopping continuous rotation

Continuous rotation is stopped when the first operation statement with no continuous rotation speed instruction added is started since a continuous rotation speed instruction was started.

When continuous rotation is stopped, the operation on the other axes for the same operation group also terminates. The robot, therefore, decelerates even if the positioning format for the previous operation is CNT.

The robot starts decelerating to stop on the continuous rotation axis after it has completely stopped on the other axes. At this time, the robot is not necessarily at the specified position on the continuous operation axis. Thus, the synchronization of the operation on the continuous rotation axis with the operation on the other axes (including those for other operation groups) is lost.

If an operation statement is specified next, the robot rotates in the same direction as the previous continuous rotation direction to move to the specified position.

Notes

- Continuous rotation continues even if logic instructions (instructions other than those in operation statements) are executed.
- During program playback, the turn number for the continuous rotation axis is ignored, and is always assumed to be "0."
- The turn number for the continuous rotation axis at a point specified when this function is enabled is always stored as "0."
- If the rotation axis speed for a continuous rotation speed instruction is specified as "0," continuous operation is not performed. If an operation statement is specified next, shorter-way operation is performed on the continuous rotation axis. This feature is useful if continuous rotation about the continuous rotation axis is to be stopped temporarily but temporary stop of the robot due to the end of the continuous rotation is to be avoided. (See the next section, "Example of use.")
- In single-step execution (both forward and backward), continuous rotation is not performed even if a continuous rotation speed instruction is added; shorter-way operation is performed.
- Continuous rotation stops due to a hold. If program execution is subsequently restarted, if the target position has already been reached on axes other than the continuous rotation axis, continuous rotation

is not performed. If the target position has not been reached on axes other than the continuous rotation axis, continuous rotation is restarted.

- Continuous rotation about the continuous rotation axis is possible from jog feed.

Example of use

The following shows an example of using the continuous rotation speed instruction.

```

1:J P[1] 100% FINE
2:J P[2] 100% CNT100 CTV100
3:J P[3] 100% FINE
4:J P[4] 100% CNT100 CTV100
5:J P[5] 100% FINE CTV0
6:J P[6] 100% FINE
7:J P[7] 100% FINE CTV100
8:WAIT 100.0sec
9:J P[8] 100% FINE

```

- Description of lines 1 to 3:
During operation from P[1] to P[2], continuous operation is performed. Although the positioning format specified on line 2 is "CNT", the robot decelerates (stops temporarily on all axes at the start of the operation on line 3) because a continuous rotation speed instruction is not added to the next line, line 3.
- Description of lines 4 to 6:
Continuous rotation starts as soon as the execution of line 4 starts. Because the rotation axis speed specified with the continuous rotation speed instruction on line 5 is 0, continuous rotation stops temporarily at the start of the execution of line 5. Because continuous rotation continues, the positioning format CNT100 on line 4 is valid and the robot does not decelerate.
When line 6 is executed, shorter-way operation is performed on the continuous rotation axis.
- Description of lines 7 to 9:
Continuous rotation starts at the start of operation on line 7. Continuous rotation continues during the execution of the wait instruction (logic instruction) on line 8.
The robot stops temporarily on all axes at the start of operation on line 9, and continuous rotation stops.

Notes/restrictions

Note the following when using this function:

- When continuous rotation is to be performed on a robot axis or built-in additional axis, The X and Y components of the tool coordinate system must both be 0. (Only the Z-axis component can have a value other than 0.)
If this condition is not satisfied, the path of linear or circular operation cannot be guaranteed in normal operation other than continuous rotation.
- This function cannot be used together with the following functions:
 - Asynchronous addition axis speed instruction. (The synchronous additional axis speed instruction can be used.)
 - Arc sensor
 - Weaving
 - TCP speed estimation function (sealing flow rate control)
 - Incremental instruction (INC) for Continuous rotation axis. (Can use only for non-continuous rotation axis or the continuous rotation axis is disabled case)
- This function automatically updates the mastering data (for the continuous rotation axis only) according to the amount of rotation about the continuous rotation axis. Thus, previously recorded mastering data may not match the current mastering data.
After this function is disabled, it is not necessary to perform mastering.
- When this function is disabled, the current position on the continuous rotation axis may fall outside the stroke limits. If this occurs, move the position on continuous rotation axis within the stroke limits using jog feed or a program.

- If, on a multi-group system, the settings on the SETUP Continuous T screen are changed and the F4 DONE key is pressed, it is necessary to set system variable \$PARAM_GROUP[group].\$SV_OFF_ENBL[i] (where i is an axis number) to FALSE to disable break control for all the axes for all operation groups before turning the power back on with a cold start.
- On a multi-group system, even if there are multiple continuous rotation axes, separate continuous rotation speeds cannot be specified for them.
- At the end of continuous rotation, one or more rotations about the continuous rotation may be performed to ensure smooth deceleration and stop. (The amount of rotation differs depending on the acceleration/deceleration constant.)
- Even during backward execution (single-step execution), shorter-way operation is performed on the continuous rotation axis. If, therefore, forward step execution and backward execution are performed sequentially in an operation statement with the movement angle being very close to 180°, rotation may be performed about the continuous rotation axis in the same direction during the forward and backward executions.
- Original Path Resume feature is disabled when continuous turn loaded. Even if system variable (\$SCR.\$ORG_PTH_RSM) set enabled, feature is disabled when cycle power.
- If continuous turn axis is in any of the motion groups, Cycle Time Priority feature for cartesian motion and Constant Path feature are disabled in all groups.
 - If Cycle Time Priority feature for cartesian motion is disabled, the cycle time is possible to be longer.
 - If Constant Path feature is disabled in servogun system, alarm “SVGN-0149 CP enable is required” is possible to be posted.
- Following setup allows Cycle Time Priority feature for cartesian motion and Constant Path feature to work in those motion groups that DOES NOT involve continuous turn axis.
 - (1) Set bit22 of \$SCR.\$UPDATE_MAP1 TRUE [Example 1].
 - (2) Set \$SCR_GRP[g].\$USE_TBCART = TRUE.
 - (3) Set \$GROUP[g].\$CARTFLTRENB = TRUE.
 - (4) Set \$CPCFG.\$CP_ENABLE = TRUE (Default TRUE).
 - (5) Reset bit 0, bit4 and bit10 of \$SCR.\$MB_DSBL_MSK False [Example 2].
 - (6) If \$SGSYSCFG.\$LIN_DST_ENB and \$SGSYSCFG.\$LIN_DST_ENB are changed for SVGN-149 countermeasure, reset TRUE.
 - (7) Cycle power.

[Example 1] In case of \$SCR.\$UPDATE_MAP1 = 52983009

- Set the value binary.
11 0010 1000 0111 0100 1110 0001 (0x32874E1)
- Set bit22 TRUE.
11 0110 1000 0111 0100 1110 0001 (0x36874E1)
- Reset the value decimal and update the system variable.
57177313

[Example 2] In case of \$SCR.\$MB_DSBL_MSK = 3089

- Set the value binary
1100 0001 0001 (0xC11)
- Set bit0, bit4 and bi10 FALSE.
1000 0000 0000 (0x800)
- Reset the value decimal and update system variable.
2048

Note that above setting affects robot behavior such as path and cycle time.
Please execute program carefully after above setup.

- Note that following limitations when Constant Path feature is allowed to work.
 - Continuous turn does not work in Robot (except Independent axis and Positioner). If continuous turn axis is set in robot, “CPMO-004 Feature not Supported (G:%d^2) ” alarm is posted at starting program.
 - Only Independent axis and Positioner are allowed to involve continuous turn axis.
 - For multi group program, if continuous turn axis is in any of the groups, alarm “CPMO-004 Feature not Supported (G:%d^2) ” is posted. To use continuous turn in a program, please separate groups that involve continuous turn axis from groups that DOES NOT involve continuous turn axis. [Example 3]

[Example 3] Following configuration is supported to use continuous turn in Constant Path and Cycle Time Priority enabled program.

- G1 : Robot, G2 : Servogun, G3 : Continuous turn axis (Independent axis with 1 axis)
 - Program 1 : G1 + G2 (Robot + Servogun) --- < 1, 1, * >
 - Program 2 : G3 (Continuous Turn axis) only --- < *, *, 1 >
- In Program1, Constant Path and Cycle Time Priority are active.
- In Program2, Continuous Turn can work in a program.
- To combine Constant Path and Cycle Time Priority with continuous turn, run Program2 in Program1 concurrentl

9.6 POSITION REGISTER LOOK-AHEAD EXECUTION FUNCTION

While the robot is executing a program, it reads the lines ahead of the line currently being executed (look-ahead execution).

Conventionally, look-ahead execution was performed for motion statements having normal position data (not using position registers). Look-ahead execution could not be performed for motion statements that used position registers for their position data.

Motion statements using position registers could not 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 statement using a position register prior to its execution, the value of the position register may yet be changed by a program or another function (such as data transfer). Such a change is not reflected in the motion statement that has already been read by the robot. Consequently, the robot's operation may be unpredictable.

Motion statements that use position registers can be classified into two types:

- Motion statements with the target position specified by a position register
- Motion statements 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 statement, look-ahead execution was not performed for the statement, for the reason explained above.

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 the registers are newly provided. By means of these instructions, the user can explicitly specify a program portion. Then, for the specified program portion, even when it contains motion statements that use position registers, look-ahead execution can be performed.

Function

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 alarm message is issued:

[INTP-128 Pos reg is locked]

When a function (such as the data transfer function) other than the program attempts to change the value of a locked position register, the following alarm message is issued, and the attempt fails:

[VARS-037 Pos reg 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.

NOTE

Before using position register instructions, lock position registers. When position register instructions are used with the position registers unlocked, operation may become tight.

Operation

The following program instructions have been added:

- **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 (See Subsection 5.3.5, "Teaching a Control Instruction").

Example

The following shows how to use the LOCK PREG and UNLOCK PREG instructions in a program:

```

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 statements 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 statements in lines 7 and 8.

NOTE

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.)

Notes

Note the following when using this function:

- The LOCK PREG and UNLOCK PREG instructions are not executed in backward program execution mode.
- Look-ahead 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.

9.7 OPERATION GROUP DO OUTPUT FUNCTION

The operation group DO output function outputs information about the operation groups that are capable of jog feed, and about the operation groups of the programs being executed/temporarily stopped, to an external device with a digital output signal (DO) or robot output signal (RO). This allows devices other than the teach pendant to recognize the currently effective operation groups, thus improving safety. This function is effective when the multi-group option is used.

Function

This function allows the allocation of two DOs (jog signal and program signal) to a single operation group. For DOs, any digital output signals or robot output signals of the robot can be used.

Each allocated DO signal turns on/off under the following conditions:

- Jog signals

When the teach pendant is disabled, all signals turn off.

When the teach pendant is enabled, the signal for the currently selected operation group on the teach pendant turns on, while the other signals turn off.

- Program signals

Regardless of whether the teach pendant is enabled or disabled, the signal for the operation group of the program currently being executed/temporarily stopped turns on. (The signal does not turn on when the program is merely selected.)

If other programs are being executed/temporarily stopped with the multitask option, the signals for the operation groups of these programs also turn on.

Setup

To set up the operation group DO output function, use the [Setup - Motion DO] screen.

To change the signal number for an operation group, move the cursor to the signal number and enter a new value.

Motion group DO			JOINT 10 %
GROUP NO.	PROGRAM	JOG	
1	RO[1]	RO[2]	
2	DO[3]	DO[3]	
3	RO[0]	RO[0]	
[TYPE]		RO	DO

To change the type of a signal, position the cursor to the signal number and press function key F4 "RO" or F5 "DO."

To disable a signal, set the number of the signal to "0."

The same signal can be set for both the program and jog signals for the same operation group. In this case, the output signal is the OR of the two signals. That is, the signal turns on if either the program or jog signal turns on. (The signal turns off only if both the program and jog signals turn off.)

Example of using this function with the multitask option

This section explains the operation of this function when a subprogram call or the multitask option is used. The output program signal is the OR of the signals for all the operation groups of the program currently being executed or temporarily stopped.

If a program without an operation group calls a program having an operation group by using a subprogram call, the signal for the operation group of the subprogram turns on only while the subprogram is being executed. (The signal does not turn on when the main program without an operation group is merely selected/executed.)

If RUN instruction of the multitask function is to start another program that operates the robot (the main program that has RUN instruction does not have an operation group), the signal for the operation group of the program started by RUN instruction does not turn on when the main program is merely selected/executed. The program signal turns on when the program that operates the robot is actually started. Consider the following three example programs:

```

PROGRAM MAIN : Operation group[*,*,*,*,*,*,*]
1:RUN PRG A
2:RUN PRG B
:

PROGRAM PRG A : Operation group[1,*,*,*,*,*,*]
1:J P[1] 100% FINE
:

PROGRAM PRG B : Operation group[*,*,*,*,*,*,*]
1:L P[1] 500mm/sec CNT100
:

```

Program MAIN, which does not have an operation group, starts PRG A and PRG B having operation groups by using RUN instructions. PRG A uses operation group 1 and PRG B uses operation group 2.

- The program signals for the groups do not turn on when program MAIN is merely selected.
- When line 1 of MAIN is executed, PRG A is started and the signal for operation group 1 turns on.
- When line 2 of MAIN is executed, PRG B is started and the signal for operation group 2 turns on.
- When PRG A and PRG B terminates, the respective signals for operation groups 1 and 2 turn off.

Notes

Note the following when using this function:

- The same signal cannot be defined for different operation groups.
- While a program is being executed/temporarily stopped, the type (DO or RO) and number of the program signal cannot be changed.

9.8 TIME BEFORE FUNCTION

This function calls a subprogram and output a signal before or after the specified time at which robot operation is to terminate.

For example, if a signal output instruction is specified in a subprogram, this function allows a signal to be output during robot operation. It can also eliminate the wait time associated with the transfer of data to and from peripheral devices, thus reducing the cycle time.

Function

This function allows a main program to call and execute a subprogram before or after the specified operation termination time. Signal output is executed before or after the specified operation termination time.

Using an instruction in a program, specify the time at which a subprogram is to be called (in seconds). (This specified time is called the execution start time.) The time at which operation terminates is assumed to be 0 seconds, which differs depending on the positioning type (FINE or CNT).

Using an instruction in a program, specify the name of the subprogram to be called.

The TIME BEFORE (or AFTER) instruction is an operation add instruction. Both the subprogram name and execution start time must be specified with the operation add instruction.

Instruction statement

Specify the execution start time and subprogram after an operation statement.

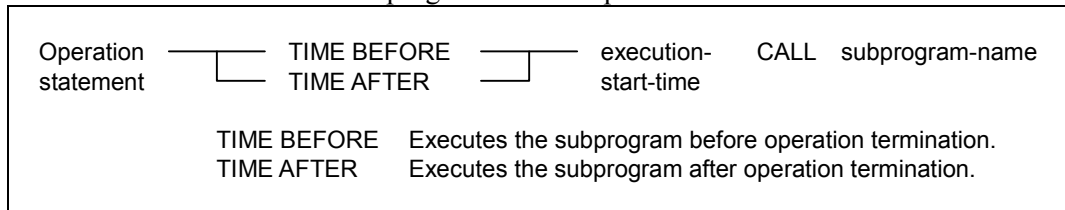


Fig. 9.8 (a) TIME BEFORE/AFTER Instruction (Operation Add Instruction)

Example

```

1:J P[1] 100% FINE
:TIME BEFORE 1.0sec CALL OPEN HAND

1:J P[1] 100% FINE
:TIME AFTER 1.0sec CALL OPEN HAND

```

Description of execution start time

According to the specified execution start time, the subprogram is executed at the following time:

If execution start time, "n" seconds, is specified with a TIME BEFORE instruction, the subprogram is executed n seconds before operation termination.

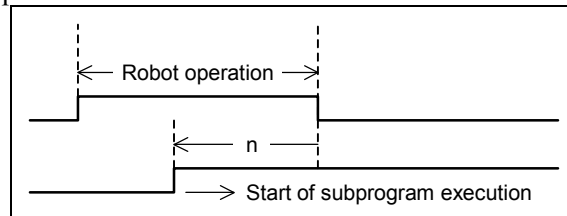


Fig. 9.8 (b) Timing of Subprogram Execution (TIME BEFORE Instruction)

If execution start time, "n" seconds, is specified with a TIME AFTER instruction, the subprogram is executed n seconds after operation termination.

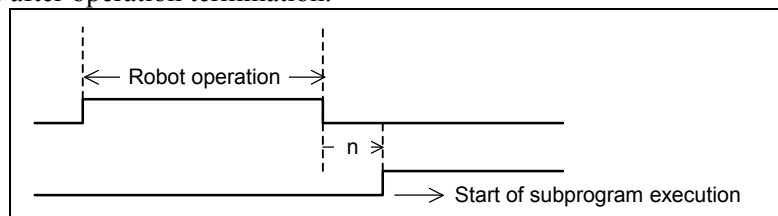


Fig. 9.8 (c) Timing of Subprogram Execution (TIME AFTER Instruction)

If the execution start time specified with a TIME BEFORE instruction exceeds the operation time, the subprogram is executed as soon as operation starts.

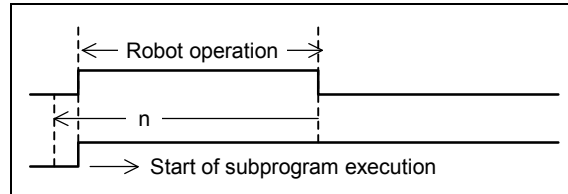


Fig. 9.8 (d) Timing of Subprogram Execution (TIME BEFORE Instruction)

The execution start time that can be specified in a program is

- 0 to 30 seconds for a TIME BEFORE instruction
- 0 to 0.5 seconds for a TIME AFTER instruction



CAUTION

Even if the robot operation time is changed due to a change in the override, the time at which subprogram execution is to start depends on the execution start time. The execution start position of the subprogram is, therefore, changed due to a change in the override.

Search/replace functions

- Search function

By selecting CALL program for search item CALL, the search function searches for the call instructions of TIME BEFORE instructions.

- Replace function

- By selecting replace item TIME BEFORE/AFTER, TIME BEFORE/AFTER replacement and execution start time replacement can be performed.
- By selecting CALL program of replace item CALL, the subprogram names for TIME BEFORE instructions can be replaced.

Single step

When an operation statement with an execution start TIME BEFORE instruction specified is executed in single-step mode, operation stops temporarily at the time when the subprogram is called. Subsequently, the rest of the operation is executed in sync with single-step execution of the subprogram.

Power failure handling

If power failure handling is enabled and the power is removed during subprogram execution, execution starts with the remaining instructions of the subprogram due to a restart after the power is turned on again. In this case, the subprogram is executed with the position the robot was located when the power was removed. Thus, the subprogram is executed with timing different from the usual timing. Great care must be taken regarding this point.



WARNING

Be sure you know the point where the robot is going to begin robot motion when power is restored after a power failure. Otherwise, you could injure personnel or damage equipment.

Procedure 9-11 Specifying the TIME BEFORE instruction

Step

- 1 Position the cursor to the operation add instruction specification area (space following an operation instruction).

```

PNS0001                                JOINT 10 %
                                      1/2
  1:J P[1] 100% FINE
[End]

[CHOICE]

```

- 2 Press function key F4 CHOICE. A list of operation add instructions appears.

```

Motion Modify                        JOINT 10 %
  1                                5
  2                                6
  3 TIME BEFORE                     7
  4 TIME AFTER                      8
PNS0001                                1/2
  1:J P[1] 100% FINE
[End]

[CHOICE]

```

- 3 Select item TIME BEFORE.

```

PNS0001                                JOINT 10 %
                                      1/2

  1:J P[1] 100% FINE
    :TIME BEFORE sec ...
[End]

Enter value

[CHOICE]

```

- 4 Specify the time and press the Enter key. Example: 2 seconds.

```

TIME statement                        JOINT 10 %
  1 CALL program                     5 GO [ ]=
  2 CALL program( )                  6 AO [ ]=
  3 DO [ ]=                          7
  4 RO [ ]=                          8
PNS0001                                1/2
  1:J P[1] 100% FINE
    :TIME BEFORE 2.0sec
[End]

Select item

[CHOICE]

```

- 5 Select item CALL program.
Select item CALL program() to use the argument.

PROGRAM list		JOINT 10 %
1	Open hand	5
2	Close hand	6
3		7
4		8
PNS0001		
		1/2
1:J P[1] 100% FINE		
:TIME BEFORE 2.0sec, CALL		
[End]		
Select item		
STRINGS		

6 Select item Open hand.

PNS0001		JOINT 10 %
		1/2
1:J P[1] 100% FINE		
:TIME BEFORE 2.0sec, CALL Open hand		
[End]		
Select item		
[CHOICE]		

Program example

Main program: PNS0001

```

1:J P[1] 100% FINE
2:J P[2] 100% CNT100
:TIME BEFORE 1.0sec CALL Open hand
3:CALL Close hand

```

Subprogram: Hand open

```

1:DO[1]=on

```

Operation performed when the main program is executed

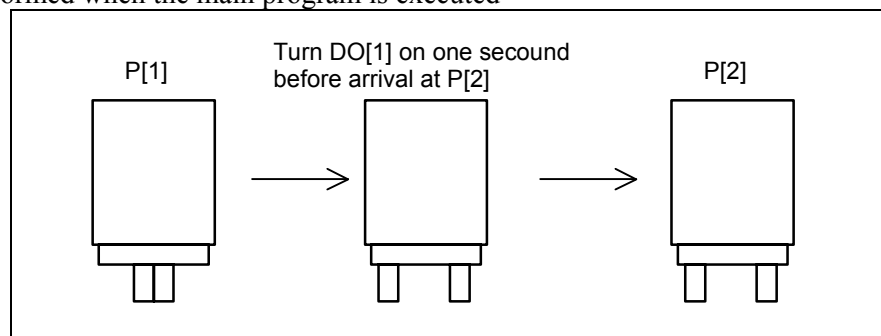


Fig. 9.8 (e) Program Example Using a TIME BEFORE Instruction

Notes/restrictions

In the subprogram specified for Call, operation statements cannot be specified. (The operation group in the subprogram must be [*, *, *, *, *, *, *, *].)

Since the called sub routine and the main program are concurrently executed, the main program is sometimes executed earlier than the called sub routine.

When you do not want to proceed with the execution of the main program ahead until the execution of the called sub routine is finished, please change a system variable as follows.

\$TIMEBF_VER=3 (the standard value) → 2

No limit is imposed on the number of instructions that can be specified in a subprogram.

The TIME BEFORE/AFTER add instructions can be used together with other operation add instructions (except application instructions such as spot [] and skip instructions).

If the positioning specification for an operation statement is CNT, the time of operation termination changes depending on the degree of CNT. The time at which the subprogram is called changes accordingly. Depending on the situation, even if the execution time is set to 0 seconds with a TIME BEFORE instruction, the subprogram may be executed too quickly. If this occurs, use a TIME AFTER instruction.

If a TIME BEFORE instruction is specified on the last line of a main program, the execution of the main program may terminate before the subprogram is called, in which case, the subprogram is not called. Do not, therefore, specify a TIME BEFORE instruction on the last line of a program.

For direct specification of signal output, only DO, RO, GO, and AO are supported.

If the override is modified dynamically while the motion instruction which has TIME BEFORE instruction is executed, the time at which the subprogram is called may not be proper.

9.9 DISTANCE BEFORE FUNCTION

9.9.1 Overview

This function calls program or outputs signal when TCP is going into a region which is within specified distance from destination point. This program call and signal output is done on a parallel with main program execution.

Example

- 1 J P[1] 100% FINE
- 2 L P[2] 1000mm/sec FINE DB 100mm,CALL A

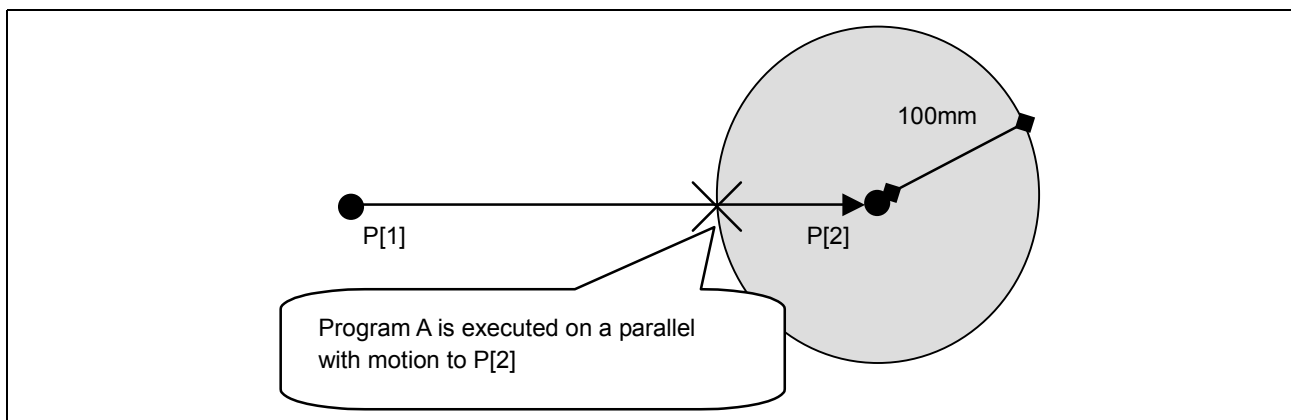


Fig. 9.9.1 Execution timing of Distance Before

9.9.2 Specification

Item	Specification	Limitation
Distance value	0.0 to 999.9[mm]	Distance value and actual execution timing is different. The error depends on speed of TCP.

Item	Specification	Limitation
Trigger condition (*1)	TCP goes into a region, which is within specified distance from destination point.	Distance value and actual execution timing is different.
Available instructions	- Signal output (ex. DO[1] = ON) - CALL program	Program to be called cannot use motion group. Only logic instruction is available.

NOTE (*1)

This is condition to process instruction part.

9.9.3 Configuration

Before using Distance Before, set following system variable.

\$SCR_GRP[1].\$M_POS_ENB = TRUE

9.9.4 Instruction

1 Format

Distance Before is taught in following format.

Motion statement + DB distance value, instruction part

Example

L P[2] 1000mm/sec FINE DB 100mm, CALL A

└ Instruction part (Please refer to 3.)

└ Distance value (Please refer to 2.)

NOTE

Distance Before is a motion option. You cannot use DB as a standard instruction.

2 Distance value

(i) Distance value

Distance Before executes instruction part when TCP goes into a spherical region whose center is destination point. Distance value decides the radius of this sphere. Distance value is taught in millimeter. Distance value is from 0 to 999.9mm. This sphere is referred as trigger region hereafter.

- 1: L P[1] 2000mm/sec FINE DB 100.0mm DO[1] = ON
- 2: L P[2] 2000mm/sec FINE DB 100.0mm DO[1] = ON

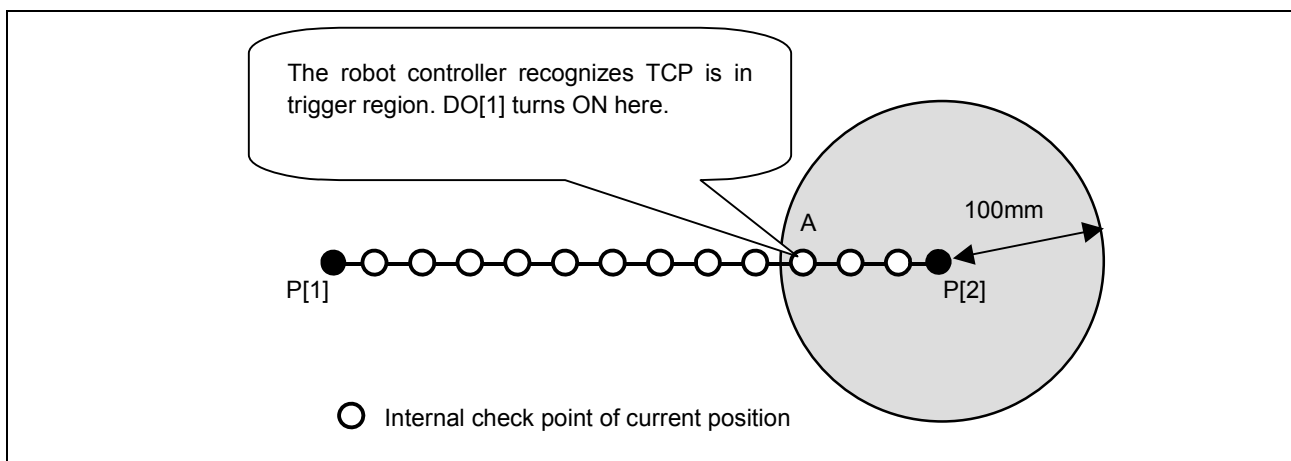


Fig. 9.9.4 (a) Cyclical checks if TCP goes into trigger region.

Internally, Robot controller calculates current position to judge if TCP is in trigger region or not. Instruction part is executed when this calculated position is in trigger region.



CAUTION

Execution timing of instruction part is decided by distance (in millimeter). Because judgment to trigger is done by calculating distance between current position and destination point, actual execution timing is different from distance value.
(Error in case of 2000mm/sec is estimated around 16mm)

(ii) Radius of trigger region.

Radius of trigger region is as follows.

$$\text{Radius} = (\text{distance value or } \$DB\_MINDIST) + \$DB\_TOLERANCE$$

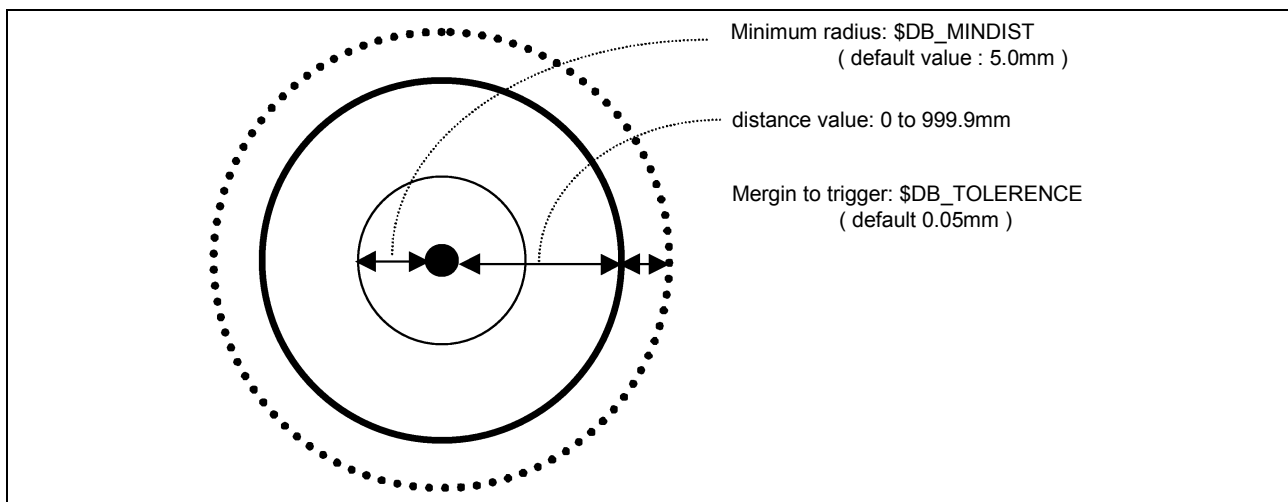


Fig. 9.9.4 (b) The size of trigger region

If distance value is less than \$DB_MINDIST, \$DB_MINDIST is used as distance value.

Example Suppose following motion statement is taught with \$DB_MINDIST = 5.0

```
L P[1] 2000mm/sec FINE DB 0.0mm DO[1]=ON
```

In this case, Robot controller interprets it as DB 5.0mm. Then \$DB_TOLERANCE is added to decide radius of trigger region. Consequently, radius of trigger region is 5.05mm with default system variables.

3 Instruction part

This part shows what is done when TCP goes into trigger region. DB can do following action.

- CALL program
- Signal output

(i) DB Call program

Specified program is executed when condition is triggered.

Program to be called cannot use motion group.

(Change group mask to [*,*,*,*,*,*,*] in program header information screen.)

You can use arguments to call program.

Example)

```
L P[2] 1000mm/sec FINE DB 100mm, CALL A_(1,2)
```

(ii) DB signal output

You can teach following signal output.

You can use one signal output for one DB.

DO[] RO[]	=	ON OFF R[] pulse
GO[] AO[]	=	Constant R[] AR[]

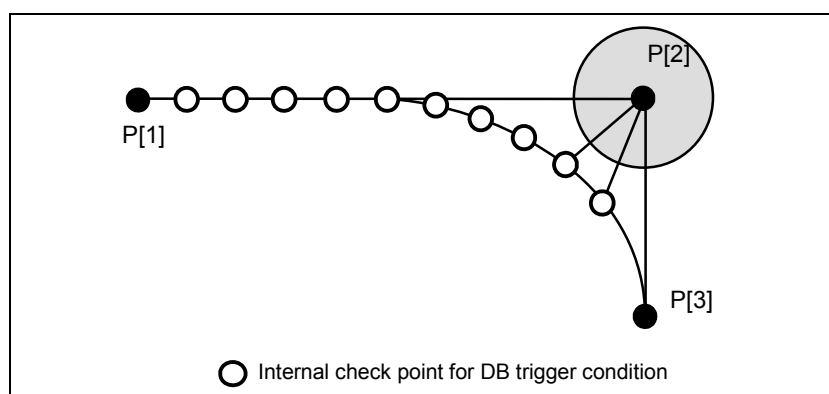
You can also output signal by calling program which use signal output instruction. But to output only one signal with one DB, this direct signal output is better. It's easier to read and maintain.

4 Changing trigger condition

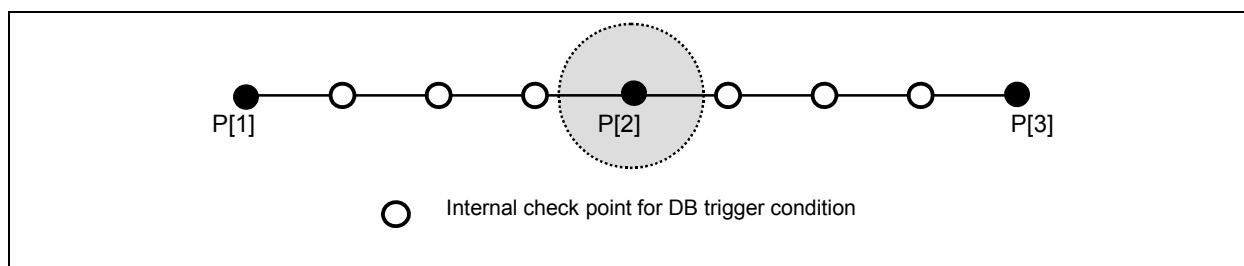
Instruction part is executed when Robot controller recognizes that TCP is in trigger region.

But in some cases like following “going away” and “penetrate”, robot controller doesn't recognize that TCP is in trigger region. These cases are described in this section.

Case 1 Trajectory of CNT motion doesn't go through trigger region.(“going away”)



Case 2 Trigger region is too small for controller to check current position in time.(“penetrate”)



For these case, the condition for instruction part to be executed (referred as DB condition) is changed by \$DB_CONDTYP.

\$DB_CONDTYP	DB condition	When alarm is posted.
0	TCP is in trigger region. (“region trigger”) + end of motion (*2)	“going away” + “penetrate” + end of motion
1 (default value)	“region trigger” + “going away” + “penetration” + end of motion	end of motion + (“going away”) (*1)

\$DB_CONDTYP	DB condition	When alarm is posted.
2	"region trigger"	"going away"

"going away" and "penetration" is defined in (i), (ii) and (iii) respectively.

Distance Before executes instruction part when DB condition is satisfied. Otherwise, posts alarm. There are two alarms for not-triggered DB. They are INTP-293 and INTP-295. \$DBCONDTRIG decides which alarm is posted. Message is same but severity is different. Please refer to 5 for details.

NOTE

- (*1) When Distance Before is triggered by "going away" in case of \$DB_CONDTYP = 1, you can post alarm in addition to execution of instruction part. Please refer to 4 (i) for details.
- (*2) By default configuration, if motion statement with Distance Before completes and robot stops before neither "region trigger" nor "going away" nor "penetration" trigger happens, Distance Before executes instruction part and post alarm. Please refer to 4 (iii).

(i) In case of going away.

If termination type is CNT and distance value is small, TCP may not go into trigger region.

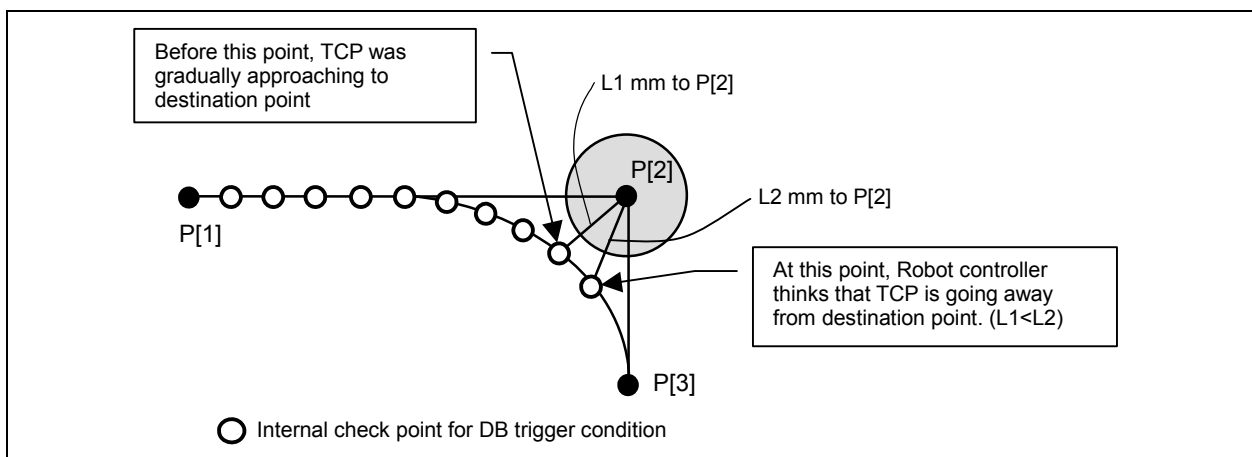


Fig. 9.9.4 (c) TCP doesn't go into trigger region.

In case of Fig. 9.9.4 (c), TCP doesn't go into trigger region. TCP starts to go away from destination point (P[2]). Robot controller cyclically judges if TCP is going away from destination point or not in addition to DB condition. Robot controller recognizes that TCP is going away when calculated distance between current position and destination point is greater than distance between destination point and nearest point by more than (\$DB_AWAY_TRIG) millimeter. This case is referred as "going away" in this manual.

- To post alarm in addition to execution of instruction part only when the DB is triggered by "going away" trigger, set \$DB_AWAY_ALM to TRUE. DB executes instruction part and post following alarm.

INTP-295 (program name, line number) DB too small (away) (%dmm)
This is warning.

(ii) Penetration

This function cyclically checks if DB condition is triggered or not. Because of this cyclical check, CNT motion with high-speed may cause for Robot controller to omit cyclical check in small trigger region. See Fig.9.9.4 (d).

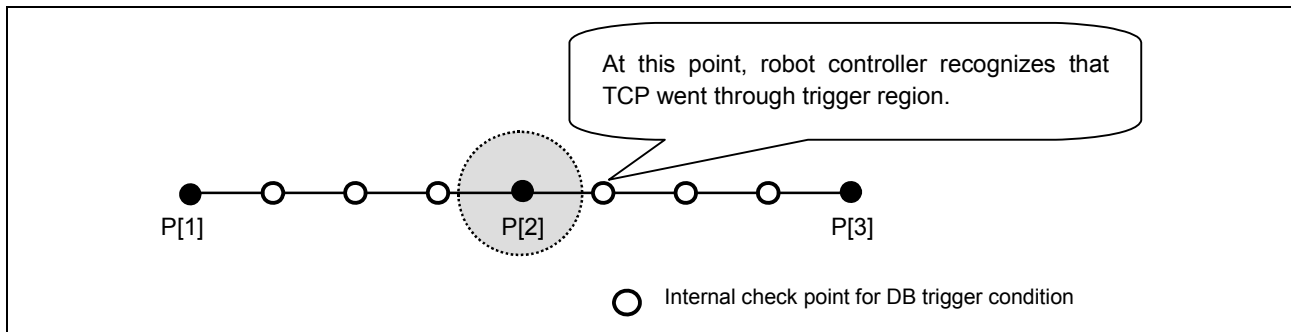


Fig. 9.9.4 (d) Penetration

In this case, TCP moves too fast for robot to check DB condition in small trigger region. Because cyclical check is done outside of trigger region, the fact TCP is in trigger region is not recognized by the robot controller. This case is referred as “penetration” in this manual.

To handle cases like Fig. 9.9.4 (d), Distance Before checks if TCP went through trigger region or not. If trajectory of TCP penetrated trigger region (penetration), instruction part is executed by default configuration.

But in this case, execution of instruction part is done after TCP passed away destination point.

- Motion with termination type FINE doesn’t cause trigger by “penetration”.

(iii) End of motion

If motion statement with DB completes and robot stops before “region”, “going away” and “penetration” is satisfied, DB executes instruction part and post following alarm.

INTP-297 (program name, line number) DB too small (done) (mm).

This alarm is not posted by FINE motion.

If you don’t want this trigger, set \$DB_MOTNEND to FALSE (default value: TRUE).

Distance displayed by this alarm is distance to destination.

⚠ CAUTION

- 1 If you stop your robot by E-stop when motion statement is about to complete, Distance Before may be trigger just after resume of the program.
- 2 If you halt a program when motion statement with DB is near its completion, DB may not be triggered. In this case, Distance Before executes its instruction part after resume of program.

5 Alarms for not -triggered Distance Before

Distance Before posts alarm if condition is not triggered. What is posted depends on \$DBCONDTRIG.

\$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 configuration, INTP-295 is posted. Because severity of this alarm is WARN, execution of program doesn’t stop.

If you want to halt program when condition was not triggered, set \$DB_CONDTRIG to 1. INTP-293 is posted when condition was not triggered. Program is halted for severity of this alarm is PAULSE.L. Robot decelerates to stop. Displayed distance is recommended value for the DB to be triggered by region trigger.

6 Step execution

If Distance Before CALL program is executed by step execution, program is halted at the timing sub program is called. The rest of motion statement is done by next step execution that executes sub program step by step.

Step execution of motion statement with DB signal output is just same as motion statement without DB except signal output is done.



CAUTION

If distance value is small, program may be halted before completion of motion and before DB conditions are satisfied. In this case, Distance Before is not triggered by step execution of the line it is taught. The DB is triggered by execution of next line.

7 Halt and resume

Halt and resume of motion statement with DB changes its radius of trigger region. After resume, radius of trigger region is changed to minimum radius ($\$DB\_MINDIST + \$DB\_TORELENCE$).

The purpose of this process is to execute instruction part after TCP reaches to its destination point. This prevents earlier trigger because of halt and resume.

This means that halt and resume of program changes trigger timing of Distance Before. Not to change radius of trigger region, set $\$DISTBF\_TTS$ to 0 (default value: 1).

Example Default configuration

Suppose following program is executed.

```
1: L P[1] 2000mm/sec FINE
2: L P[2] 2000mm/sec CNT100 DB 100.0mm CALL SUB
3: L P[3] 2000mm/sec CNT100
```

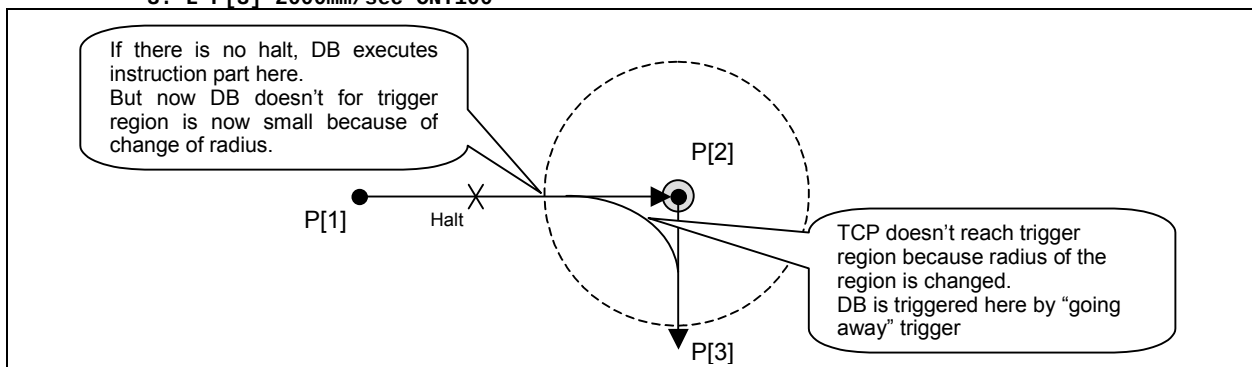


Fig. 9.9.4 (e) Trigger timing after resume of program.

Example Resume with $\$DISTBF\_TTS = 0$

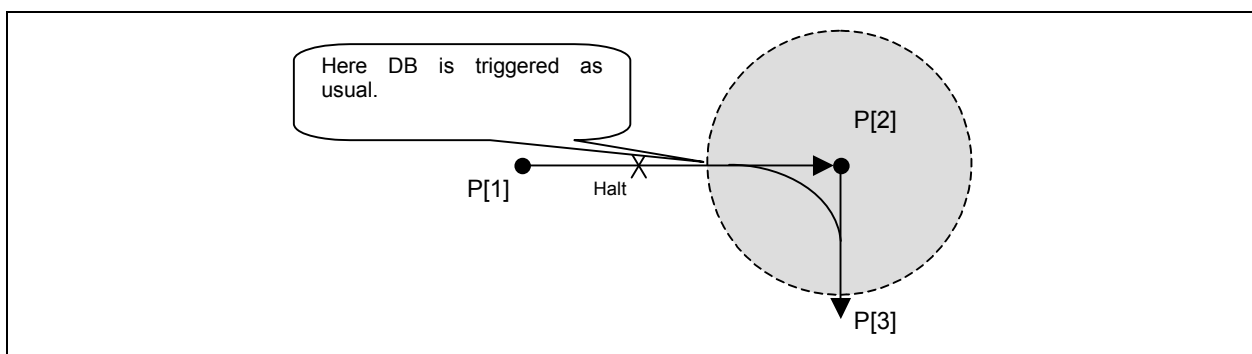


Fig. 9.9.4 (f) $\$DISTBF\_TTS = 0$

8 Resume after JOG

If you halt motion statement with DB, JOG robot and resume program, execution timing depends on TCP position at the instant of program resume.

Because this procedure is accompanied by program halt, execution timing depends on $\$DISTBF\_TTS$, too.

(i) Default configuration ($\$DISTBF\_TTS = 1$)

After resume of program, radius of trigger region changed to minimum value ($\$DB\_MINDIST + \$DB\_TOLERANCE$). If TCP is in new (diminished) trigger region, DB is triggered just after resume of program. If not, DB is triggered when DB condition is satisfied.

Example

Suppose following program is executed and halted on line two. DB condition is not triggered yet.

```

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 CNT100

```

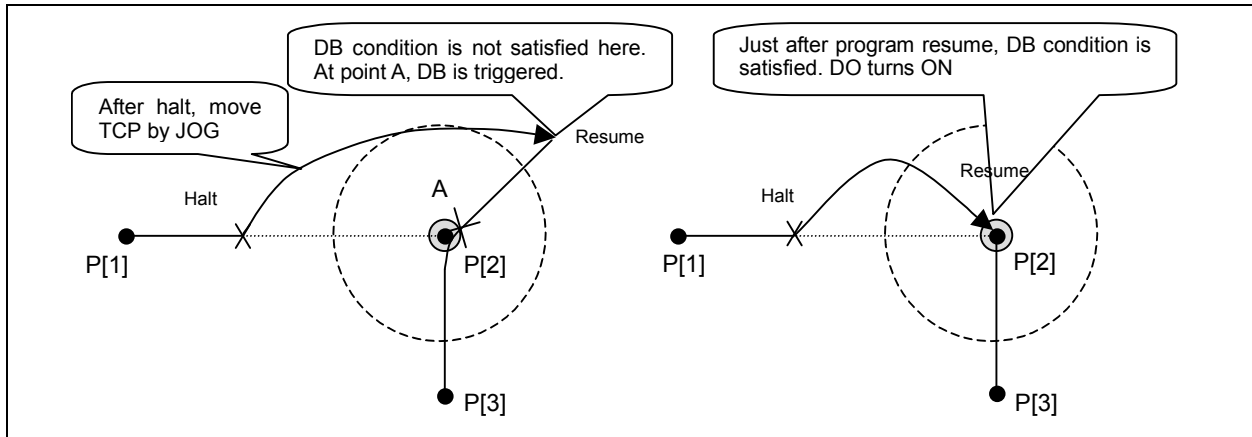


Fig. 9.9.4 (g) Resume after JOG

(ii) \$DISTBF_TTS = 0

Radius of trigger region is not changed.

If TCP is in trigger region, DB is triggered just after resume of program. If not, DB is triggered when DB condition is satisfied.

Example)

Suppose following program is executed and halted on line two.

DB condition was not satisfied yet.

```

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 CNT100

```

If TCP is distant from P[2] enough not to trigger (more than 100mm away), DO[1] turns ON when DB condition is triggered by motion after resume, at point A in left diagram in Fig. 9.9.4 (h).

If TCP is in trigger region when you resume program, DO[1] turns ON just after resume. (right diagram in Fig. 9.9.4 (h)).

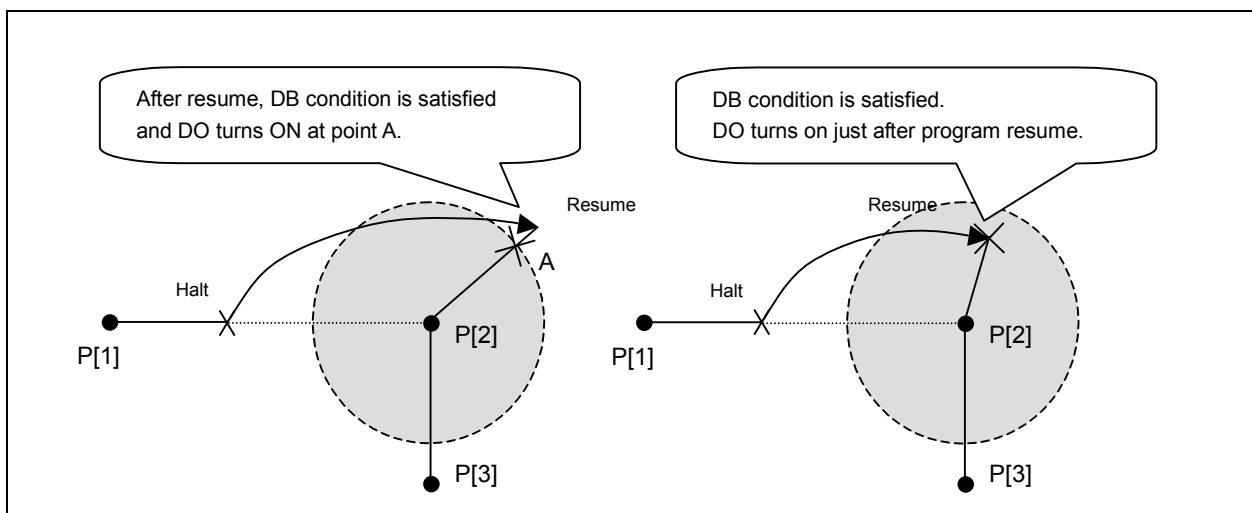


Fig. 9.9.4 (h) Resume after JOG(\$DISTBF_TTS = 0)

9 Power failure recovery

If power is turned down during sub program execution and power failure recovery is enabled, resume after power failure recovery executes the rest of sub program. In this case, sub program is executed where TCP was at power failure. Execution timing is different from usual one.

9.9.5 Entering Distance before

1 DB call program

- 1) Move cursor to motion option area.

```

PNS0001      LINE 1      ABORTED
PNS0001      JOINT 10%
                1/2

    1: J P[1] 100% FINE
[End]

                [CHOICE]

```

- 2) Press F4. List of motion option is displayed.

```

Motion Modify      JOINT 10%
1 TIME BEFORE      5
2 TIME AFTER        6
3 DISTANCE BEFORE   7
4                   8 ---next page---
PNS0001
                1/2

    1: J P[1] 100% FINE
[End]

Select item
                [CHOICE]

```

- 3) Select DISTANCE BEFORE. DB is added to program.

```

PNS0001      LINE 1      ABORTED
PNS0001      JOINT 10%
                1/2

    1: J P[1] 100% FINE DB 0.0mm ...
[End]

Enter value
REGISTER      [CHOICE]

```

- 4) Input distance value and press Enter. Submenu to select instruction part is displayed.

```

TIME statement      JOINT 10%
1 CALL program      5 GO[ ]=...
2 CALL program( )    6 AO[ ]=...
3 DO[ ]=...          7
4 RO[ ]=...          8
PNS0001
                1/2

    1: J P[1] 100% FINE DB 100.0mm ...
[End]

```

- 5) To use argument, select CALL program(). If you don't, select CALL program. Program list is displayed anyway.

Program list		JOINT 10%
1	HAND_OPEN	5
2	HAND_CLOSE	6
3		7
4		8
PNS0001		
		1/2
1: J P[1] 100% FINE DB 100.0mm,		
CALL ...(...)		
[End]		

- 6) Select program to call.

Program select		JOINT 10%
1	R[]	5 <None>
2	Constant	6 <Insert>
3	String	7
4	AR[]	8
PNS0001		
		1/2
1: J P[1] 100% FINE DB 100.0mm,		
CALL HAND_OPEN(...)		
[End]		

To specify argument, following procedure is needed.

- 7) Select argument type. Screen displayed below is example to use Constant.

PNS0001	LINE 1	ABORTED
PNS0001		JOINT 10%
		1/2
1: J P[1] 100% FINE DB 100.0mm,		
CALL HAND_OPEN (Constant)		
[End]		
[CHOICE]		

- 8) Input value of argument.

- To use more than 2 arguments, move cursor to “)” and press F4[CHOICE]. Submenu to select argument type is displayed. Teach argument by procedure 7) and 8) described above.
- To delete argument, move cursor to argument you want to delete and press F4. Then select <None>.

To add argument to CALL without argument, following procedure is needed.

- 1 Move cursor to program name.

1: J P[1] 100% FINE DB 100.0mm,
CALL A

- 2 Press PREV key 2 times. Following submenu is displayed.

```

TIME statement                                JOINT 10%
1 CALL program          5 GO[ ]=...
2 CALL program( )      6 AO[ ]=...
3 DO[ ]=...            7
4 RO[ ]=...            8
PNS0001

```

- 3 Select CALL program ().

```

Program list                                JOINT 10%
1 HAND_OPEN          5
2 HAND_CLOSE         6
3                     7
4                     8
PNS0001
1/2
1: J P[1] 100% FINE DB 100.0mm,
CALL ...(...)

```

- 4 Select program to call and teach argument.

2 DB Signal output

- 1) Do just same procedure 1)-4) for DB CALL program. Submenu to select instruction is displayed.

```

TIME statement                                JOINT 10%
1 CALL program          5 GO[ ]=...
2 CALL program( )      6 AO[ ]=...
3 DO[ ]=...            7
4 RO[ ]=...            8
PNS0001
1/2
1: J P[1] 100% FINE DB 100.0mm ...
[End]

```

- 2) Select signal output instruction.

```

PNS0001                                JOINT 10%
1/2
1: J P[1] 100% FINE DB 100.0mm,
DO[...]=...
[End]

Select item                                [CHOICE]

```

- 3) Input index and output value just as you do for normal I/O instruction.

```

PNS0001                                LINE 1                                ABORTED
PNS0001                                JOINT 10%
2/2
1: J P[1] 100% FINE DB 100.0mm,
DO[1]=ON
[End]

```

3 Finding/Replacing Instructions

- Finding Instructions
You can find program which is used for DB by “find” on F5 pull-up menu. By selecting “CALL” then “Call program” to find program used in DB.
You can find signal output instruction by this function, too. Select item “I/O” on submenu.
- Replacing Instructions
Distance Before can be replaced to TIME BEFORE/AFTER by “replace” on F5 pull-up menu. Select “TIME BEFORE/AFTER” on replace item submenu.
You can also replace CALL and signal output in instruction part just as you do when you replace usual CALL and DO etc.

9.9.6 Caution and Limitations

- Distance Before cannot be used with TIME BEFORE/AFTER.
- More than 6 motion statement with Distance Before cannot be processed at the same time.
- Distance Before calculates distance between current position and destination point cyclically. Because trigger condition is judged by this cyclical check, actual execution timing of instruction part is different from distance value. Instruction part may be executed inside of trigger region. This means the point where instruction is executed is closer than distance value. Degree of error depends on speed of robot. The slower TCP moves, the more accurate execution timing.
- Distance Before is not recovered by power failure recovery if it was attached to CNT motion statement and power is down when the motion is about to complete.
- Distance Before cannot be used with INC, skip and quick skip in a motion statement.
- Distance Before cannot be used in the program which has two or more groups. But, in case that the software series is 7DA5 or later, Distance Before can be used in the program which has two or more groups. In this case, the group that the group number is least is used to calculate the distance to the destination position.
- Robots that don't have Cartesian coordination are not supported.
- Position data in matrix form is not supported.
- Integrated axis is not supported.
- FANUC Robot F-200i is not supported.
- Line tracking is not supported.
- If program ends before DB condition triggers, execution instruction is not processed even if DB condition is triggered after program execution completed.
- During deceleration due to program halt, “going away” trigger may not work.
In this case, DB is triggered after program resume.
- After E-stop, DB doesn't work. If TCP passes by destination point, DB is triggered after resume of program.
- After E-stop and resume of program, DB may be triggered just after resume.
- Single step execution of DB of small distance value may fail for program is paused before motion statement completes and DB condition satisfied. The DB is triggered by execution of next line.
- If DB condition is satisfied after pause of program, DB is not triggered by step execution of the line. In this case, the DB is triggered by execution of next line.
- When there is DB/TB that is not yet triggered in previous line, DB/TB does not trigger even when trigger condition is satisfied. (DB/TB always triggers after previous DB/TB triggered.)

9.10 STATE MONITORING FUNCTION

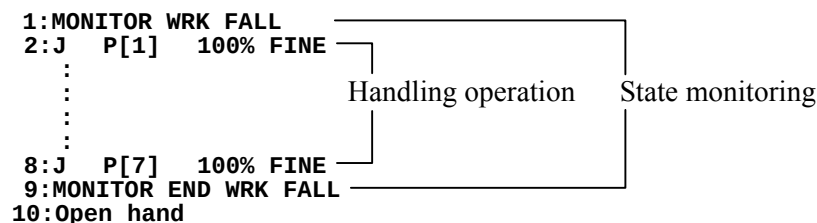
This function accepts, as conditions, the values of the input/output signals, alarms, and registers of the robot controller (referred to simply as the controller), and executes the specified programs if the conditions are satisfied. The controller itself monitors these conditions.

This function consists of the following instructions and programs:

- Monitor start instruction
Specifies the condition program to be monitored and the start of monitoring.
Example:
 1:MONITOR WRK FALL
 Condition program name
- Monitor stop instruction
Specifies the condition program to terminate.
 9:MONITOR END WRK FALL
 Condition program name
- Condition program
Describes the condition to be monitored and specifies the program to be executed if the condition is satisfied.
Program example:
 1:WHEN DI[2]=Off, CALL STP RBT
 *1 *2
This condition program states that when RI[2] turns off, STP RBT is to be called.
*1 Describe the desired monitoring condition by following instruction WHEN. The types of monitoring condition are explained in the WHEN section.
*2 Specify the program to be executed if the condition described in *1 is satisfied. The action program can be created and named in the same way as a normal program.
- Action program
Called if the condition is satisfied. The same instructions as those used in normal programs can be used.
Program example:
 1:DO[2]=On ! Notification to a peripheral device
 2:R[8]=R[8]+1 ! Drop count
 3:UALM[1] ! Alarm and robot stop
 \$UALRM MSG[1]=WORK HAS FALLEN

With the following program example, if the robot performing handling drops a workpiece, the user is alerted with an error message and the robot is stopped.

SAMPLE. TP (program for handling operation)



WRK FALL. condition (condition program)

1:WHEN DI[2]=0ff, CALL STP RBT

STP RBT. TP (action program)

```

1:DO[2]=0n           ! Notification to a peripheral device
2:R[8]=R[8]+1        ! Drop count
3:UALM[1]            ! Alarm and robot stop
[End]

```

Monitor types

There are two main types of monitors: the program monitor and system monitor.

- The program monitor starts/stops from a mnemonic program (referred to simply as a program). When the program terminates, monitoring also terminates.
- The system monitor is started/stopped from the dedicated screen. It performs monitoring constantly regardless of the execution state of the program. (Monitoring continues even after the program terminates.)

- Program monitor

This type of monitor depends on the execution state of the program. It is suitable for state monitoring within a separate program.

Monitoring starts with an instruction (monitor start instruction) in the program. Monitoring terminates with a monitor stop instruction or program termination.

The program monitor can be switched between two settings: setting 1 in which the monitor stops when the program stops temporarily, and setting 2 in which the monitor continues monitoring.

NOTE

Settings 1 and 2 cannot be used at the same time.

- System monitor

This type of monitor does not depend on the execution state of the program. It is suitable for monitoring the state of the entire system.

The monitor is started and stopped from the state screen. It cannot be operated with instructions in the program.

The system monitor can be switched between two settings: setting 1 in which the monitor stops after a cold start, and setting 2 in which the monitor continues monitoring.

NOTE

The program monitor and the system monitor can be used at the same time.

The monitors can be switched between the settings using the following system variables:

Following system variables can be changed by system variable menu under controlled start mode.

\$TPP_MON.\$LOCAL_MT = 1 or 3 Switches the program monitor to setting 1 (default).

\$TPP_MON.\$LOCAL_MT = 2 or 4 Switches the program monitor to setting 2
(Same specification as that for KAREL)

\$TPP_MON.\$GLOBAL_MT= 0 Enables the system monitor (default).

\$TPP_MON.\$GLOBAL_MT= 1 Switches the system monitor to setting 1.

\$TPP_MON.\$GLOBAL_MT= 2 Switches the system monitor to setting 2.

Notice) About \$TPP_MON.\$LOCAL_MT = 3,4

Setting \$TPP_MON.\$LOCAL_MT = 3,4 is supported in 7DA4/P07 and later version.

In 7DA4/P07 and later version, default value is changed to 3.

In case \$TPP_MON.\$LOCAL_MT = 3,4,

Action program with motion group cannot be used. (Alarm will occur when program is executed)

This is to prevent unintentional motion caused by a wrong action program with motion instruction.

When the alarm occur, please specify the operation group in action program as [*,*,*,*].

When the operation group in the action program cannot be set as [*,*,*,*], please use traditional setting. (Use \$TPP_MON.\$LOCAL_MT = 1 or 2)

Monitor state transition

The states of the monitors assumed when each operation is performed are listed in the table below:

Operation	Program monitor		System monitor	
	Setting 1	Setting 2	Setting 1	Setting 2
MONITOR instruction	A	A	E	E
RESTART (state screen)	B	B		
START (state screen)			A	A
Program Stop	C	E *1)	E	E
Program End/Enforced End	D	D	E	E
MONITOR END	D	D	E	E
PAUSE (state screen)	C	C	D	D
END (state screen)	D	D		
RESUME	B	B	E	E
Power failure handling Power off with monitoring state	E	E	E	E
Power failure handling Power off without monitoring state	E	E	E	E
START (COLD)	D	D	D	E
CONTROLLED START	D	D	D	D
Other operation	E	E	E	E

Meanings of symbols

A : State monitoring is started. / B : State monitoring is restarted if it is stopped.

C : State monitoring is stopped. / D : State monitoring is deleted. (Cannot be restarted)

E : The state of state monitoring does not change due to the operation.

*1 : Monitoring continues, but the action program will pause even if the conditions are satisfied.

Blank : The operation is impossible.

Operation-by-operation description

Operation	State
MONITOR instruction	When a monitor start instruction in the program is executed, monitoring with the specified program monitor starts.
RESTART (state screen)	When function key RESTART is pressed on the Program monitor screen of the state screen, monitoring with the program monitor specified with the cursor restarts.
START (state screen)	When function key START is pressed on the System monitor screen of the state screen, monitoring with the system monitor specified with the cursor starts.
Program Stop	When the temporary stop key is pressed or if the program stops temporarily due to the occurrence of an alarm, state monitoring with the program monitor previously started by the temporarily stopped program stops, if the program monitor is set to 1.
Program End/Enforced End	When the program terminates due to program termination, forced termination, or the occurrence of an alarm, the program monitor previously started by the terminated program is deleted. The deleted program monitor does not start unless a monitor start instruction is executed.
MONITOR END	When a monitor stop instruction in the program is executed, the specified program monitor is terminated. The terminated program monitor does not start unless a monitor start instruction is executed.
PAUSE (state screen)	When function key PAUSE is pressed on the Program monitor screen of the state screen, monitoring with the program monitor specified with the cursor stops. The stopped monitor restarts when the "Restart" key is pressed or the program restarts. When function key PAUSE is pressed on the System monitor screen of the state screen, monitoring with the system monitor specified with the cursor stops.
END (state screen)	When function key END is pressed on the Program monitor screen of the state screen, the program monitor specified with the cursor stops.
RESUME	When the temporarily stopped program restarts, the stopped program monitor restarts.

Operation	State
Power failure handling	If power failure handling is enabled and the monitor is monitoring, the following occurs when the power is turned OFF/ON.
Cold start	If power failure handling is disabled and the power is turned OFF/ON, all monitors terminate except the system monitor of setting 2. The system monitor of setting 2 maintains the state assumed before the power was removed.
Other	For operations other than the above, the monitor state is preserved.

Instruction statements

State monitoring is performed in the section enclosed by the following instructions:

- **MONITOR** <conditional-program-name>
Monitoring starts under the condition described in the condition program.
- **MONITOR END** <conditional-program-name>
Monitoring performed under the condition described in the condition program stops.

Condition program

The monitoring condition program, which has the subtype called WHEN, can specify condition instructions only.

- **WHEN** <conditional-expression>, **CALL** <program-name>

The following conditions can be used:

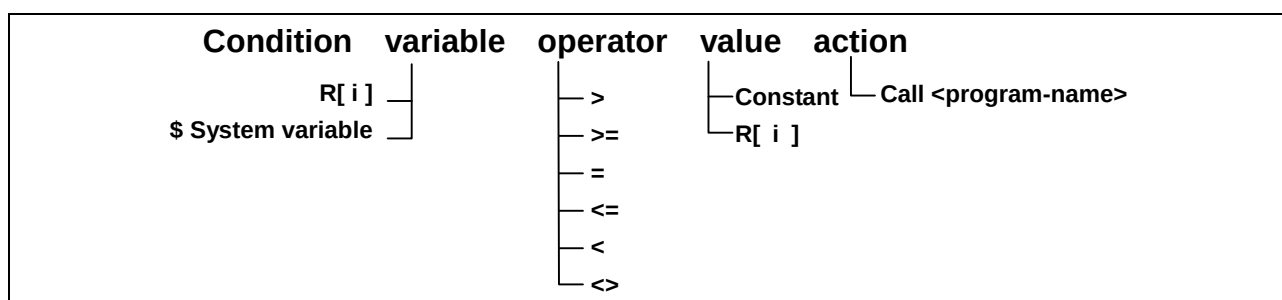


Fig. 9.10 (a) Register/System Variable Condition Compare Instruction

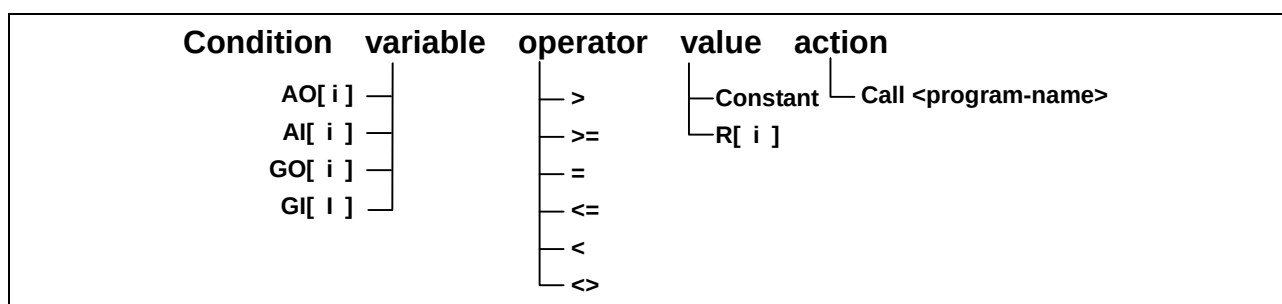


Fig. 9.10 (b) I/O Condition Compare Instruction 1

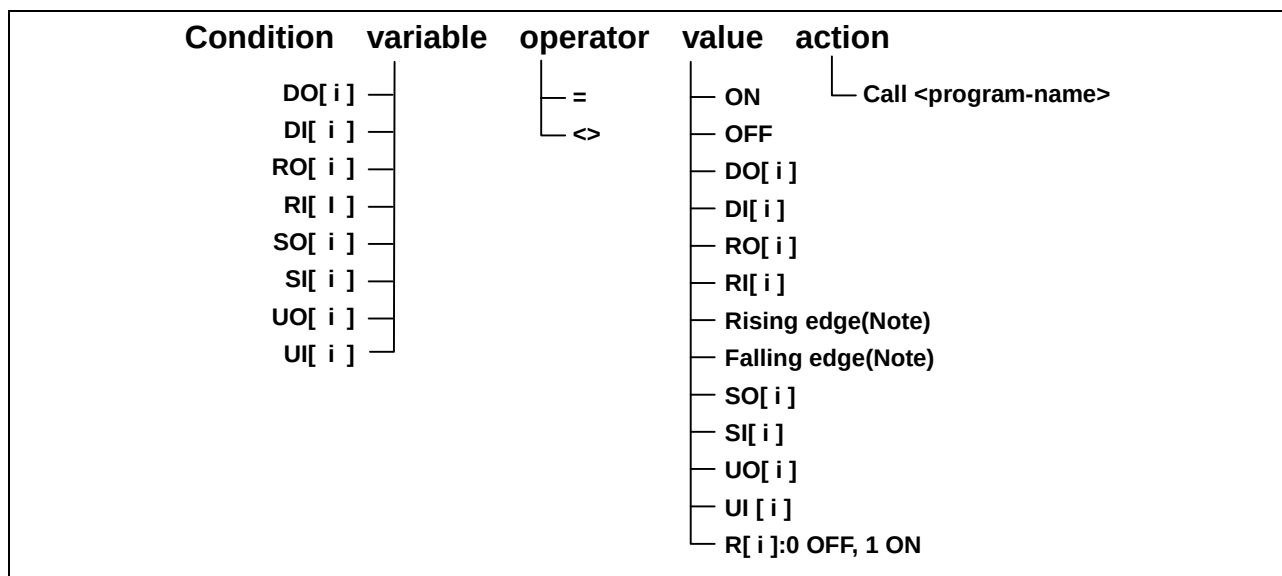


Fig. 9.10 (c) I/O Condition Wait Instruction 2

NOTE

Falling edge:

The falling edge of a signal is regarded as being a detection condition. The condition is not satisfied when the signal remains off. The detection condition is satisfied when the signal changes from the on state to the off state.

Rising edge:

The rising edge of a signal is regarded to be a detection condition. The condition is not satisfied when the signal remains on. The detection condition is satisfied when the signal changes from the off to the on state.

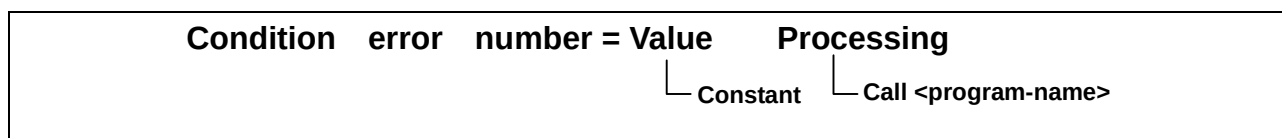


Fig. 9.10 (d) Error Condition Compare Instruction

NOTE

An error number is specified with an alarm ID followed by an alarm number.

Error number = aabbb

where aa = alarm ID

bbb = alarm number

For an explanation of alarm IDs and numbers, refer to the Alarm Code List Operator's Manual.

(Example)

For SRVO006 Hand broken, the servo alarm ID is 11, and the alarm number is 006. Thus,

Error number = 11006

In the condition compare instruction, multiple conditions can be specified on a single line in the condition statement, using the logical operators ("and" and "or"). This simplifies the program structure, allowing the conditions to be evaluated efficiently.

Instruction format

- Logical product (and)
WHEN <Cond.1> AND <Cond.2>, CALL <PRG Name>
- Logical sum (or)
WHEN <Cond.1> OR <Cond.2>, CALL <PRG Name>

If the “and” (logical product) and “or” (logical sum) operators are used in combination, the logic becomes complex, impairing the readability of the program and the ease of editing. For this reason, this function prohibits the combined use of the “and” and “or” logical operators.

If multiple “and” (logical product) or “or” (logical sum) operators are specified for an instruction on a single line, and one of the operators is changed from “and” to “or” or from “or” to “and,” all other “and” or “or” operators are changed accordingly, and the following message appears:

TPIF-062 AND operator was replaced to OR
TPIF-063 OR operator was replaced to AND

Up to five conditions can be combined with “and” or “or” operators on a single line.

(Example) **WHEN <Cond.1> AND <Cond.2> AND <Cond.3> AND <Cond.4> AND <Cond.5>,
CALL <PRG Name>**

Specification

Step

- 1 Enter a condition program name.
On the program list screen, press F2 CREATE and enter a program name.
- 2 Select Cond as the subtype.
Press F2 DETAIL to move to the program details screen.
Position the cursor to the subtype item and press F4 CHOICE.
Select Cond from the subwindow.

NOTE

At this time, the operation group is automatically set as [*,*,*,*,*,*].
A condition program requires no operation group.

State monitoring screen

The state of state monitoring can be monitored using the program monitor screen and the system monitor screen.

Program monitor screen

For the program monitor currently being executed or stopped, the name and state (under execution, stopped) of the condition program is displayed, as well as the name of the parent program(*1) of the program that started the program monitor.

NOTE

If program “A” calls program “B” with a subprogram call, and program “B” executes a monitor start instruction, the name of the parent program, “A,” is displayed in the program name column.

Program monitor		JOINT 10%	
CH	Prog.	Status	Program
1	WRK FALL	Running	Sample
2	WLD TIME	Paused	Sample
3	NO WRK L	Paused	Sample 2
TYPE		SYSTEM	RESTART PAUSE END

Table 9.10 (a) Items and Function Keys on the Program Monitor Screen

Item	Description
CH Prog.	Condition program name
Status	State of the program, either being executed or stopped
Program	Name of the parent program of the program that started the program monitor
F2 SYSTEM	Switches the screen to the system monitor screen. If the system monitor is disabled (\$TPP_MON.\$GLOBAL_MT=0), this key is not effective.
F3 RESTART	When pressed, this key restarts the stopped monitor.
F4 PAUSE	Stops the monitor.
F5 END	Terminates the monitor. The terminated monitor is cleared from the screen.

System monitor screen

All condition programs are displayed. System monitors can be started and stopped.

System monitor		JOINT 10%	
CH	Prog.	Status	
1	WORKDROP	Running	
2	HANDCHCK		
3	NO WRK L		
4	VERFY HND		
[TYPE]		PROGRAM	START END

Table 9.10 (b) Items and Function Keys on the System Monitor Screen

Item	Description
CH.Prog.	Condition program name
Status	State of the program, either being executed or not started (blank)
F2 PROGRAM	Switches the screen to the program monitor screen.
F3 START	Starts the system monitor.
F5 END	Stops the monitor. In the "State" column, a blank is displayed for the stopped monitor.

Notes/restrictions

If multiple condition instructions are specified in a condition program, multiple monitors are started at the same time.

```
1:WHEN (conditional-expression1), CALL (Program name1)
2:WHEN (conditional-expression2), CALL (Program name2)
3:WHEN (conditional-expression3), CALL (Program name3)
```

If, before one monitor start instruction terminates, another monitor start instruction is executed, both monitors are executed at the same time.

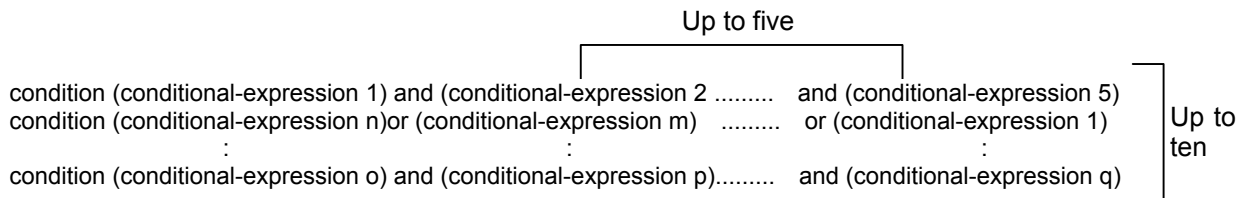
If the condition program names specified in the monitor start instructions are the same, the first condition program is overwritten by the second.

The program monitor stops state monitoring under the following conditions:

- The MONITOR END instruction is executed.
- The program terminates.
- The program stops temporarily. (State monitoring restarts when the program restarts.)

Up to ten conditions can be monitored at the same time.

Up to five “and” or “or” operators can be specified in a single monitoring condition instruction.



While the program is being executed or while it is stopped, the condition statements (condition program) cannot be edited.

In the action program for a system monitor and a program monitor, an operation group cannot be specified.

In the action program, the operation group must be specified as [*,*,*,*,*,*,*].

If the condition is satisfied, the condition program enters the END state. If condition monitoring is to continue, specify a monitor start instruction in the program. Clear the monitoring condition beforehand.

Example

```

MON1. TP
1:WHEN R[1]=1 CALL ACT1
ACT1.TP
1:R[1]=0          <----- Clear the condition
2:
3:(Action)
:
9:MONITOR MON1    <----- Start the monitoring condition again
  
```

If there is no line on which the condition on line 1 is dropped, the condition is immediately satisfied on the monitor start instruction on line 9, causing a MEMO-065 error.

The condition program cannot be executed directly.

9.11 AUTOMATIC ERROR RECOVERY FUNCTION

9.11.1 Overview

This section consists of the following items:

- 9.11.2 Outline of the automatic error recovery function
- 9.11.3 Defining a resume program
- 9.11.4 Teaching the RETURN_PATH_DSBL instruction
- 9.11.5 Setting screen of the automatic error recovery function
 - Enabling/disabling the automatic error recovery function
 - Defining alarm codes to be monitored
 - Defining the recovery switch DI
 - Defining the error recovery information DO (indicating the conditions for executing the resume program)
 - Enabling/disabling the alarm-time automatic start feature
 - Setting the maximum number of automatic start repetitions
 - Defining the automatic start count register
 - Defining automatic error recovery alarm conditions
- 9.11.6 Flowchart for resuming a suspended program

- 9.11.7 Manual operation screen of the automatic error recovery function
 - 9.11.8 Execution of the resume program from the teach pendant and test mode
 - 9.11.9 Changing conditions for executing the resume program
 - 9.11.10 Other specifications and restrictions
 - 9.11.11 Warnings (Be sure to read this section for safety.)
- This function is an optional function.

9.11.2 Outline of the Automatic Error Recovery Function

Background

Robots are sometimes stopped by various alarms even during production. If a robot is stopped, it is necessary to perform recovery operation then resume the program that was originally running. For example, suppose that a robot is performing arc welding. An alarm due to an arc start failure may be issued, stopping the robot. In such a case, the operator must jog the robot to a safe position to, for example, cut the end of the wire or clean the nozzle, then resume the original program. The automatic error recovery function is provided to support automatic operation of the above sequence.

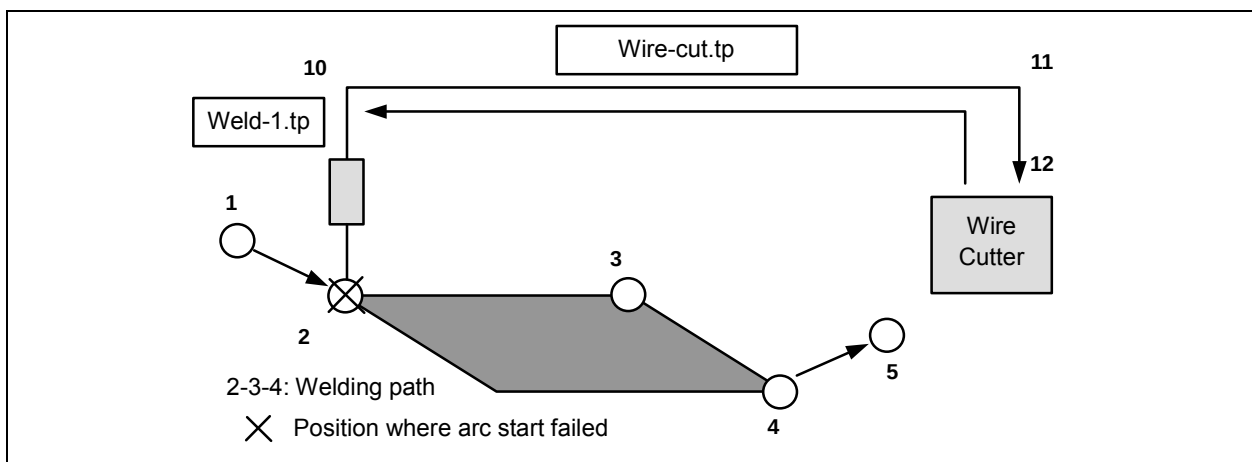


Fig. 9.11.2 Example:

Alarm code monitoring function

In the example shown above, the robot is operated by executing Weld-1.TP to perform welding along the path from 2 to 3 to 4. Assume that an arc start failure occurs at the arc start position 2. With the automatic error recovery function, another program called the resume program, which is Wire-Cut.TP in this case, can be started at the next start signal input. After this program terminates, another start signal input resumes the original program. If the resume operation function is then enabled (which is set on the welding system setting screen), the robot automatically returns to the original position where the robot was stopped, then the original program is resumed. If the return distance for resume operation is set, the robot returns from the stop position by the set distance, then the original program is resumed. If no arc is produced, a scratch start takes place.

In the above example, the automatic error recovery function operates only when an arc start miss alarm is issued. The alarm to be monitored may be changed, or the number of alarms to be monitored may be increased. For example, an arc off alarm can be added as an alarm to be monitored. Then, when an arc off alarm is issued, the same operation sequence as explained above can be performed automatically.

The standard maximum number of monitored alarm codes that can be defined is ten. If no alarm code is defined, the alarm code monitoring function is disabled. In this case, before the suspended original program (Weld_1.TP in the example) is resumed, the resume program is always executed.

Recovery switch DI function

With the recovery switch DI function, whether to start the resume program or not can be selected at the time of start input according to the defined DI status. If the recovery switch DI is off, the original program is resumed without executing the resume program. When this DI is not defined, this function is disabled.

Error recovery information DO function

With the error recovery information DO function, whether the next start input resumes the original program or executes the resume program can be indicated. If the error recovery information DO is on at start input, the resume program is executed. When this DO is not defined, this function is disabled.

Alarm-time automatic start feature

When an alarm code is defined as explained before, and the defined alarm is issued, the program outputs an alarm signal and stops running. Input of the start signal executes the defined resume program. After the resume program terminates, another start signal input restarts the suspended original program.

When the alarm-time automatic start feature is enabled, and a defined alarm is issued, the resume program is automatically executed without outputting the alarm signal and stopping the robot. When the resume program has terminated, the original program is resumed automatically. When this feature is enabled, therefore, the input of the start signal is no longer needed.

Because the alarm signal is not output, other robots are not stopped when multiple robots are operating. The robot for which the alarm was issued moves by itself to the recovery station, and after recovery work, the original program is resumed.



CAUTION

Basically, the automatic error recovery function is designed so that it functions when the teach pendant is disabled. When the teach pendant is enabled, the automatic error recovery function does not function unless the manual test mode is set on the automatic error recovery manual operation screen. For manual testing, see "Execution of the resume program from the teach pendant and test mode."

DI alarm function

By inputting a defined digital input signal, an automatic error recovery alarm can be issued. When this alarm is defined for the alarm-time automatic start feature, the resume program can be executed automatically by inputting the digital input signal.

As the message for an automatic error recovery alarm, a message defined for a user alarm can be used. The alarm severity can be set to either LOCAL or GLOBAL selectively. When LOCAL is selected, the alarm is issued only for a program that defines the resume program. The status of a digital input signal to be monitored can be set by selecting the signal type from among DI, RI, and WI, changing the signal number, and selecting the trigger status between on and off.

Function for disabling the resume operation function after execution of the resume program

In arc tool systems, the resume operation function is generally enabled. With this function enabled, return to the original stop position is always performed then arc is produced when the original program resumes after the resume program terminates.

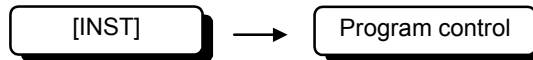
In some systems, however, return to the original stop position should not sometimes be performed. For example, when the nozzle touch state is input through DI, a resume program is used to relieve the torch slightly in the torch direction. If the resume operation function operates, return to the original stop position is performed even when relieve operation has been performed. As a result, the nozzle touch state is observed again. In such a case, the resume operation function needs to be kept enabled, but it should be disabled only after the execution of the resume program.

This can be performed with the RETURN_PATH_DSBL instruction. By using this instruction within the resume program, the resume operation function can be disabled only when the original program is resumed

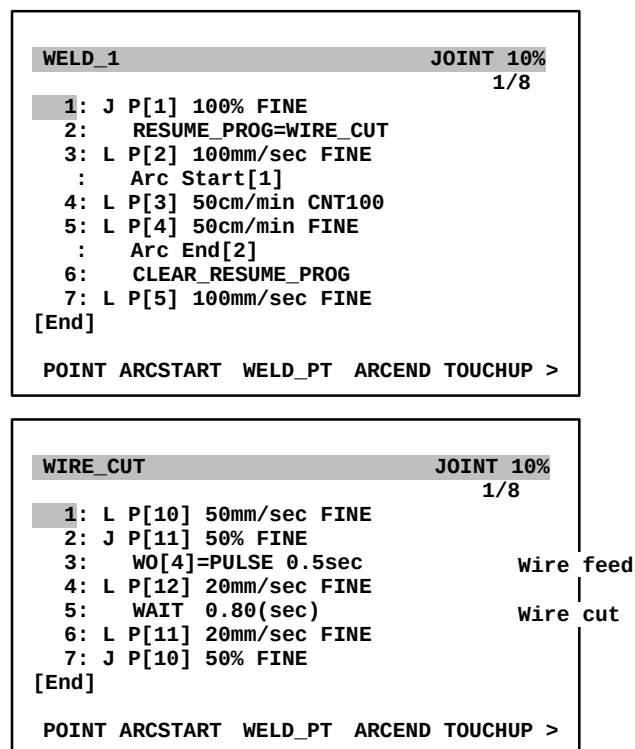
next. This instruction is valid only when it is executed within a resume program; the instruction is invalid when executed in a program other than the resume program.

9.11.3 Defining a Resume Program

The automatic error recovery function executes a resume program defined in an original program, in lieu of the original program. To define a resume program, use the `RESUME_PROG=` instruction. To erase the defined resume program, use the `CLEAR_RESUME_PROG` instruction. These instructions can be displayed on the edit screen by following the procedure shown below.



In the example given in Fig. 9.11.2, the following programs are used:



In the above program example, the `WIRE_CUT` program is taught in the second line of the `WELD-1` program and is erased in the sixth line. Since the `WIRE_CUT` program is defined as the resume program between the third to seventh lines, it is executed as the resume program. In the seventh and subsequent lines, the resume program has been erased, so the resume program is not executed.

The resume program is erased also when:

- Backward execution is performed.
- The cursor line is changed manually.
- The program terminates.

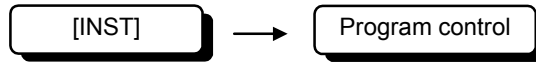


CAUTION

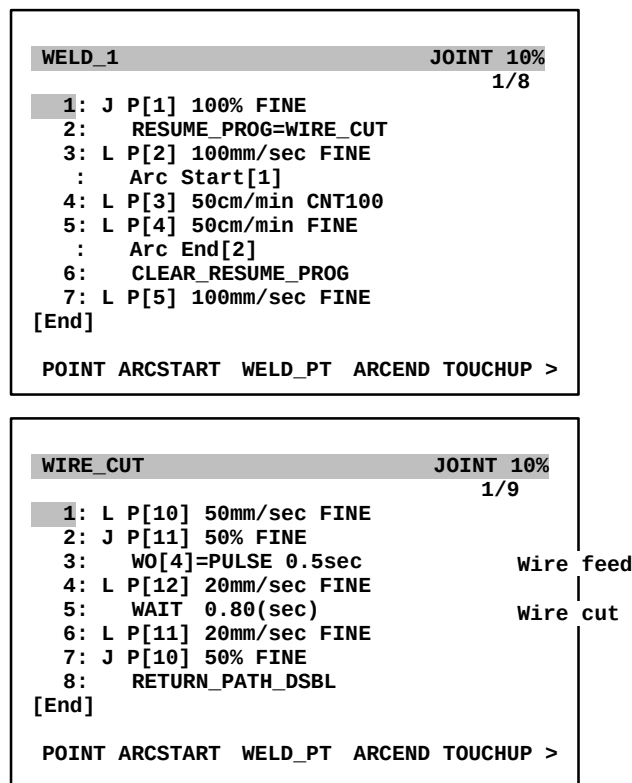
When the `RESUME_PROG` instruction is executed within the resume program, it is defined as a resume program for the original program.

9.11.4 Teaching the RETURN_PATH_DSBL Instruction

The RETURN_PATH_DSBL instruction appears in the menu containing the RESUME_PROG instruction.



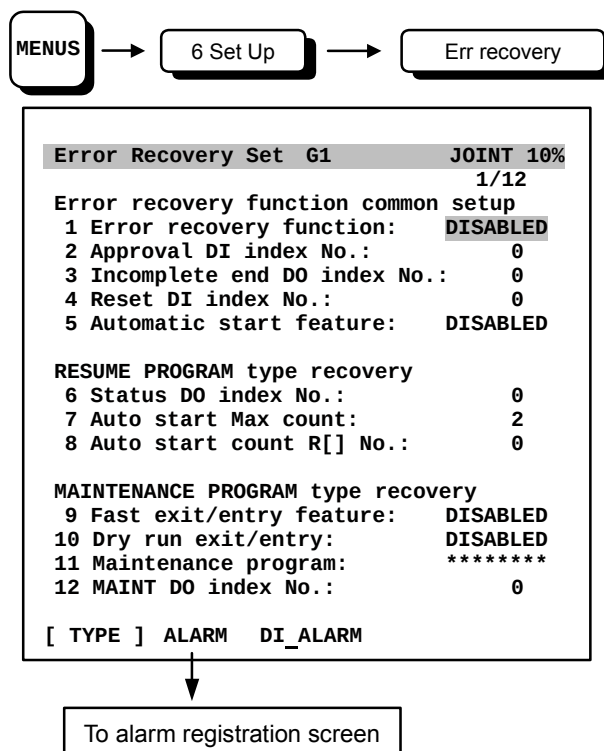
The RETURN_PATH_DSBL instruction is valid only when it is taught within resume program instructions. Use this instruction as shown in the sample program given below. If the instruction is taught as shown below, the resume operation function does not operate when the original program resumes after the resume program terminates, even if the resume operation function is enabled.



9.11.5 Setting the Automatic Error Recovery Function

On the setting screen of the automatic error recovery function, the following settings can be made:

- Enabling/disabling the automatic error recovery function
- Defining alarm codes to be monitored
- Defining the recovery switch DI
- Defining the error recovery information DO (indicating conditions for executing the resume program)
- Enabling/disabling the alarm-time automatic start feature
- Setting the maximum number of automatic start repetitions
- Setting the automatic start count register
- Enabling/disabling the fast exit/entry feature
- Enabling/disabling dry run exit/entry operation
- Defining a maintenance program
- Defining the maintenance DO
- Defining automatic error recovery alarm conditions



Alarm setting screen

Error Recovery Set G1		JOINT 10%
		1/10
1 Monitored alarm code 53013		
2 Monitored alarm code 53018		
3 Monitored alarm code 0		
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		
[TYPE]		DONE HELP

53013 indicates "ARC-013 Arc Start failed."

53018 indicates "ARC-018 Lost arc detect."

For alarm codes, refer to the Alarm Code List Operator's Manual.

Enabling/disabling the automatic error recovery function

This item enables or disables the automatic error recovery function. When the automatic error recovery function is enabled, and neither monitored alarm codes nor the recovery switch DI are defined, the resume program is always executed at restart from the suspended state (except when the error recovery information DO is off). When this item is disabled, the resume program is not executed.

Defining alarm codes to be monitored

To define alarm codes to be monitored, press the F2 (ALARM) key. A screen for defining alarm codes is displayed.

When a defined alarm code is issued, and a program is suspended, the resume program is executed at restart. Each alarm code consists of an alarm code ID and alarm number. The alarm code ID indicates the type of alarm. For an arc start failure alarm, for example, the following alarm code is indicated:

ARC - 013 Arc Start failed = 5 3 0 1 0

ID (53) Number ID Number

For alarm numbers, refer to the Alarm Code List Operator's Manual.

Up to ten alarm codes can be defined as standard. To change the maximum number of alarm codes (up to 20 codes) that can be defined, change system variable \$RSMPRG_SV.\$NUM_ALARM, turn the power off then on.

Pressing the F5 (HELP) key displays the following screen:

```

Error Recovery Set G1      JOINT 10%
HELP      Arrows to scroll, PREV to exit

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,

```



CAUTION

Do not define any warning alarm as an alarm code.

Even when a defined alarm is issued, the resume program is not executed if the recovery switch DI is off. When there is no alarm code defined, that is, when all defined values are 0, the alarm code monitoring function is disabled.

The specifications of the alarm code monitoring function are listed below.

Table 9.11.5 (a) Specifications of the Alarm Code Monitoring Function

Alarm code definition	Alarm code function status	Issuance of defined alarm	Execution of resume program at restart
All 0s	Disabled	—	Executed
At least one alarm code is defined	Enabled	Issued	Executed
		Not issued	Not executed

Defining the recovery switch DI

To use the recovery switch DI function, define an DI number. After the number is defined, power must be turned off then back on.

With this function, the operator can choose whether to execute the resume program or not at the time of restart from the suspended state by using a peripheral device. If this signal number is not defined, this function is disabled.

The specifications of the recovery switch DI function are listed below.

Table 9.11.5 (b) Specifications of the Recovery Switch DI Function

DI number definition	Recovery switch DI function status	DI status	Execution of resume program at restart
0	Disabled	—	Executed
Valid number defined	Enabled	On	Executed
		Off	Not executed

**CAUTION**

To continue a resume program at program restart after the resume program is suspended, input the on state of the recovery switch DI. If it is off, the original program is executed.

Defining the error recovery information DO (conditions for executing the resume program)

When the alarm code monitoring function and recovery switch DI function are both disabled, the resume program is always executed at the time of restart after the original program is suspended. When both the functions are enabled, it is difficult to determine whether the original program or resume program is to be executed at restart.

The error recovery information DO is on only when the resume program is executed at restart. When the signal is off, the original program is executed at restart. With this function, the operator can know which program is to be executed next.

If the following conditions are met, the error recovery information DO goes on:

- The automatic error recovery function is enabled.
- The program to be executed is not in single step mode.
The single step LED on the teach pendant indicates the single step status of a program currently selected (more precisely, the program set in \$TP_DEFPROG). When the resume program is suspended, the error recovery information DO is on even if the single step LED on the teach pendant lights. This is because the resume program to be executed is not in single step mode.
- The resume program is defined in the currently selected program (original program).
- The currently selected program (original program) has a motion group.
- The currently selected program (original program) is suspended, and the resume program is not yet completed.
- There is no optional function that disables the automatic error recovery function. See “Other specifications and restrictions.”
- The user condition parameter (\$AUTORCV_ENB) is true. See “Conditions for executing the resume program.”
- When the teach pendant is enabled:
 - The operation mode (on the automatic error recovery manual operation screen) is TP_TEST.
- When the teach pendant is disabled:
 - The operation mode (on the automatic error recovery manual operation screen) is AUTO.
 - The remote conditions are met when system variable \$RMT_MASTER is 0.
 - There is no alarm code defined. If any alarm code is defined, the alarm code is issued.
 - The recovery switch DI function is disabled. If this function is enabled, the recovery switch DI signal is on.

**CAUTION**

- The selected program means the program to be executed by inputting the start signal.
- While the resume program is being executed, single step operation cannot be performed.
- Even if the error recovery information DO is on, the resume program is not executed when backward execution of the original program is performed.
- Backward execution in the resume program is possible.
- The update cycle period for the error recovery information DO is 300 ms. When the conditions listed above have been changed, wait 300 ms before program execution.

The timing chart for the error recovery information DO status and start signal is shown below.

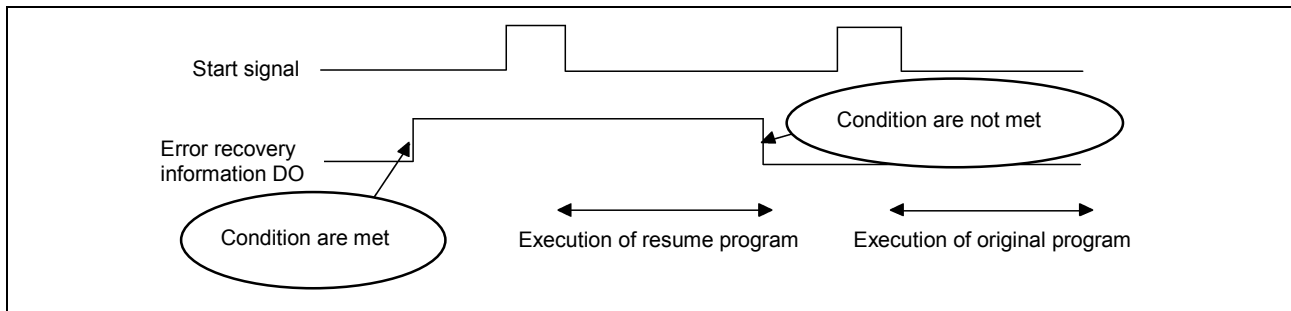


Fig. 9.11.5 (a) Error Recovery Information DO Output Timing Chart

Defining the incomplete end DO

When an incomplete end DO number is defined, the incomplete end DO is output if a certain forced termination alarm is issued during execution of the resume program. The output incomplete end DO is turned off by the next start signal input.

Before inputting the start signal, the operator must check the incomplete end DO signal status. If this signal is on, the resume program terminates in the middle, so the robot is not in a specified position. In such a case, inputting the start signal causes the robot to perform resume operation to return from the current position to the stopped position of the original program, which may interfere with obstacles such as a jig. Therefore, before inputting the start signal, check the current robot position. If an interfering object exists, jog the robot to a position near the stopped position of the original program, then input the start signal.

This signal may be added to the PLC start signal acceptance conditions.

If this signal is set to 0, this function is disabled.

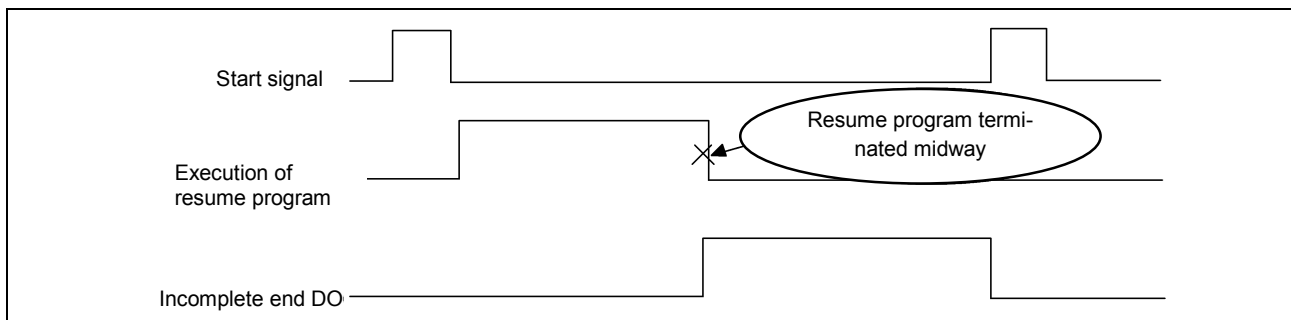


Fig. 9.11.5 (b) Incomplete End DO Output Timing Chart

Defining the incomplete-end reset DI

When the incomplete end DO is included in the PLC start signal acceptance conditions, the operator requires a means to turn off the incomplete end DO externally. Inputting the incomplete-end reset DI signal turns off the incomplete end DO. When the incomplete end DO is output, the operator must first perform appropriate operation such as jogging the robot to near the stopped position of the original program, input the incomplete-end reset DI, then input the start signal.

When this signal is set to 0, this function is disabled.

Enabling/disabling the alarm-time automatic start feature

When this item is enabled, and an alarm code to be monitored is defined, this feature functions if the defined alarm is issued. If a defined alarm is issued, this feature automatically executes the resume program. In this case, the alarm signal is not output. When the execution of the resume program has terminated, the original program is resumed automatically. During then, the operator need not input the start signal. Since the alarm signal is not output, other robots operating in the same line are not stopped.

⚠ CAUTION

- 1 Defined alarms must have the suspension alarm attribute.
- 2 If the error recovery information DO is off when the resume program is executed automatically, alarm "INTP-135 Recovery DO OFF in auto start mode" is issued.
- 3 While the resume program is being executed, the UOP PAUSED signal is output. This is because the original program is in the suspended state. This specification is the same as that for multitasking systems.

⚠ WARNING

- 1 The alarm-time automatic start feature works on a program selected on the teach pendant. For example, suppose that program A having a resume program instruction has been executed from the teach pendant, then program B without a resume program instruction is selected and executed from the teach pendant. If an alarm defined in program A is then issued, the alarm signal is not output, but the resume program is not executed automatically. The reason for this is that the automatic start feature works on a selected program. In this example, program B that is currently selected does not define any resume program.
- 2 When the automatic start feature item is enabled, and no monitored alarm code is defined, inputting the start signal for executing the resume program automatically executes the resume program then resumes the original program. In other words, when the start signal is input while the original program is suspended, the resume program is executed. As the resume program has terminated, the original program is then restarted automatically.

Setting the maximum number of automatic start repetitions

When a defined alarm is issued, the alarm-time automatic start feature automatically executes the resume program, then resumes the original program. If the defined alarm is issued again when the original program is resumed, the automatic start feature functions again. For example, the automatic start feature is activated by an alarm indicating an arc start failure, then the same alarm is issued again when the original program has resumed.

To prevent such an endlessly repeated condition, set the maximum number of automatic start repetitions. The number of times the resume program is started repeatedly is counted internally. If the count exceeds the set value, "INTP-134 Over automatic start Max counter" is issued, and the error recovery information DO is turned off at the same time. If this occurs, eliminate the cause of the alarm issued in the original program. Then input the start signal.

⚠ CAUTION

The number of repetitions counted internally is cleared when the execution of a move statement has terminated and when the CLEAR_RESUME_PROG instruction has been executed.

Defining the automatic start count register

As mentioned above, the resume program may be executed several times repeatedly by the automatic start feature. When the automatic start count register is defined, a different program can be executed as the resume program each time the resume program is executed. For example, when the resume program is executed for the first time by the automatic start feature, the register value is 1. When an alarm is issued again during execution of the original program, and the resume program is then executed again by the automatic start feature, the register value is 2. By executing a different subprogram in the resume program according to the register value, different resume program operation can be performed each time the repetition count is incremented.

**CAUTION**

When the resume program is executed by other than the automatic start feature, the register value is 0. Therefore, a resume program must be created so that the same subprogram is called when the register value is 0 and when the value is 1.

Enabling/disabling the fast exit/entry feature

If an alarm is issued during operation in a complicated environment, the robot moves from the stopped position to the taught point to execute the resume program. In this case, the robot may interfere with part of a workpiece or peripheral devices. After recovery operation, similar interference may occur when an attempt is made to execute the original program. The fast exit/entry feature is provided to avoid the possibility of such interference. The feature can be enabled or disabled by setting this item.

The fast exit/entry feature causes the following operation automatically:

- 1 From the stopped position, disable arc welding, and execute only the move statements of the original program up to the end.
- 2 Execute a maintenance program.
- 3 Disable arc welding, execute the move statements of the original program from the beginning to move the robot to the stopped position.
- 4 Enable arc welding, and resume the original program operation.

Even when this feature is enabled, the resume program takes priority over this feature if the resume program is enabled in the original program. In other cases, the maintenance program is executed.

Enabling/disabling dry run exit/entry

In the fast exit/entry feature, this item specifies whether exit from the stopped position and return to the stopped position after maintenance program execution are to be performed at dry run speed.

Defining a maintenance program

Define the name of a maintenance program used as the standard maintenance program. The maintenance program name can also be specified using the maintenance program instruction on the edit screen.

Defining the maintenance DO

Define the number of the DO for indicating that the fast exit/entry feature is operating.

Defining automatic error recovery alarm conditions

Define the conditions for issuing an automatic error recovery alarm on the definition screen that is displayed by pressing F3 (DI_ALARM) on the setting screen of the automatic error recovery function.

Error Recovery Set G1				JOINT 10%
	UALM	Severity	Type	1/3 Value
1	[1]	LOCAL	DI[1]	ON
2	[5]	GLOBAL	RI[2]	OFF
3	[10]	LOCAL	WI[5]	ON
PLEASE POWER OFF AFTER CHANGING				DI/DO
[TYPE]				DONE HELP

On this screen, the items shown below can be set. The alarm code of the automatic error recovery alarm is 12278.

- User alarm number

When the automatic error recovery alarm is issued, the user alarm message with the set number is displayed as an alarm message. When this item setting has been changed, the new setting becomes effective immediately.

- **Alarm severity**
This item can choose whether the automatic error recovery alarm is a local alarm or global alarm. When LOCAL is set, the automatic error recovery alarm is issued only for the program that defines a resume program. If there is no program that defines a resume program, the alarm is regarded as a global alarm. If the automatic error recovery alarm is issued when there is no program being executed, a warning is generated. When this item setting has been changed, the new setting becomes effective immediately.
- **Signal type**
Choose the type of the digital signal for issuing the automatic error recovery alarm from among DI, RI, and WI. When this item setting has been changed, the power must be turned off then back on for the new setting to become effective.
- **Signal number**
Set the number of the digital signal for issuing the automatic error recovery alarm. When this setting has been changed, the power must be turned off then back on for the new setting to become effective.
- **Detection signal status**
Set the status of the digital signal for issuing the automatic error recovery alarm to ON (high) or OFF (low). When this setting has been changed, the power must be turned off then back on for the new setting to become effective.

The standard number of automatic error recovery alarm conditions is three. This number can be increased to up to five by changing system variable \$RSMPRG_SV.\$NUM_DI_ALM. After this system variable has been changed, the power must be turned off then back on for the new setting to become effective.

9.11.6 Flowchart for Resuming a Suspended Program

The resume program is executed according to the following flowchart:

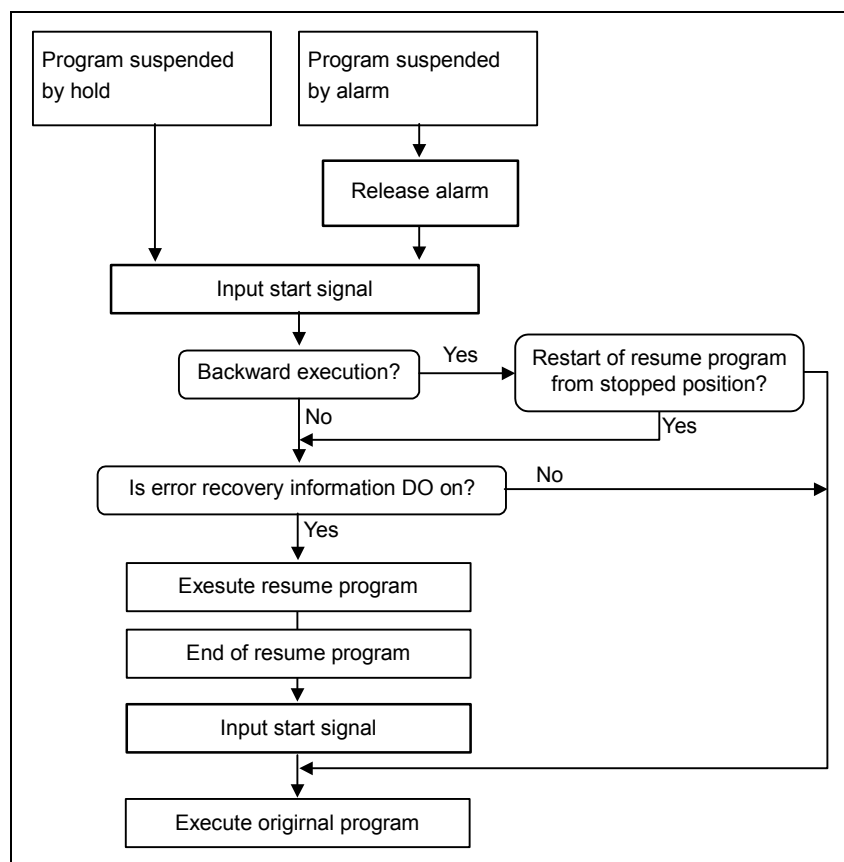


**Fig. 9.11.6 Flowchart for Executing the Resume Program
(When Automatic Start is Disabled)**

⚠ CAUTION

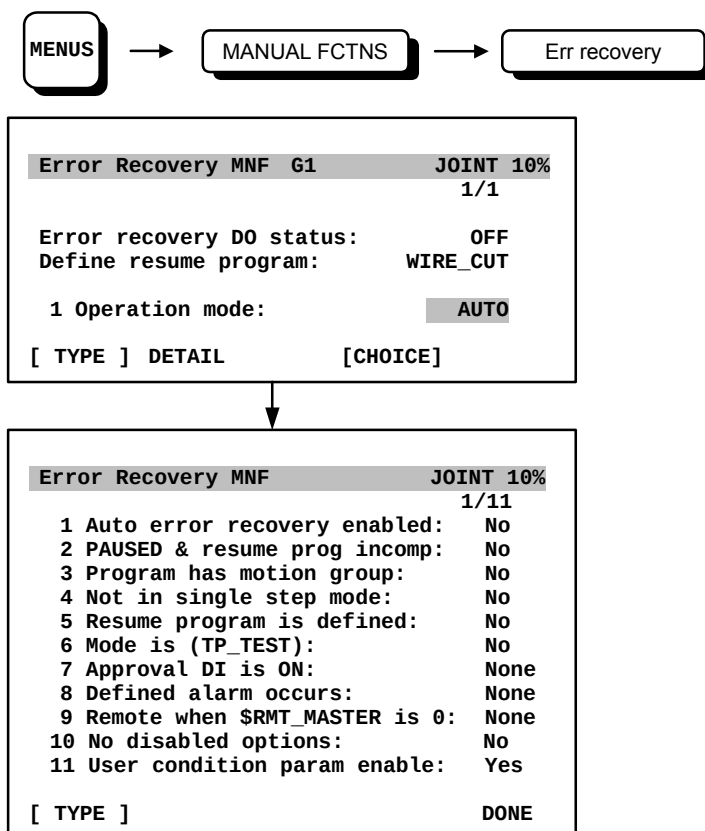
- 1 When forward execution is specified while the original program is suspended, the resume program is executed if the error recovery information DO is on; if it is off, the original program resumes.
- 2 When forward execution is specified while the resume program is suspended, the resume program resumes if the error recovery information DO is on; if it is off, the original program is executed.
- 3 When backward execution is specified while the original program is suspended, backward execution is performed for the original program without executing the resume program.
- 4 When backward execution is specified while the resume program is suspended, backward execution is performed for the resume program.

9.11.7 Manual Operation Screen of the Automatic Error Recovery Function

A manual operation screen is supported for the automatic error recovery function. This screen contains the following:

- Error recovery information DO status
- Name of the resume program defined in the currently selected program
- Operation mode setting
- Detail information about the error recovery information DO

This screen can be selected by following the procedure shown below.



Error recovery information DO status

The error recovery information DO status is indicated. Even when the error recovery DO is not defined, its status can be indicated. From this information, the operator can know which program, the resume program or original program, is to be executed.

Defined resume program

The name of the resume program defined in the currently selected program is indicated. From this information, the operator can check whether a wrong resume program is defined or not.



CAUTION

If a wrong program is defined as the resume program, the robot operation is unpredictable. Therefore, check that the resume program is correct.

Operation mode

There are three operation modes. The standard setting is AUTO. When the display changes from this screen to another, AUTO is automatically set again.

- **AUTO**
This mode should be set when the teach pendant is disabled. When this mode is selected, the resume program is executed according to the status of the alarm code monitoring function and recovery switch DI function. If this mode is selected when the teach pendant is enabled, the resume program is not executed.
- **NO_EXEC**
 When this mode is selected, the error recovery information DO is always off. Therefore, in this mode, the resume program is not executed.
- **TP_TEST**
This mode should be set when the teach pendant is enabled. When this mode is selected, and when the teach pendant is enabled, the resume program is always executed regardless of the status of the alarm code monitoring function or error recovery switch DI function.

Displaying detail conditions of the error recovery information DO

When F2 (DETAIL) is pressed on the manual operation screen of the automatic error recovery function, detail conditions related to the error recovery information DO status are displayed. When all items on the detail screen are set to Yes or None, the error recovery information DO is turned on. When the error recovery information DO is off, and you cannot find the cause of the DO being off, check this screen.

- **Auto error recovery enabled**
 This item indicates whether this function is enabled or disabled on the setting screen of the automatic error recovery function.
- **PAUSED & resume prog incomp**
 This item indicates the following conditions:
 - The selected program must exist.
 - The selected program must be in the suspended state.
 - A resume program must be defined in the selected program, and the execution of the resume program must not have been completed.
- **Program has motion group**
 This item indicates that the selected program has a motion group.
- **Not in single step mode**
 This item indicates that the single step mode is not set.
 The single step LED on the teach pendant indicates the single step status of the selected program (\$TP_DEFPROG). Even when the single step key is pressed while the resume program is suspended,

and the single step LED goes on, the error recovery information DO is held on. This is because the selected program is the original program, and the LED indicates that the original program is in single step mode; the resume program is not in single step mode.

- **Resume program is defined**

This item indicates that a resume program is defined in the selected program.

- **Mode is (xxxx)**

This item indicates that the operation mode is suitable for the current status. For example, when the teach pendant is disabled, "AUTO" is indicated in the portion "xxxx." When the teach pendant is enabled, "TP_TEST" is indicated.

- **Approval DI is ON**

This item indicates the recovery switch DI status. When the DI number is not defined, or when the teach pendant is enabled, "None" is indicated.

- **Defined alarm occurs**

This item indicates that an alarm code is defined, and that alarm is issued. When no alarm code is defined, or when the teach pendant is enabled, "None" is indicated.

- **Remote when \$RMT_MASTER is 0**

This item indicates that remote conditions are met. This function is enabled only when the teach pendant is disabled, system variable \$RMT_MASTER is 0, and system variable \$RSMPRG_SV.\$CHK_REMOTE is true.

- **No disabled options**

There are options that cannot be used together with the automatic error recovery function. This item indicates whether such options are present or not.

- **User condition param enable**

This item indicates the status of system variable \$AUTORCV_ENB for user conditions. For how to use this system variable, see "Changing conditions for executing the resume program."

9.11.8 Execution of the Resume Program from The Teach Pendant and Test Mode

Normally, the automatic error recovery function is used when production is started with the teach pendant disabled. When checking the resume program during teaching, set the operation mode to TP_TEST on the manual operation screen. In TP_TEST mode, the resume program can be executed regardless of the recovery switch DI status and whether a defined alarm is issued or not.

9.11.9 Changing Conditions for Executing the Resume Program

To use resume program execution conditions other than alarm codes, use user condition system variable \$AUTORCV_ENB and the status monitoring function (A05B-2500-J628).

For example, to execute the resume program when R[1] is 1, create the following monitor program, and start MONIT1.CH on the system monitor screen.

MONIT1.CH 1: WHEN R[1]=1,CALL DO_RESUME 2: WHEN R[1]<>1,CALL NO_RESUME	DO_RESUME. TP 1: \$AUTORCV_ENB=1 2: MONITOR MONIT_3
MONIT2.CH 1: WHEN R[1]=1,CALL DO_RESUME	DO_RESUME. TP 1: \$AUTORCV_ENB=0 2: MONITOR MONIT_2
MONIT3.CH 1: WHEN R[1]<>1,CALL NO_RESUME	

The start conditions can be changed by modifying the monitor program. For how to use the status monitoring function, refer to 9.10 STATE MONITORING FUNCTION. In this case, the automatic start function is unavailable.

9.11.10 Other Specifications and Restrictions

- While the resume program is being executed, single step operation is not performed. Single step mode is valid only for the original program.
- When the cursor line is changed and executed while the original program is suspended, the resume program is not executed.
- While the resume program is being executed, the resume program execution status cannot be checked on the program edit screen. On the edit screen, the suspended original program is displayed.
- When a multitasking program (a main program and subprogram) is being executed with the alarm code monitoring function disabled and the recovery switch DI undefined, pressing the hold button causes both the main program and subprogram to stop. Suppose that a resume program is defined in the subprogram, but that no resume program is defined in the main program. In this case, when the main program is selected and re-executed, the resume program for the subprogram is not executed. This is because the selected program is the main program, and the error recovery information DO is off.

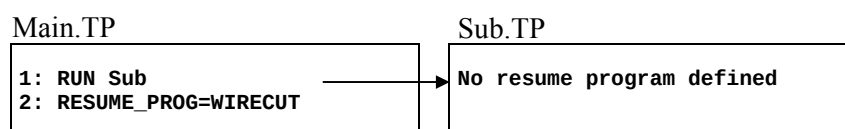
In this case, when the subprogram is selected and executed, the error recovery information DO is turned on, so the resume program is executed.



CAUTION

When using the automatic error recovery function in multitasking systems, define a resume program only in the main program. Even when a resume program is defined in a subprogram, that resume program cannot be executed.

Definitions in the main program and subprogram are shown below.



- While the resume program is being executed, the suspended original program is displayed on the edit screen.
- When the cursor line in the original program is moved while the resume program is suspended, then program re-execution is performed, the resume program is resumed. After the resume program terminates, specifying program execution displays a popup menu confirming the cursor movement. When Yes is entered in response, the original program is executed starting from the new cursor line.
- In a single task system, when the resume program is suspended, selecting a program other than the original program on the program directory screen causes the original program to terminate.

- Never teach the arc and weaving instructions in the resume program. If an arc instruction is executed in the resume program while arc welding is being performed by the original program, alarm “ARC-034 Task does not control welding” is issued. In addition, weaving operation is not performed within the resume program.
- The automatic error recovery function supports the power failure handling function.
- The automatic error recovery function is disabled when one of the following options is loaded:
 - Arc sensor
 - AVC (TIG arc length control)
 - MIG EYE option
 - Root path memorization
 - Line tracking
 - Soft float
 - Continuous turn
 - Coordinated motion
 - Remote TCP
 - Accurate path function
 - Constant joint path function (path not overridden)
 - Multi robot control

9.11.11 Warnings

When using the automatic error recovery function, observe the following safety precautions:

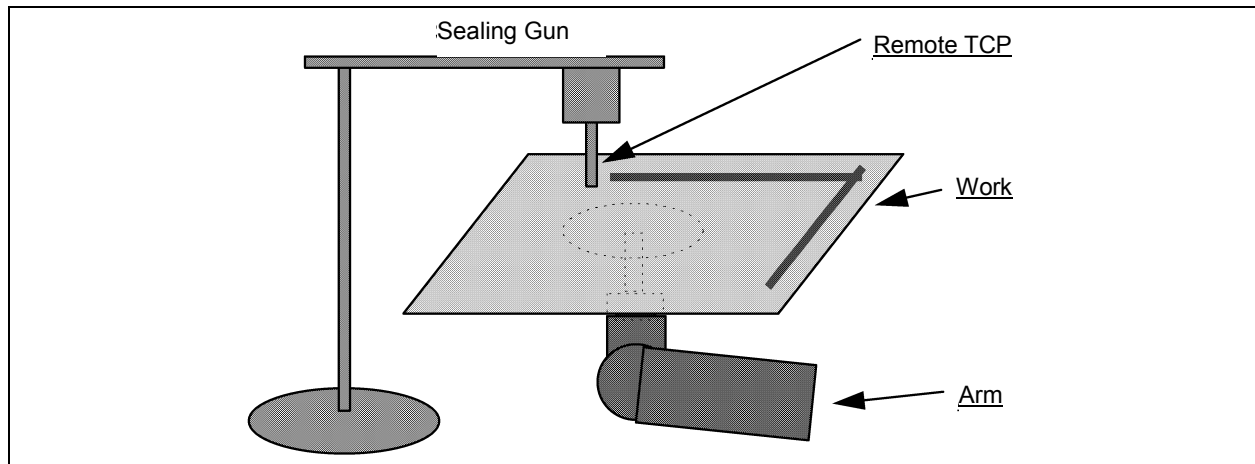
- If a wrong program or a program causing wrong operation is defined as a resume program, the robot moves in a direction the operator cannot predict. Define a correct program.
- Before inputting the start signal and before pressing the execution key on the teach pendant, for safety, check the error recovery information DO status to confirm whether the original program or resume program is to be started.
- If the operation mode is set to TP-TEST on the manual operation screen of the automatic error recovery function, the resume program is started even when a defined alarm is not issued or when the recovery switch DI is off.
- When an operation mode other than AUTO is set on the manual operation screen of the automatic error recovery function, then the display is changed to another screen, the operation mode is set to AUTO again automatically. To use an operation mode other than AUTO, always keep displaying the manual operation screen of the automatic error recovery function.

9.12 REMOTE TCP FUNCTION

9.12.1 Summary

The REMOTE TCP function is used to process the work by moving the work which is placed on the robot hand. Robot can keep the relation between the tool fixed on the ground and work.

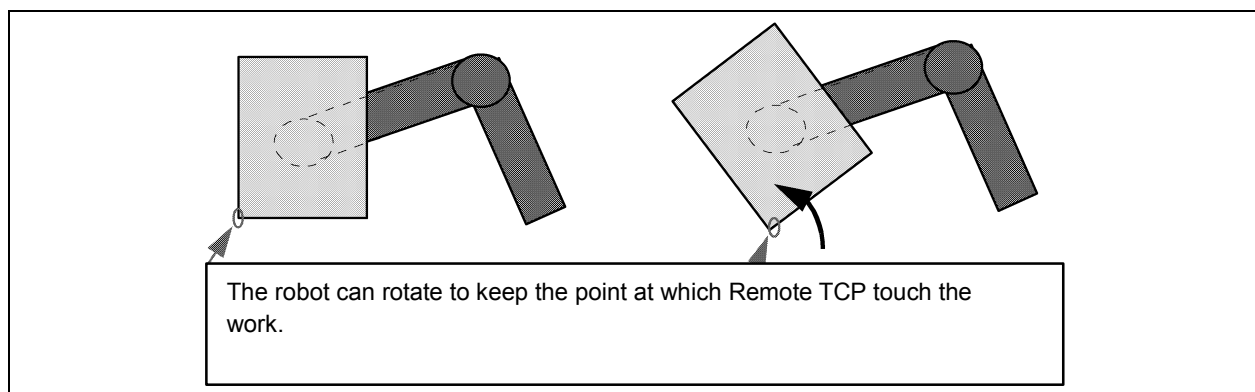
Example of using REMOTE TCP function (by Sealing application)



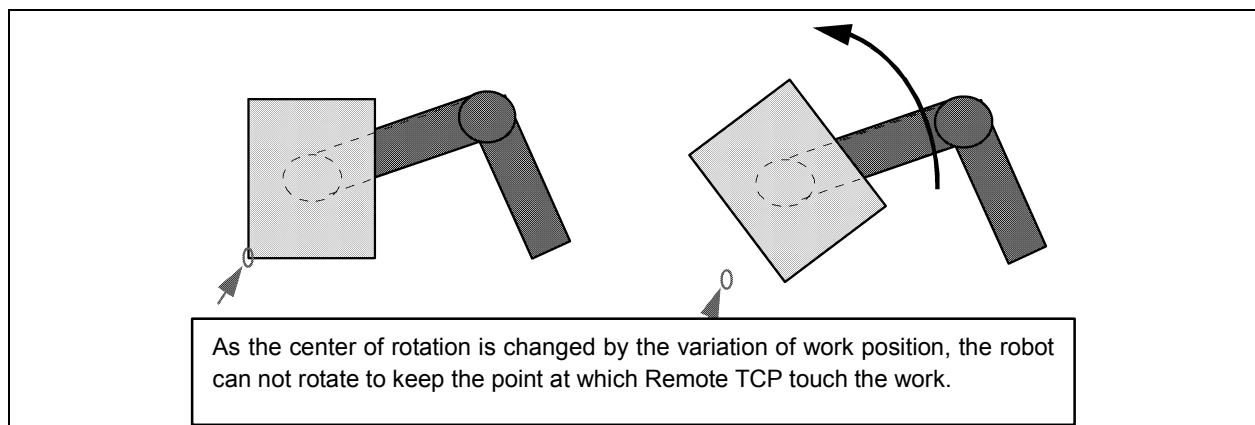
Advantage of Remote TCP

- The gun is fixed on the ground, then the cabling of tool is easy.
- It is not necessary to take the heavy gun.
- Even if you do not use the REMOTE TCP function, you could process the work by moving the work which is placed on the robot hand. But if you rotate the work against the TCP, the tool can not do the coordinate motion against the work.

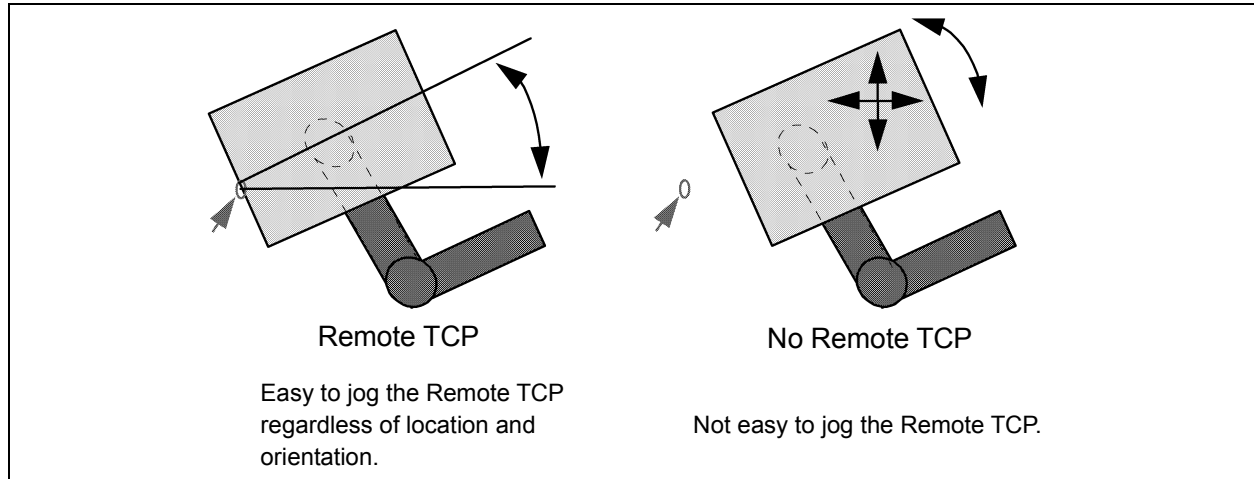
Case of using REMOTE TCP function



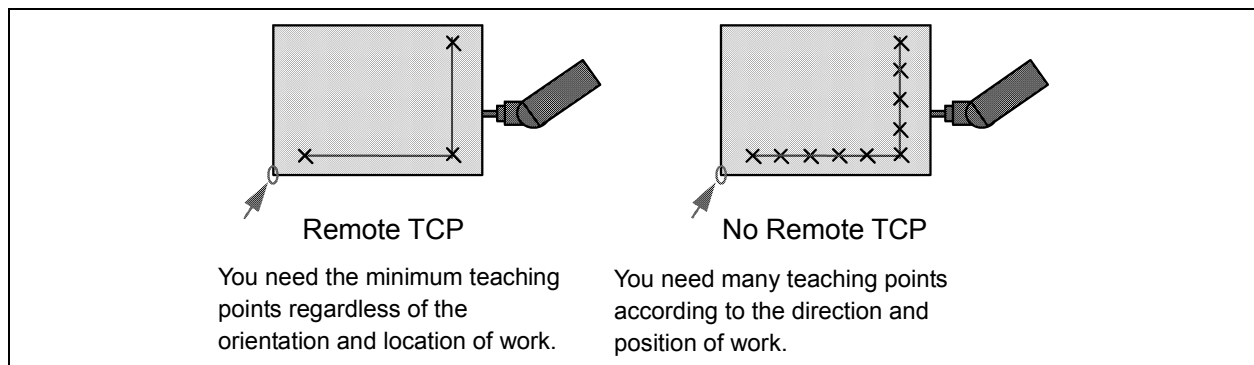
Case of NOT using REMOTE TCP function



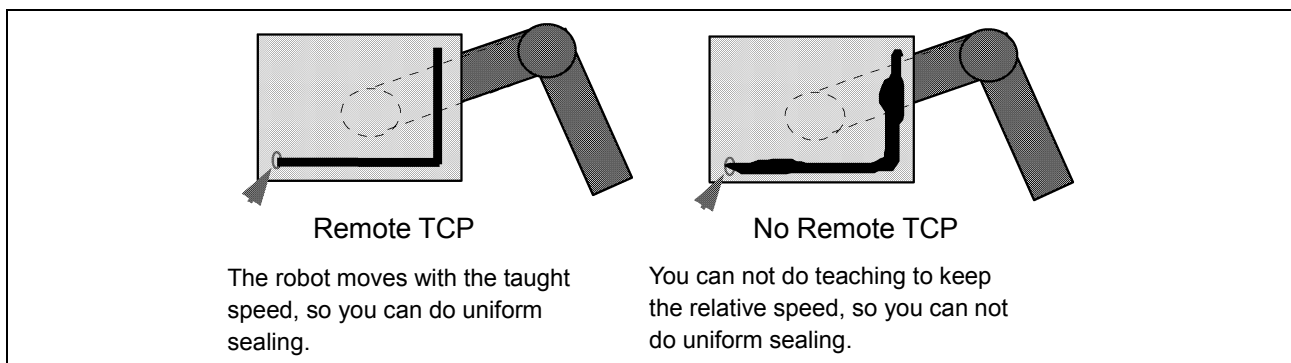
- You can get the easy teaching operation by remote TCP jog function.



- You can reduce teaching point drastically.



- You can do uniform sealing against work.



Required option

- Remote TCP (A05B-2500-J624)

Limitation

The Specification of REMOTE TCP is as follows.

- Support 6-axes robot and M-410i only.
- Support Spot, Sealing, Handling tool only.
- Not support Arc tool function and tracking function.

The limitation of REMOTE TCP is as follows.

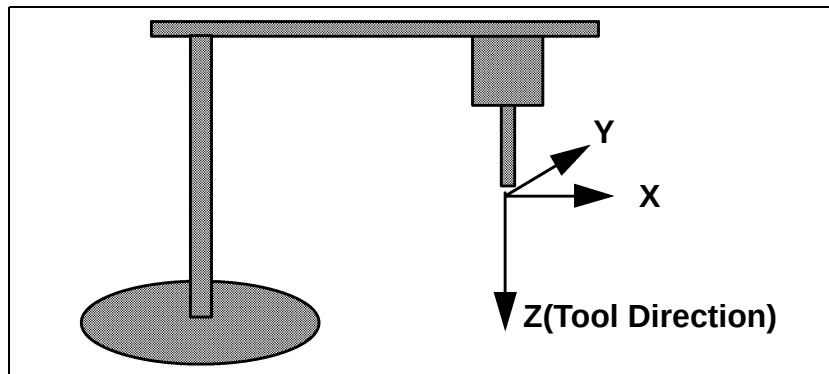
- This function is applied to the linear and circular motion.
- REMOTE TCP supports incremental motion. However, the incremental motion is relative to the user frame.

- REMOTE TCP does not support Wrist joint motion. The motion option of wrist joint REMOTE TCP do not coexist in the one TPE motion instruction.
- When the robot motion changes from REMOTE TCP to normal, you can not use Speed Prediction function.

9.12.2 Setup

Setting up

If you use the REMOTE TCP function, you need to teach the tool center point which is fixed on the ground to the robot.



You can set REMOTE TCP position by using the same method for User frame.

- 1 Press MENUS.
- 2 Select SETUP.
- 3 Press F1[TYPE].
- 4 Select Frames.
- 5 If user frame is not displayed, press F3[OTHER], and select User Frame. If F3,[OTHER], is not displayed, press PREV.
- 6 Move the cursor to the REMOTE TCP frame to use.

SETUP Frames				JOINT 10%	
User/RTCP	Setup/Direct	Entry	1/5		
	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	[	]
[TYPE] DETAIL [OTHER] CLEAR SETIND >					

- 7 Press F2,[DETAIL].
- 8 Select the method from Direct Entry, Three Point or Four Point, and set the frame by using the same method for User frame.

Remote TCP jog

When you do transitional jogging under the remote TCP mode, the robot behaves the same as it is under normal jogging.

REMOTE TCP jogging is done by using the following procedure.

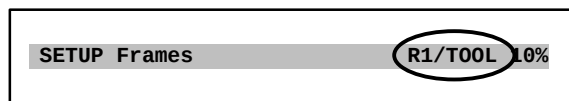
- 1 Enter to REMOTE TCP jogging mode.
- 2 Select the frame.
JOINT, REMOTE TCP USER, REMOTE TCP TOOL, REMOTE TCP JOG
- 3 Jog the robot.

4 Return from REMOTE TCP jogging mode.

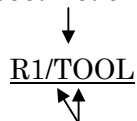
< Change to REMOTE TCP jogging mode >

Please perform the following procedure to enter to REMOTE TCP jogging mode.

- 1 Press FCTN.
Select "TOGGLE REMOTE TCP".
The screen is changed as follows.



The robot motion is performed according to REMOTE TCP TOOL No.1



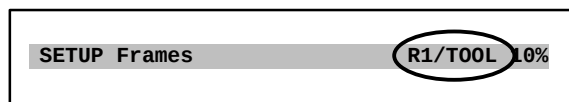
REMOTE TCP motion is performed according to the tool frame.

Please perform the following procedure to return from REMOTE TCP jogging mode.

- 1 Press FCTN.
- 2 Select "TOGGLE REMOTE TCP".

Select frame

You change the coordinate system by pressing the COORD key. Please select the frame except for JOINT.



The display status is changed as follows.

JOINT → R1/JFRM → R1/TOOL → R1/USER

Select REMOTE TCP frame

When the robot enters to REMOTE TCP mode, you can select REMOTE TCP frame by using how to select the normal frame.

- 1 Press FCTN.
- 2 Select "CHANGE RTCP FRAME".

The number is changed as follows.

1 → 2 → 3 → 4 → 5 → 1

The screenshot shows a screen with a table of coordinates and a circled "R1/TOOL 10%" at the top right. The table has columns for "User/RTCP", "Setup/Direct", "Entry", and "Comment". The rows are numbered 1 to 5. The "R1/TOOL 10%" text is circled in black.

	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	[]

[TYPE] [DETAIL] [OTHER] [CLEAR] [SETIND] >

Programming and running by using REMOTE TCP

If you specify REMOTE TCP motion option, the robot moves remote TCP instead of robot TCP.

Please perform the following procedure to specify REMOTE TCP motion option.

- 1 Move the cursor to end of program line.
- 2 Press F4,[CHOICE].

- 3 Select "RTCP".

Please perform the following procedure to remove REMOTE TCP motion option.

- 1 Move the cursor to end of program line.
- 2 Press F4,[CHOICE].
- 3 Select "No Option".

You can not use RTCP with JOINT motion.

<Programming example>

```
L P[1] 100mm/sec FINE RTCP
```

The work is moved to P[1] by the relative speed 100 mm/sec between the work and the remote tool.

```
C P[1]
P[2] 100mm/sec FINE RTCP
```

The work is moved to P[2] via P[1] by the relative speed 100 mm/sec between the work and the remote tool.

9.13 HIGH-SENSITIVITY COLLISION DETECTION

9.13.1 Overview

High-sensitivity collision detection 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.

9.13.2 Specification

- 1) When collision is detected, the function issues an alarm and stops the robot quickly by decelerating it in such a way that shocks to the robot can be decreased.
- 2) Since the detection sensitivity has previously been adjusted for each robot type, you can use this function without adjusting it.
- 3) You can change the detection sensitivity by the program instruction or in the setting screen.
- 4) You can temporarily enable/disable this function by the program instructions. (It is impossible to disable this function regularly.)
- 5) You can output signal to a DO when collision is detected.
- 6) You can output status of the collision detection (enabled / disabled) to a DO.
- 7) The detection sensitivity is automatically increased during teaching operation.
- 8) This function is enabled since the power of the robot controller is on.

9.13.3 Settings

Payload information (Basic)

Set the function with load information and the information about devices installed on the robot.

Since the function uses the load information and device information to detect a collision, it is necessary to set the function with these pieces of information. Be sure to specify the weight of the load, the center of gravity, and the weight of each device on the robot accurately.

If the inertia (shape) of the load is large, it may be necessary to specify the inertia around the gravity center of the load. (If the tool is big, and simply specifying its weight and gravity center does not assure accurate detection, specify its inertia.)

See Section 9.14, "LOAD SETTING," or Section 9.15, "LOAD ESTIMATION," for how to specify load information.

Collision Guard Setup screen (Optional)

The detection sensitivity has previously been adjusted for each robot type. If you want to change the sensitivity, set in this screen.

Also, If you want to output the status of the collision detection (enabled / disabled), set in this screen.

See Section 9.13.4, "Collision Guard Setup Screen."

Program instructions (Optional)

If it is previously anticipated that a strong force will be exerted during an operation and users want to temporarily disable collision detection or change the detection sensitivity, insert program instructions. See Section 9.13.5, "Program Instructions."

9.13.4 Collision Guard Setup Screen

Set as below.

- 1 Push menu key.
- 2 Select "6 SETUP."
- 3 Push F1(TYPE.)
- 4 Select "Coll Guard". The screens as below is shown.

Fig. 9.13.4 Collision Guard Setup Screen

COL GUARD SETUP	
Group:	1
Collision Guard status:	ENABLED
1 Sensitivity:	100%
2 Sensitivity Def. Reg.:	R[0]
3 Collision Guard Error:	DO[0]
4 Col. Guard enabled:	DO[0]
[TYPE] HELP GROUP	

- 5 Setup each item of Collision Guard Setup screen referring Table 9.13.4.

Table 9.13.4 Setup Items of Collision Guard Setup Screen

Item	default	Description
Group	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 F3(GROUP) and entering the group number you wish to display. You will only be able to select a group that supports High-Sensitivity Collision Detection.
Collision Guard Status	ENABLED	This item specifies whether collision detection is enabled or disabled for the currently selected group. This item cannot be changed in this setup screen.
Sensitivity	100%	This item allows you to set the level of sensitivity for Collision Detection for the currently selected group. Minimum is 1%. Maximum is 200%. 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 Def. Reg.	0	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 Detection within a program. A register number of 0 indicates that the register is not used.

Item	default	Description
Collision Guard Error	0	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	0	This item allows you to specify a Digital Output that will be turned on whenever Collision Detection is enabled via this setup menu, and turned off whenever Collision Detection is disabled via this setup menu. A value of 0 indicates that no output will be used.

**CAUTION**

You can make the collision detection sensitivity insensitive by setting "Sensitivity" less than 100, but in this case, this function cannot work effectively. Please use this function by not decreasing "Sensitivity" as much as possible, and if you decrease "Sensitivity" by necessity, please be very careful.

If robot repeatedly absorb force as much as collision is detected, the mechanical parts become strained and the life of the robot is shorten. It is desirable to change program to lessen the force instead of changing the sensitivity.

NOTE

It is impossible to set the "Collision Guard Setup" at the group which includes a robot that is not supported "High-Sensitivity Collision Detection" function.

9.13.5 Program Instructions

The program instructions below are reserved.

- COL DETECT ON
- COL DETECT OFF
- COL GUARD ADJUST

9.13.5.1 COL DETECT ON / COL DETECT OFF

These instructions allows you to enable/disable collision detection during program execution. By default, collision detection is enabled.

- To disable collision detection temporarily, include the "COL DETECT OFF" instruction in a teach pendant program.
- To enable collision detection again, include the "COL DETECT ON" instruction in a teach pendant program.

When the program is finished or aborted, collision detection turns enabled automatically.

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[2] 50% FINE

```

This program disables collision detection with lines 12 to 14.

**CAUTION**

While collision detection is disabled, not only "High-Sensitivity Collision Detection" function but also the basic collision detection function is disabled. Please be careful adequately. For safety's sake, please set the collision detection disable region as short as possible.

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 Detection will only be disabled for group 2; it will remain enabled for all other groups.

9.13.5.2 COL GUARD ADJUST

This instruction allows you to adjust the sensitivity of collision detection 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.

This instruction can be used in three way. (See Table 9.13.5.2.)

Table 9.13.5.2 COL GUARD ADJUST instruction

number of arguments	the new sensitivity value
0	value of register whose register number is specified in the Collision Guard Setup menu
1 (directly set)	value entered explicitly
1 (indirectly set)	value of set register

**CAUTION**

You can make the collision detection sensitivity insensitive by setting "Sensitivity" less than 100, but in this case, this function cannot work effectively. Please use this function by not decreasing "Sensitivity" as much as possible, and if you decrease "Sensitivity" by necessity, please be very careful.
If robot repeatedly absorb force as much as collision is detected, the mechanical parts become strained and the life of the robot is shorten. It is desirable to change program to lessen the force instead of changing the sensitivity.

NOTE

"COL GUARD ADJUST" instruction 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 High-Sensitivity Collision Detection, 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.

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
16: COL GUARD ADJUST R[1]
17: 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.

9.13.6 Cautions

- 1 Collision Detection might detect a false collision when a collision has not occurred in the following cases:
 - Payload information has not been set correctly.
 - The payload is larger than the maximum payload for the robot, or the inertia of the payload is too large.
 - Not enough voltage has been supplied to the controller.
 - Low temperature.
 - The ACC motion option has been used, causing jerky robot motion.
 - Severe motion such as reverse motion which Cnt. is used.
 - Linear motion occurs near singularity point where axes revolve in high speed.

Action: If collision misdetection occurred for the reasons above, try removing these reasons first. If necessarily, insert COL GUARD ON / OFF or COL GUARD ADJUST instructions at the top and bottom of where misdetection occurs for avoiding alarm stop.
- 2 Collision Detection is disabled in the following cases:
 - Soft Float is enabled.
 - The robot brakes are on.
 - Calibration is not finished.
 - While not releasing SHIFT key after pushed SHIFT + RESET
- 3 In order to decrease the force of collision, Collision Detection allows the robot axes to sag away from the collision for a short time after detecting a collision. When this happens, vertical robot axes might fall slightly after detecting a collision, due to the effect of gravity.

9.14 LOAD SETTING

9.14.1 Overview

Setting the information about the load on the robot appropriately can cause the following effects:

- Increase in motion performance (such as lower vibration and shorter cycle time)
- More effective reaction of functions related to dynamics (such as increase in performance related to collision detection and gravity compensation)

For effective use of the robot, it is recommended to appropriately set information about loads such as the hand, workpiece, and devices mounted on the robot.

A load estimation function is optionally available. This function enables the robot to calculate load information automatically.

9.14.2 Motion Performance Screens

There are three motion performance screen types: List screen, load setting screen, and device setting screen. They are used to specify load information and the information about devices on the robot.

These screens let you easily specify the information that has conventionally been set in system variables (\$PAYLOAD, \$PAYLOAD_X, \$PAYLOAD_Y, \$PAYLOAD_Z, \$PAYLOAD_IX, \$PAYLOAD_IY, and \$PAYLOAD_IZ in the \$PARAM_GROUP). They also let you switch the load setting among two or more loads.

- 1 Press **MENUS** to display the screen menu.
- 2 Select "6 SYSTEM" described on the next page.
- 3 Press **F1 (TYPE)** to display the screen switch menu.
- 4 Select **Motion**. The list screen appears. (If any other screen appears, press **[PREV]** several times until the list screen appears.) For a multi-group system, the list screen of another group can be reached by pressing **F2 (GROUP)**.

MOTION PERFORMANCE			JOINT 10%
Group1			1/10
No.	PAYLOAD[kg]	Comment	
1	0.00	[]	
2	0.00	[]	
3	0.00	[]	
4	0.00	[]	
5	0.00	[]	
6	0.00	[]	
7	0.00	[]	
8	0.00	[]	
9	0.00	[]	
10	0.00	[]	
Active PAYLOAD number = 0			
[TYPE] GROUP DETAIL ARMLoad SETIND >			

- 5 Load information can be specified for condition No. 1 to No. 10. As stated later, an appropriate condition number can be selected as the load is changed by a hand change. Move the cursor to the desired No., and press **F3 (DETAIL)** to display the related load setting screen.

MOTION/PAYLOAD SET				JOINT 10%
				1/8
Group 1				
1	Schedule No[1]:	[*****]		
2	PAYLOAD	[kg]		0.00
3	PAYLOAD CENTER X	[cm]		0.00
4	PAYLOAD CENETR Y	[cm]		0.00
5	PAYLOAD CENTER Z	[cm]		0.00
6	PAYLOAD INERTIA X	[kgfcm ²]		0.00
7	PAYLOAD INERTIA Y	[kgfcm ²]		0.00
8	PAYLOAD INERTIA Z	[kgfcm ²]		0.00
[TYPE]	GROUP	NUMBER	DEFAULT	HELP

- 6 Specify the mass and gravity center of the load, and inertia around its gravity center.
The X, Y, and Z directions displayed on the load setting screen are in reference to the default tool coordinate system (which is valid when no other tool coordinate system is set up).

NOTE

$$1 [\text{kgf cm s}^2] = 980 [\text{kg cm}^2]$$

When a value is entered, the confirmation message Path and Cycle time will change. Set it? appears. Press F4 (YES) or F5 (NO) whichever is necessary.

- 7 Pressing F3 “NUMBER” lets you go to the load setting screen for another condition No. For a multi-group system, pressing F2 “GROUP” lets you move to the setting screen of another group.
- 8 Press PREV to go back to the list screen. Press F5 “SETIND”, and enter a desired load setting condition No. The last condition No. selected is used during program execution and jog operation. (The initial condition No. is 0. Using the condition without changing from the initial setting causes the initial system variable setting to be used. Using the setting on the load setting screen requires enabling that setting.)
- 9 Pressing F4 “ARMLoad” on the list screen lets you move to the device setting screen.

MOTION/ARMLoad SET				JOINT 10%
Group 1				
1	ARM LOAD AXIS #1	[kg]		0.00
2	ARM LOAD AXIS #3	[kg]		0.00
[TYPE]	GROUP		DEFAULT	HELP

- 10 Specify the mass of the devices on the J1 and J3 arms.
Entering values displays the message Path and Cycle time will change. Set it?. Press F4 (YES) or F5 (NO) whichever is necessary.
After setting the mass of a device, turn the power off and on again.

9.14.3 Program Instructions

Pressing F5 “SETIND” on the list screen lets you switch the screen, using program instructions rather than selecting a desired load setting condition No. (Even after program execution is finished, the last condition No. selected is used during later program execution and jog operation.)

Instruction		JOINT 10%
1 Miscellaneous	5 Multiple control	
2 Skip	6 SENSOR	
3 Payload	7 Program control	
4 Offset/Frames	8 ---next page---	
PRG		

(1) PAYLOAD[i]

This instruction changes the load setting condition No. to be used to i.

PAYLOAD statement		JOINT 10%
1 PAYLOAD[...]	5	
2	6	
3	7	
4	8	
PRG		

Example

1: PAYLOAD[i]

This program selects load setting condition 1.

Multi-operation group environment

The PAYLOAD[i] instruction usually selects a load setting condition No. for all operation groups enabled for the program. For a multi-group system, however, it is possible for this instruction to specify what group to be subjected to load setting condition No. switching.

PRG		JOINT 10%
		1/2
1	PAYLOAD[...]	
[End]		
Enter value		
GROUP	DIRECT INDIRECT	

Pressing F1 (GROUP) displays a menu that contains choices for specifying a group. You can select a group from the menu.

Example

PAYLOAD[GP2,3:1]

This program selects load setting condition No. 1 for groups 2 and 3.

9.15 LOAD ESTIMATION

9.15.1 Overview

Load estimation is a function for estimating the weight of the load, such as tool and workpiece, mounted on the hand of the robot.

The function enables the information stated above to be estimated automatically by running the robot.

Using the load estimation function requires the load estimation option (A05B-****-J669)(*). Using the function also requires that your model support the load estimation function. If your model does not support the function, you cannot use it.

NOTE

(*) The portion indicated **** is a four-digit number that varies with the series. For the 7D70 series, for example, the option code is A05B-2500-J669.

9.15.2 Operating Procedure

Load is estimated in the following flow:

- 1 Set the range of motion to be subjected to load estimation
- 2 Execute load estimation.

Once a mechanical part such as a motor is replaced, it becomes necessary to make calibration.

If no calibration is made after mechanical part replacement, the precision of load estimation becomes lower.

9.15.3 Load Estimation Procedure (for 6-Axis Robots)

This procedure is performed on the load estimation screen.

This screen is entered from the motion performance screen.

- 1 Press **MENUS** to display the screen menu.
- 2 Select “6 SYSTEM” described on the next page.
- 3 Press **[F1] (TYPE)** to display the screen switching menu.
- 4 Select Motion. The list screen appears. (If any other screen appears, press **[PREV]** several times until the list screen appears.) For a multi-group system, the list screen of another group can be reached by pressing **F2 (GROUP)**.

MOTION PERFORMANCE			JOINT 10%	
Group1			1/10	
No.	PAYLOAD[kg]		Comment	
1	0.00	[		]
2	0.00	[		]
3	0.00	[		]
4	0.00	[		]
5	0.00	[		]
6	0.00	[		]
7	0.00	[		]
8	0.00	[		]
9	0.00	[		]
10	0.00	[		]
Active PAYLOAD number = 0				
[TYPE] GROUP DETAIL ARMLoad SETIND >				

- 5 Press **NEXT**, then **[F2] F2 “IDENT”**. The load estimation screen appears.

MOTION/PAYLOAD		JOINT 10%
		1/4
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 [Kg 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 *****		
[TYPE]	GROUP	NUMBER EXEC APPLY >

- 6 Place the robot in the position where load estimation is to be performed.



CAUTION

- 1 Only the J5 and J6 axes move during load estimation. The other axes stay in the position where they are when load estimation begins.
The range of motion is defined as an interval between two points specified on estimation position 1 and 2 screens. (See steps 10 and 12.)
- 2 Put the J5 rotation axis in a horizontal position.
The more vertical posture the J5 rotation axis takes, the lower the precision of estimation becomes.

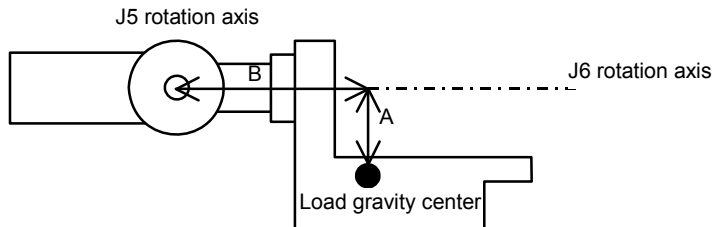
- 7 Press F3 “NUMBER”, and select the load setting condition No. for which a load estimate is to be set up.
- 8 If the mass of the load for which load estimation is to be performed is known, move the cursor to line 2, select “YES”, and specify (enter) the mass.

NOTE

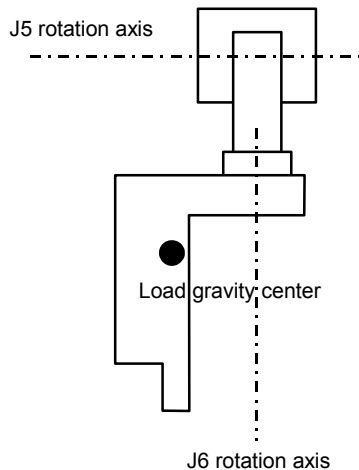
The estimation precision becomes higher when a mass is specified. Specify the mass as much as possible.

Even if no mass is specified, estimation is possible provided that the following condition is satisfied. However the precision becomes lower.

- The moment around the J5 and J6 axes must be sufficiently high.



- The mass must be sufficiently great, and the distance between points A and B must be sufficiently large.
- The load gravity center must be sufficiently far from the J5 and J6 rotation axes.
- As for positions set up on estimation position 1 and 2 screens, the gravity center of the load must be in or near the plane that contains the J5 and J6 rotation axes.



- As for the J6 axis, the interval between points specified on the estimation position 1 and 2 screens must be 180° in terms of angle.

9 Press NEXT, then [F4] (DETAIL). The estimation position 1 screen appears.

MOTION/ID POS1 ID		JOINT 10%
		1/5
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%>
[TYPE] POS.2 DEFAULT MOVE_TO RECORD		

- 10 Specify estimation position 1. (Alternatively, the initial value can be used.)
Specify the positions of the J5 and J6 axes by entering their values directly. Alternatively, move the robot to the desired position by jogging, then press [Shift] + F5 "RECORD" to record the position. Now pressing [Shift] + F4 "MOVE_TO" moves the robot to estimation position 1. Use this procedure to identify the set position.
- 11 Pressing [F2] POS.2 displays the estimation position 2 screen.

MOTION/ID POS2 ID		JOINT 10%
		1/5
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%>
[TYPE] POS.1 DEFAULT MOVE_TO RECORD		

- 12 Specify estimation position 2. (Alternatively, the initial value can be used.)
Specify the positions of the J5 and J6 axes by entering their values directly. Alternatively, move the robot to the desired position by jogging, then press [Shift] + F5 "RECORD" to record the position. Now pressing [Shift] + F4 "MOVE_TO" moves the robot to estimation position 2. Use this procedure to identify the set position.
- 13 Press [PREV] to return to the estimation screen.
- 14 Set the teach pendant enable switch to OFF, and press F4 "EXEC". The message Robot moves and estimates. Ready? appears.
- 15 Specify whether to execute load estimation. (Selecting "YES" causes the robot to move. Pay sufficient care to avoid danger.)
 - To perform load estimation by running the robot, press [F4] (YES).
 - To quit execution, press [F5] (NO).
- 16 After low-speed and high-speed operations are finished, load information is estimated. (Operation switches automatically from low speed to high speed. Even when the robot is running at low speed, do not get close to it, because otherwise you may get in a dangerous situation when the robot suddenly starts running at high speed.)
- 17 Press F5 "APPLY" to set the estimate at a load setting condition No. The message Path and Cycle time will change. Set it? appears.
- 18 Specify whether to set the estimate.
 - To set the estimate, press [F4] (YES).
 - Not to set the estimate, press [F5] (NO).
- 19 If the value to be set is greater than the maximum allowable load (indicated in parentheses), the message Load is OVER spec ! Accept? appears. Specify whether to set this value, just as in the above step.

9.15.4 Calibration Procedure (for 6-Axis Robots)

Once a mechanical part such as a motor is replaced, it becomes necessary to make calibration. If no calibration is made after mechanical part replacement, the precision of load estimation becomes lower. Calibration is controlled, using the load estimation screen. Calibration is started by setting the calibration switch to ON and executing load estimation.

- 1 Make sure that there is nothing on the hand of the robot.
Calibration must be made without attaching anything to the hand of the robot.

NOTE

If calibration is performed with anything attached to the robot hand, incorrect calibration data is set up, thus hampering a normal estimation.
In this case, make calibration again, properly this time.

- 2 Press MENUS to display the screen menu.
- 3 Select "6 SYSTEM" described on the next page.
- 4 Press [F1] (TYPE) to display the screen switching menu.
- 5 Select Motion. The list screen appears. (If any other screen appears, press [PREV] several times until the list screen appears.) For a multi-group system, the list screen of another group can be reached by pressing F2 "GROUP".

MOTION PERFORMANCE			JOINT 10%
Group1			1/10
No.	PAYLOAD[kg]	Comment	
1	0.00	[	]
2	0.00	[	]
3	0.00	[	]
4	0.00	[	]
5	0.00	[	]
6	0.00	[	]
7	0.00	[	]
8	0.00	[	]
9	0.00	[	]
10	0.00	[	]
Active PAYLOAD number = 0			
[TYPE] GROUP DETAIL ARMLoad SETIND >			

- 6 Press NEXT, then F2 "IDENT". The load estimation screen appears.

MOTION/PAYLOAD		JOINT 10%
		1/4
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 [Kg 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	*****
[TYPE] GROUP NUMBER EXEC APPLY >		

- 7 Place the robot in the position where load estimation is to be performed.

NOTE

- 1 Only the J5 and J6 axes move during load estimation. The other axes stay in the position where they are when load estimation begins.
The range of motion is defined as an interval between two points specified on estimation position 1 and 2 screens. (See steps 9, 10, and 12.)
- 2 Put the J5 rotation axis in a horizontal position.
The more vertical posture the J5 rotation axis takes, the lower the precision of estimation becomes.

- 8 Press NEXT, then F4 (DETAIL). The estimation position 1 screen appears.

MOTION/ID POS1 ID		JOINT 10%
		1/5
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%>
[TYPE] POS.2 DEFAULT MOVE_TO RECORD		

- 9 Specify estimation positions 1 and 2. Try to use default values as much as possible.
Press F3 “DEFAULT”, and specify default values for estimation positions 1 and 2, speed, and acceleration.
- 10 Pressing [Shift] + F4 “MOVE_TO” moves the robot to estimation position 1. Make sure that it is safe to move the robot to estimation position 1.
If it is dangerous to move the robot to estimation position 1, manipulate the J1 to J4 axes by jogging to move the robot to a position where the robot can move safely.
- 11 Pressing F2 “POS.2” displays the estimation position 2 screen.

MOTION/ID POS2 ID		JOINT 10%
		1/5
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%>
[TYPE] POS.1 DEFAULT MOVE_TO RECORD		

- 12 Pressing [Shift] + F4 "MOVE_TO" moves the robot to estimation position 2. Make sure that it is safe to move the robot to estimation position 2.
If it is dangerous to move the robot to estimation position 2, manipulate the J1 to J4 axes by jogging to move the robot to a position where the robot can move safely.
If you moved any of the J1 to J4 axes, press F2 "POS.2" to go back to the estimation position 1 screen, and follow this procedure again from step 10.
- 13 Press [PREV] to return to the load estimation screen.
- 14 Move the cursor to CALIBRATION MODE on line 3 to turn it "on."

NOTE

Once calibration is completed, CALIBRATION MODE becomes "off" automatically.
Do not change CALIBRATION MODE during calibration or load estimation.
Otherwise, calibration may be made incorrectly or may not be made at all.

- 15 Move the cursor to line 4 (so that "EXEC" appears at [F4]), and set the teach pendant enable switch to OFF, then press "EXEC". The message Robot moves and estimates. Ready? appears.
- 16 Specify whether to perform load estimation. (Selecting "YES" causes the robot to move. Pay sufficient care to avoid danger.)
 - To perform load estimation by running the robot, press [F4] (YES).
 - To quit execution, press [F5] (NO).
- 17 After low-speed and high-speed operations are finished, calibration is completed. (Operation switches automatically from low speed to high speed. Even when the robot is running at low speed, do not get close to it, because otherwise you may get in a dangerous situation when the robot suddenly starts running at high speed.)

9.15.5 Other Related Matters

(1) Motion range

If the motion range between estimation positions 1 and 2 becomes narrower, the estimation precision may get lower.

The actual motion range should preferably be as wide as the default motion range.

(2) Acceleration for motion used in load estimation

The estimation precision is low for the load whose moment inertia is relatively low compared with the maximum allowable load of the robot. This is because the influence by the moment inertia to the torque of the robot motor is weak.

The estimation precision for this light load may be able to be increased by increasing the acceleration used during operation for load estimation.

Try to increase the acceleration by specifying a larger value in "ACCEL - High" on the estimation position 1 and 2 screens; however, do not specify so large a value that vibration becomes serious during operation.

(3) Calibration data

The following system variable hold calibration data.

\$PLCL_GRP[group].\$TRQ_MGN[axis]

group: Group number

axis: Axis number

If improper calibration data is set up, for example, by making calibration with a load mounted by mistake, reassigning the previous data to the system variable can restore the previous calibration data.

It is recommended to keep note of the previous calibration data so as to enable restoration.

9.16 COLLISION DETECTION for AUXILIARY AXIS

9.16.1 General

The Collision Detection Function is the feature that stops the robot immediately and reduces the damage to the robot, when the robot collides with other objects.

Generally, this feature has been applied for the robot axes.

But, this feature has not been applied for the auxiliary axis. Because of the auxiliary axis is design by customer, then the parameters for this feature can not be set beforehand.

To apply this feature to the auxiliary axis, the parameter tuning is required with tuning procedure on this manual.

NOTE

To tune the collision detection parameters for auxiliary axis,
Collision Detection for Auxiliary Axis Option (A05B-2500-J645)
or
High Sensitive Collision Detection Package Option (A05B-2500-J684) that
includes above is required.

9.16.2 Caution

The load ratio of auxiliary axis should be less than 5.

Load ratio = (Load Inertia + Motor Inertia) / Motor Inertia

When the auxiliary axis is designed, you must consider above.

If the load ratio of auxiliary axis is more than 5 times, the motion performance and sensitivity for collision detection may deteriorate.

9.16.3 Initial Setting

- 1 Setup auxiliary axis (Gear ratio, acceleration time, and etc.) normally
- 2 Turn power on
- 3 Set the following system variables
 $\$SBR[n].\$PARAM[112] = 2097152 / (\$SBR[n].\$PARAM[47])$
 $\$SBR[n].\$PARAM[119] = 7282$
 $\$SBR[n].\$PARAM[120] = -7282$
 n : Hardware axis number of auxiliary axis
 n=7~ for aux. Axis / n=1~6 for robot axes
- 4 Cycle power

9.16.4 Tuning Procedure

The sensitivity of collision detection will be tuned by below procedure. It should be tuned without mis-detection.

- 1 Create the program that includes heavy motion like an inverse motion with CNT100 beforehand.
 If the program for production is already exist, It can be used to tune. In this case, the sensitivity can be optimized for production with this program.
 (However, if other program was run, the mis-detection might occur. Also, if this program was modified, the re-tuning might be required.)
- 2 Run the above program.
 Also this program must not be paused.
 Because of the disturbance torque, see below, will be cleared at just re-start the program

- 3 Measure the max. / min. disturbance torque on STATUS/AXIS/ DISTURB screen after running the program.

PNS0001	Line 1	ABORTED
STATUS Axis	G1	JOINT 30%
GRP[1]		
Disturbance Torque (A)		
Curr. / Max.(Allowed)/Min.(Allowed)		
J1 :	0.0 20.0(40.0)	-19.0(-40.0)
J2 :	0.0 19.0(40.0)	-20.0(-40.0)
J3 :	0.0 22.0(40.0)	-10.0(-40.0)
J4 :	0.0 12.0(20.0)	-5.0(-20.0)
J5 :	0.0 10.0(20.0)	-11.0(-20.0)
J6 :	0.0 8.0(20.0)	-4.0(-20.0)
J7 :	0.0 24.0(56.0)	-30.0(-56.0)
[TYPE] MONITO TRACKING DISTURB [UTIL] >		

As said above, the disturbance torque will be reset at the start of each program.

If there are some programs,

- make new program that call all program for tuning and run this main program.
 - record the max. / min. disturbance torque for each programs and find max. / min. value in these recorded value.
- 4 Move the cursor to allowed value in parentheses for the axis.
Change the allowed value to same as measured max. or min. value.

PNS0001	Line 1	ABORTED
STATUS Axis	G1	JOINT 30%
GRP[1]		
Disturbance Torque (A)		
Curr. / Max.(Allowed)/Min.(Allowed)		
J1 :	0.0 20.0(40.0)	-19.0(-40.0)
J2 :	0.0 19.0(40.0)	-20.0(-40.0)
J3 :	0.0 22.0(40.0)	-10.0(-40.0)
J4 :	0.0 12.0(20.0)	-5.0(-20.0)
J5 :	0.0 10.0(20.0)	-11.0(-20.0)
J6 :	0.0 8.0(20.0)	-4.0(-20.0)
J7 :	0.0 24.0(24.0)	-30.0(-30.0)
[TYPE] MONITO TRACKING DISTURB [UTIL] >		

⚠ CAUTION

When the disturbance torque exceeds above allowed value, the following WARNING occurs

SRVO-053 Disturbance excess (G:x,A:x)

Following servo alarm (servo power off) occurs when the disturbance torque exceeds below ALARM LEVELs.

**Upper Limit = Max. allowed value + 0.3 ×
Max. current of amp.**

**Lower Limit = Min. allowed value - 0.3 ×
Max. current of amp.**

SRVO-050 Collision Detect alarm (G:x,A:x)

Part of 0.3 × Max. current of amp. is the margin to prevent the miss-detection.

For example in above screen with 40A amplifier,

Upper Limit = 24.0 + 0.3 × 40 = 36 A

Lower Limit = -30.0 - 0.3 × 40 = -42 A

- 5 Run the programs again with above disturbance allowed setting, and confirm that there is no miss-detection.
- 6 Finished.

9.17 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 (A05B-2500-J649) is necessary to use this function.

This function can not be used with Soft float (A05B-2500-J612) or Small Circle (not supported).

9.17.1 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.

9.17.2 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 Screen (Default screen)

MOTION PERFORMANCE			JOINT 10%
Group1			1/10
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.

PAYLOAD SET Screen

MOTION/PAYLOAD SET				JOINT 10%
				1/8
Group 1				
1	Schedule No[1]:	[Comment]		
2	PAYLOAD	[kg]		100.00
3	PAYLOAD CENTER X	[cm]		10.00
4	PAYLOAD CENETR 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 Cycle time 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 10%
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 Cycle time will change. Set it ?" is displayed. Please input F4(YES) or F5(NO).
After setting up the armload, cycle power.

9.17.3 Mastering

When gravity compensation is enabled, follow the procedure below to perform mastering.

- 1 Disable Gravity Compensation
- 2 Perform mastering
- 3 Enable Gravity compensation

9.18 PASSWORD FUNCTION

9.18.1 Overview of the Password Function

A password is a combination of up to 12 characters, digits, and symbols. It is used to restrict various operations and screen accesses by operators. Password protection is disabled unless an installation user is defined.

There are eight password levels, and the possible operations and displayable screens differ depending on the level. Levels 3 to 7 are user-definable. For an overview of password authentication for the four levels, see Table "Password levels".

Table 9.18.1 Password levels

Level	Operation
Install	Enables the user to allocate user names, passwords , and levels . Enables the user to clear user names and passwords. Enables the user to enable or disable the password log. Allows the user to set the number of password users in the system. Enables the user to execute all operations that settings users, programs, and operators can executed. Caution Only one install user can be set.
Setup	Enables the user to execute operations generally performed for system setting.
Program	Enables the user to execute operations more advanced than those that production users can execute.
Operator	Enables the user to execute basic operations.
User-Defined	Enables the user to execute basic operations unless otherwise defined in the password configuration file.

**CAUTION**

In a standard system, the user is set to the operator level when the controller is turned ON.

**WARNING**

There are some functions that cannot be executed without the password of the install user. If the password of the install user is forgotten, contact the FANUC service representative.

Password operations

To use passwords, set an install user first. To set an install user, it is necessary to assign an install user name and a password first and then *log in*. Once logged in, the install user can *assign* the user name, level, and password for each of other users.

**CAUTION**

The password function cannot be used unless an install user name and a password are assigned.

After the install user has assigned a user name, password level, and password to a user, this user must *log in* to work at the assigned level. The user selects his user name and input his password to log in. Only one user can log in to one device at a time. The term device, as used here, refers to a teach pendant, CRT/keyboard, KCL, and so on.

When the user has completed work, he *logs out*. If he does not log out, a *timeout* occurs when the time specified for user timeout has elapsed. When the time specified for user timeout has elapsed or the user logs out, the system returns to the operator level, so that another user can log in. If a user forgets to log out, there are cases in which the user is forced to log out by another user.

If the install user enables log recording on the password setting screen, the password log can be displayed on the alarm screen. In the password log, important information about data changes (descriptions of changes made by users and the times the changes were made) is displayed. For details, see Procedure 9-20 "Displaying the password log".

For information on the assignment of a user name, password level, and password by the install user, see the Subsection 9.18.2, "Password Operations by the Install User". For information on each of operator, program, and setup users, see the Subsection 9.18.4, "Password Operations by Program Users and Setup Users".

9.18.2 Password Operations by the Install User

The install user is required to perform the operations below.

- Assign an install user name and a password (Procedure 9-12 "Assignment a user name and a standard password for each password level")
- Assign the user names, levels, and passwords for all other users (Procedure 9-12 "Assignment a user name and a standard password for each password level")
- Enable or disable the password log and display it if enabled (Procedures 9-19 "Enabling the password log" and Procedures 9-20 "Displaying the password log" in the "Password Log" section)

For information on setting a password level, see Procedure 9-12 "Assignment a user name and a standard password for each password level".

Procedure 9-12 Assignment a user name and a standard password for each password level

Step

- 1 Press the MENUS key.
- 2 Select "SETUP".
- 3 Press F1 [TYPE].
- 4 Select "Passwords". The screen below appears.

SETUP Passwords		JOINT 10%
Passwords are inactiv. Press LOGIN to activate.		
[TYPE]	LOGIN	

- 5 Press F2 "LOGIN". The screen below appears.

SETUP Passwords		JOINT 10%
USERNAME	PWD	LEVEL
1	*	INSTALL
		TIME(min)
		0



CAUTION

It is necessary to set an install user name and a password first.

- 6 Use the steps below to assign an install user name and a password.
 - a Press the ENTER key.
 - b Use the arrow keys and function keys as appropriate to input an install user name. When finished, press the input key. The screen below appears.

SETUP Passwords		JOINT 10%
--Set password for BOB		--
Old password:	'	'
New password:	'	'
Verification:	'	'
Old Value:		

**CAUTION**

A password must consist of at least three characters.

**WARNING**

Take a note of the install password. There are some functions that cannot be executed without the install password. If you forget the install password, contact the FANUC service representative.

- c Input a new password, and press the input key.
- d Input the new password again to confirm that the previous input is correct and then press the input key. The screen below appears.

```

Would you like to be logged in?[YES]
                                YES    NO
  
```

- 7 **To log in**, press F4 "YES". Otherwise, press F5 "NO".

**CAUTION**

Unless logged in as an install user, the user cannot set other users.

If F4 "YES" is pressed, the screen below appears.

```

SETUP Passwords                                JOINT 10%
                                           1/2
Current user:      BOB
Current level:     INSTALL
1 Default user timeout: 15 min
2 Timeout occurs:  15 min
3 Log events:      DISABLED
4 Number of users: 10
  
```

- 8 If F2 "LOGIN" is pressed, the user list screen is displayed as follows.

```

SETUP Passwords                                JOINT 10%
                                           1/10
  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.
  
```

**CAUTION**

The @, which appears when the user logs in, is a symbol indicating a login user name.

- 9 **Use the steps below to assign the name, password, and level for a second user.**
- a Move the cursor to the second user name input item, press the input key, and then use the function keys as appropriate to input a name.

- b Move the cursor to the PWD field, press the input key and then use the function keys as appropriate to input a password.
- c Move the cursor to the LEVEL field, press F4 [CHOICE] and then select a level.
- d Move the cursor to the TIME field and input a user timeout value. The user timeout value may be in the range of 0 to 10080 minutes (7 days).
- e For other users that are to be permitted to access the system, perform a through d of step 9.

 CAUTION

If the user logs in with a user timeout value of 0, no timeout is performed. A value greater than 0 indicates the length of time (in minutes) of the pause until the system performs a timeout.

- 10 **To clear an input user name and password**, press the Next key and then F2 "CLEAR".
- 11 **To clear the user names and passwords** of all users other than the install user, press the Next key and then F3 "CLR_ALL".
- 12 **To change the number of users in the system, use the steps below.**

 WARNING

If the number of users in the system is changed to a number less than the number of assigned users, some users may be deleted from the system.

- a Press the PREV key to display the password setting screen, which appeared first.
- b Move the cursor to the number of users field and input a value. The allowable number of users is in the range of 10 to 100.

If increasing the number of users

- c Input a new number of users and press the input key. The screen below appears.

Perform COLD start to see new users.

If the number of users is to be decreased, the prompt below appears.

Reconfiguring. DELETE users?[N0]

To delete users, press F4 "YES". To cancel the operation, press F5 "NO".

- d Turn off the controller and back on to enable the new list of users.
- 13 Press F3 "LOGOUT" to **log out**.

 CAUTION

If the user timeout time elapsed or the user logs out, the device in use returns to the operator level. If the controller is turned off, all devices return to the operator level.

9.18.3 Disabling the Password Function

In a system in which a password is set, it is possible to disable the password function. Disabling the password function causes the install user to be cleared. No other users are cleared. Disabling the password function requires the install level. For information on disabling the password function, see Procedure 9-13 "Disabling the password function".

Procedure 9-13 Disabling the password function

Condition

- The user is logged in as an install user.

Step

- Press the MENUS key.
- Select "SETUP".
- Press F1 [TYPE].
- Select "Passwords". The screen below appears.

SETUP Passwords		JOINT 10%
		1/2
Current user:	JOE	
Current level:	INSTALL	
1 Default user timeout:	15 min	
2 Timeout occurs:	15 min	
3 Log events:	DISABLED	
4 Number of users:	10	

- Press NEXT ">".
- Press F3 "DISABLE". The screen below appears.

SETUP Passwords		JOINT 10%
		1/2
Current user:	JOE	
Current level:	INSTALL	
1 Default user timeout:	15 min	
2 Timeout occurs in:	15 min	
3 Log events:	DISABLED	
4 Number of users:	10	
Disable passwords?[NO]		



CAUTION

Disabling the password function causes the install user to be cleared. No other users are cleared. Disabling the password function requires the install level.

- To disable the password function, press F4 "YES". To cancel disabling the password function, press F5 "NO".

9.18.4 Password Operations by Program Users and Setup Users

Program users and setup users can perform the operations below.

- Log in (Procedure 9-14 "Logging in")
- Log out (Procedure 9-15 "Logging out")
- Change their own passwords (Procedure 9-16 "Changing a password")
- Displaying the password log (Procedure 9-20 "Displaying the password log" in the "Password Log" section)

Table 9.18.4 (a) Items on the password setting screen - Login screen

Item	Explanation
Current user	This item indicates the user who is currently logged in.
Current level	This item indicates the current system protection level.
Default user timeout	This item indicates the time after which the system automatically returns to the operator level if the user does not execute any operations after logged in.

Item	Explanation
Timeout occurs in	This item indicates the current remaining time after which the system returns to the operator level.
Log events	This item indicates whether to record operations in the log.
Number of users	This item indicates the maximum allowable number of users in the system.

Table 9.18.4 (b) Items on the password setting screen - User screen

Item	Explanation
USERNAME	This column indicates the names set as password users.
PWD	This column indicates whether passwords are set for users.
LEVEL	This column indicates password protection levels for users.
TIME	This column indicates timeout values for users. A timeout value is the length of time that the login status of a user is retained for the specified password level if he does not perform any operations.
INSTALL user	Assign the user name and password for an install user, so that a user can <i>log in</i> as an install user. Once logged in, the install user can <i>assign</i> the user name, level, and password for each of other users.

Procedure 9-14 Logging in

Condition

- A password has been set. (See the section, "Password Operations by the Install User".)
- No other user is logged in. Only one user can log in at a time.



CAUTION

A user who does not know his user name or password should contact the install user.

Step-a (When no user logs in)

- 1 Display the password setting screen.
 - **If the full menu is used, the procedure is as follows:**
 - a Press the MENUS key.
 - b Select "SETUP".
 - c Press F1 [TYPE].
 - d Select "Passwords".
 - **If the quick menu is used, the procedure is as follows:**
 - a Press the MENUS key.
 - b Select "SETUP PASSWORDS".
- 2 Press F2 "LOGIN". The following screen is displayed.

SETUP Passwords				JOINT 10%
				1/10
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	

- 3 Move the cursor to your user name.
- 4 Press F2 "LOGIN" to **log in**. The screen below appears.

```

SETUP Passwords                                JOINT 10%
--Password for MARY                                --
    Enter password: '                                '
Old Value:

```

- 5 Input your password and press the input key. The follwowing screen is displayed.

```

SETUP PasswordsJOINT 10%
1/2
Current user:          MARY
Current level:         SETUP
1 Default user timeout:      15 min
2 Timeout occurs:           15 min
3 Log events:              DISABLED
4 Number of users:         10

```

- 6 **To change the timeout value**, move the cursor to your user timeout value and input a new timeout value. The user timeout value may be in the range of 0 to 10080 minutes (7 days).

CAUTION

If the user logs in to the teach pendant or CRT unit by setting the user timeout value to 0, no timeout is performed.

Step-b (When someone logs in)

- 1 Display the password setting screen.
 - **If the full menu is used, the procedure is as follows:**
 - a Press the MENUS key.
 - b Select "SETUP".
 - c Press F1 [TYPE].
 - d Select "Passwords".
 - **If the quick menu is used, the procedure is as follows:**
 - a Press the MENUS key.
 - b Select "SETUP PASSWORDS".
- 2 Press F2 "LOGIN". The following screen is displayed.
(In this example, JACK has already logged in.)

SETUP Passwords			JOINT 10%
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

CAUTION

The @, which appears when the user logs in, is a symbol indicating a login user name.

- 3 Move the cursor to your user name.
- 4 Press F2 "LOGIN" to **log in**. Only one user can log in at a time. The following prompt is displayed on only the teach pendant or CRT window.

Procedure 9-16 Changing a password

Condition

- A password has been set. (See the section, "Password Operations by the Install User".)
- The user is currently logged in. (See Procedure 9-14 "Logging in".)

Step

- 1 Display the password setting screen.
 - **If the full menu is used, the procedure is as follows:**
 - a Press the MENUS key.
 - b Select "SETUP".
 - c Press F1 [TYPE].
 - d Select "Passwords".
 - **If the quick menu is used, the procedure is as follows:**
 - a Press the MENUS key.
 - b Select "SETUP PASSWORDS" The screen below appears.

SETUP Passwords		JOINT 10%
Current user:	AAAA	
Current level:	INSTALL	
1 Default user timeout:	15 min	
2 Timeout occurs in:	4 min	
3 Log events:	DISABLED	
4 Number of users:	10	

- 2 Press F4 "PASSWRD". The screen below appears.

SETUP Passwords		JOINT 10%
--Set password for AAAA --		
Old passwords:	'	'
New passwords:	'	'
Verification:	'	'
Old Value:		

- 3 Input the current password and press the input key.
- 4 Input a new password and press the input key.
- 5 Input the new password again to check that the password first input is correct, and then press the input key.
- 6 Press F3 "LOGOUT". Then, follow Procedure 9-14 "Logging in" to log in and set the new password.

9.18.5 Password Configuration File

9.18.5.1 Overview

By using an XML file describing a password configuration, it is possible to set the password function. About the method to create XML file for password function, please refer to "9.18.6 XML Syntax for Password Configuration Files".

For information on creating password configuration XML files, refer to the supplied sample files. If iPendant or Internet Explorer is connected to the robot, the website of the robot can be displayed. From the website, select the link to "Error/Diagnostic files (MD:)" and then select "PASSCFG.DG (Password Configuration)". The screen below appears.

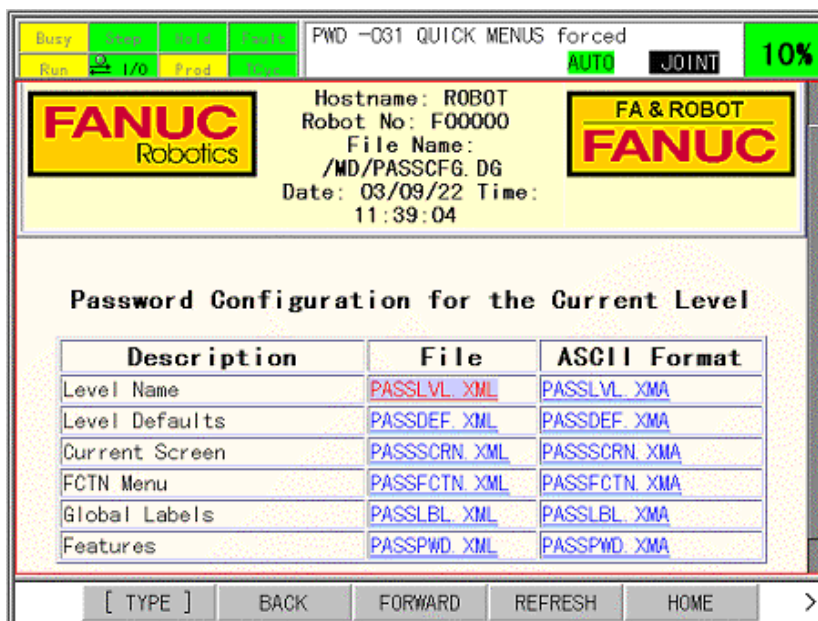


Fig. 9.18.5.1 Password configuration screen

Each of these links leads to an XML file sample. If the website of the robot cannot be displayed or if a file is to be copied, it is possible to copy a file from an MD:unit. To obtain a copy of a file, copy MD:PASSLVL.XML. A sample file describes the password configuration at the level of the user who is currently logged in.

A password configuration is saved as FRS:PASSWORD.DT. When the controller is turned on, this file is loaded so that passwords are configured. The FRS:PASSWORD.DT file can be saved and loaded as an application file on the file screen.

Table 9.18.5.1 "Password configuration settings" lists the settings that can be used for a password configuration. For information on creating a password configuration, see Procedure 9-17 "Password configuration".

Table 9.18.5.1 Password configuration settings

Setting	Explanation
VERIFY	Analyzes a selected XML file and writes alarms to a log file. Does not create actual configuration data. It is possible to select an XML file from an external storage unit. Set the external storage unit on the file screen. (See Chapter 8.)
IMPORT	Selects an XML file from an external storage unit. Set the external storage unit on the file screen. (See Chapter 8.) Analyzes the selected XML file to create configuration data.
EXPORT	Copies the current configuration data to an external storage unit, as password.xml. CAUTION It is necessary to IMPORT the file first.

Procedure 9-17 Password configuration

Condition

- The user is logged in as an install user.

Step

- 1 Press the MENUS key.
- 2 Select "SETUP".
- 3 Press F1 [TYPE].

- 4 Select "Passwords". The screen below appears.

SETUP Passwords		JOINT 10%
Current user:	JOE	
Current level:	INSTALL	
1 Default user timeout:	15 min	
2 Timeout occurs:	15 min	
3 Log events:	DISABLED	
4 Number of users:	10	

- 5 Press the Next key and then F2 "CONFIG". The screen below appears.

Configure Passwords		JOINT 10%
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.		

9.18.6 XML file for Password function

9.18.6.1 Software parts ID, Screen ID

It is necessary to know software parts ID and screen ID in order to setup password for each screen. The following procedure is how to look into the software parts ID and screen ID of target screen.

Procedure 9-18 Looking into software parts ID and screen ID of target screen

Step

- 1 Make display to double
Press shift key and display key. Display menu window like figure 9.18.6.1 (a) is appeared. Select [2 Double] in display menu window. It is changed to be figure 9.18.6.1 (b).



Fig. 9.18.6.1 (a)

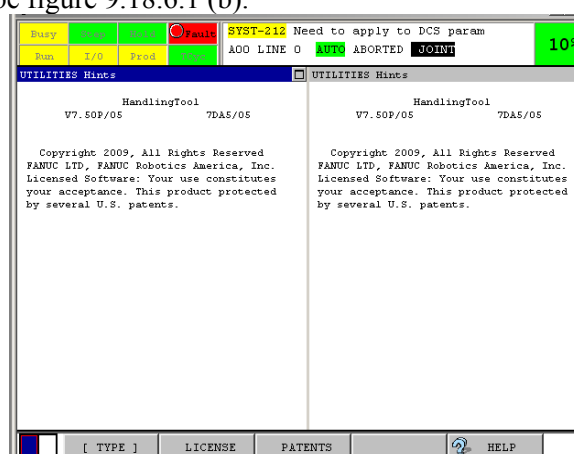


Fig. 9.18.6.1 (b)

- 2 Display system variable \$TP_CURSCRN[1] in window 2
Active window can be switched by pressing display key. Open the second window. After active window is switched in the second window(right window), press menu key [0.NEXT], [6.SYSTEM], F1[TYPE], [SYSTEM Variable] to display the system variable screen in the second window. When \$TP_CURSCRN is selected in system variable screen, \$TP_CURSCRN[1 - 4] are displayed. Then,

select \$TP_CURSCRN[1]. The second window should be figure 9.18.6.1 (c). \$SP_ID is software parts ID and \$SCRN_ID is screen ID of the first window (left window).

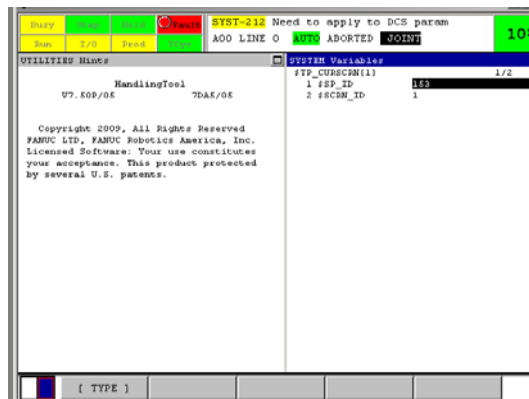


Fig. 9.18.6.1 (c)

3 Make window 1 to aim menu screen

Make active window to the first window and select your target screen in the first window. In this case, file screen is selected. In figure 9.18.6.1(d), the first window is file menu screen, and you can see \$SP_ID is 74 and \$SCRN_ID is 1 in the second window. These ID's are for file menu screen's software parts.

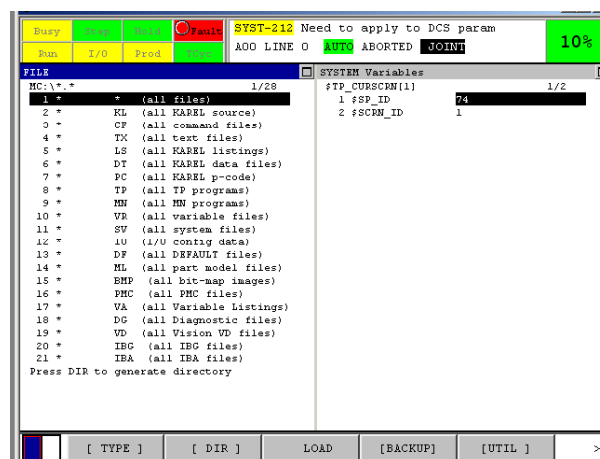


Fig. 9.18.6.1 (d)

9.18.6.2 Basic tag

XML file is composed only of the syntax which is called a tag. The beginning of tag is "<", and end of tag is ">". We explain about XML tag for password function below.

Basic tag must exist in case XML for password function in this controller.

- Starting command tag

```
<?xml version="1.0"?>
```

This tag should be at the top of your XML file

- First tag

```
<PASSWORD>
```

This tag start XML for password function. Only one tag can be specified in your file.

- Last tag

```
</PASSWORD>
```

This tag indicate the end of XML for password function. Only one tag can be specified in your file. So the beginning and end of all XML file should be like below.

```
<?xml version="1.0"?>
  <PASSWORD>
    .
    .
    .
    .
  </PASSWORD>
```

9.18.6.3 Comment tag

This is comment. ALL comments are include within <!-- -->.

```
<!--Password configuration file-->
```

9.18.6.4 Level name tag

This tag defines the level name up to 12 characters. All levels can be renamed including the “INSTALL” level.

```
< LEVEL="3" name="Maintenance" lang="eg"/>
```

- Write “LEVEL” in the beginning of tag and specify level parameter with “=”. 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

- Each level tag defines one level name and one language. In order to define multiple levels or multiple languages, multiple tags are required.

- Specify level name to “name”. For example , “Level 3” is changed to “Maintenance”
- Specify language to “lang”. But it should only be used in a multi-language system. If your controller is not multi-language system (menu select key, [6.SETUP], GENERAL, 2.Current Language is DEFAULT) this parameter is no need. Each language is defined blow. In example, it setup English and Japanese(kanji)

• eg-English	• gr-German
• jp-Japanese(kana)	• sp-Spanish
• kn-Japanese(kanji)	• ch-Chinese
• fr-French	• tw-Taiwanese

9.18.6.5 Local Labels tag

This tag defines the function key access locally within a screen for a given level. All level except INSTALL can be changed to allow/disallow function key access.

The following example is that F4[BACKUP] in file menu screen is disallowed by using local label.

```
<LLABEL level="2" name="[BACKUP]" lang="EG" sp_id="74" scrn_id="1" access="0"/>
```

- Write “LLABEL” in beginning of tag
- Specify level parameter to “level”. The value of “level” and level name is below.

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

- Specify function key’s name to “name”. Function key’s name is different form each language, so you need to specify name for each language. In an example, F4[BACKUP] is disallowed. If you change the name of function key, it is possible to disallow other function key.
- Specify language to “lang”.
- Specify software parts ID and screen ID of the target screen you want to allow/disallow. The way how to look into software parts ID and screen ID of the target screen is described in [1. Software parts ID Screen ID]
- Specify 0 or 1 to ”access”.
 access=0 Function key or pull up entry cannot be pressed
 access=1 Function key or pull up entry can be pressed

9.18.6.6 Screen Access tag

This tag defines the screen access for a given level. All levels except INSTALL can be changed to allow/disallow write access.

```
< SCREEN level="2" sp_id="23" scrn_id="1" access="1" rw_access="0"/>
```

- Write “SCREEN” in beginning of tag
- Specify level parameter to “level”.

- Specify software parts ID and screen ID of the target screen that you want to allow/disallow. The way how to look into software parts ID and screen ID of the target screen is described in [9.18.6.1. Software parts ID Screen ID]
- Specify 0 or 1 to "access".
access=0 Screen cannot be entered
access=1 Screen can be entered
- Specify 0 or 1 to "rw_access".
rw_access=0 If screen can be entered, it will only operate in read-only mode
w_access =1 If screen can be entered, it will only operate in read-write mode

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.

9.18.6.7 Features tag

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

```
< PWD level="2" const="20" access="0"/>
```

- Write "PWD" in the beginning of tag
- Specify level parameter to "level".
- Specify feature to "const". The value of "const" and feature is below.

Table 9.18.6.7 The value of const and the feature

Value of const	Restriction	Description
const=1	KCL	Disallow KAREL command
const=2	I/O configuration	Disallow I/O configuration
const=3~17	Limitation about communication	
const=18	Display Menu	Disallow select menu display
const=19	SHIFT-COORD Menu	Disallow SHIFT-COORD Menu
const=20	Jog Access	Disallow jog access. Prohibits jogging w.p.r but x,y,z is allowed. JGFRM and Group 1 are forced
const=21	Production Speed Access	Prohibits lowering production speed below 80%

- Specify 0 or 1 to "access".
access=0 Access to the feature is not allowed
access=1 Access to the feature is allowed.

9.18.6.8 Default Menu Type tag

This tag defines the default menu type for a given level.

```
<DEFMENU level="3" type="2"/>
```

- Write “DEFMENU” in beginning of tag
- Specify level parameter to “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

The default menu type for all levels in Full menus. All levels except INSTALL can be changed to have a different 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.

9.18.6.9 Default Screen tag

This tag defines the default screen for a given level. The default screen for all levels are not defined.

```
< DEFSCREEN level="0" sp_id="935" scrn_id="1"/>
```

- Write “DEFMENU” in the beginning of tag
- Specify level parameter to “level”.
- Specify software parts ID and screen ID of target screen that you want to allow/disallow. The way how to look into software parts ID and screen ID of the target screen is described in [9.18.6.1. Software parts ID Screen ID]

All levels except INSTALL can be changed to have a default screen. When a level is changed by user logs in, logs out, or time out, then the screen will automatically change to default screen.

For example, in case that the software parts ID is set to 935 and the screen ID is set to 1, the default screen is set to the password screen. Following table describes the software parts ID and Screen ID for typical screen.

Table 9.18.6.9 Software parts ID and screen ID to set default screen

Default screen	Software parts ID	Screen ID
Not defined	0	0
SETUP Passwords	935	1
ALARM	18	1
SELECT	71	1
EDIT	64	1

9.18.6.10 FCTN/Display Menu Access tag

The screen tag can also be used to define the access for each FCTN or Display Menu entry for a given level. All levels except INSTALL can be changed to allow/disallow

```
< SCREEN level="3" sp_id="64" scrn_id="20" access="0"/>
```

- Write “SCREEN” in the beginning of tag
- Specify level parameter to “level”.
- If you want to disallow FCTN menu, you need to specify software parts ID and screen ID of the FCTN menu. But, the way of [9.18.6.1 Software parts ID, Screen ID] cannot be used for FCTN. Software parts ID and screen ID of FCTN menu is below.

Table 9.18.6.10 (a) Software parts ID and screen ID for FCTN menu

FCTN menu	Software parts ID	Screen ID
ABORT(ALL)	20	1
Disable FWD/BWD	20	9
RELEASE WAIT	931	1
QUICK/FULL MENUS	64	20
SAVE	67	1
PRINT SCREEN	171	1
UNSIM ALL I/O	20	2
ENABLE/DISABLE HMI MENUS	1304	2
Diagnostic log	20	12
REFRESH PANE	381	21

- If you want to set Display Menu , Specify software parts ID and screen ID that you want to set Display Menu. But The way [9.18.6.1 Software parts ID, Screen ID] cannot look into. Software parts ID and screen ID of Display Menu is below.

Table 9.18.6.10 (b) Software parts ID and screen ID for Display Menu

Display Menu	Software parts ID	Screen ID
Single	37	101
Double	37	102
Triple	37	103
Status/Single	37	104
EDIT<->EDIT	37	112
Menu History	37	109
User Views	37	107
Menu Favorite	37	108

- Specify 0 or 1 to "access".
 access=0 Access to the feature is not allowed
 access=1 Access to the feature is allowed.

9.18.6.11 EDCMD Access tag

The screen tag can also be used to define the access for each EDCMD entry in the teach pendant Editor.

```
< SCREEN level="3" sp_id="64" scrn_id="30" access="0"/>
```

- Write "SCREEN" in beginning of tag
- Specify level parameter to "level".
- If you want to disallow EDCMD, you need to specify software parts ID and screen ID of EDCMD. But, the way of [9.18.6.1 Software parts ID, Screen ID] cannot used for EDCMD. Software parts ID and screen ID of EDCMD is below.

Table 9.18.6.11 Software parts ID and screen ID for EDCMD menu

EDCMD	Software parts ID	Screen ID
Insert	64	3
Delete	64	4
Find	64	5
Replace	64	6
Renumber	64	7
Copy	64	9
Comment	64	16

EDCMD	Software parts ID	Screen ID
Undo	64	17

9.18.7 Password Log

If the install user enables the "Password log" item on the password setting screen, the password log displays the events below.

- Password event
- Programming event
- File operation event
- SpotTool application event

When user logins, program creations, and other events occur, they are all recorded in the password log. The password log screen contains the item listed in Table "Item on the password log screen".

Table 9.18.7 Item on the password log screen

Item	Explanation
Password Log	Monitors password logins and logouts. If password log is enabled, it is possible to check who is logged in and what changes have been made.

Only the install user can enable the "Log events" setting. For information on enabling the password log, see Procedure 9-19 "Enabling the password log".

All users can display the password log. For information on displaying the password log, see Procedure 9-20 "Displaying the password log".

Procedure 9-19 Enabling the password log

Condition

- The user is logged in as an install user. (Procedure 9-12 "Assignment a user name and a standard password for each password level")

Step

- 1 Press the MENUS key.
- 2 Select "SETUP".
- 3 Press F1 [TYPE].
- 4 Select "Passwords". The screen below appears.

SETUP Passwords		JOINT 10%
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 Use the steps below to enable or disable 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 9-20 Displaying the password log

Condition

- The install user has enabled Log events. (Procedure 9-19 "Enabling the password log")
- The user is logged in at any of the install, program, and setup levels.

- A full menu appears.

Step

- 1 Press the MENUS key.
- 2 Select "ALARM".
- 3 Press F1 [TYPE].
- 4 Select "Password Log". The screen below appears.

Password Log		JOINT 10%
1	PWD -035 Login (BOB) Install from CRT	
2	PWD -031 QUICK MENUS forced	
3	PWD -039 Logout (BOB) Install from CRT	
4	PWD -052 Pwd Timeout (MARY) from Teach	
5	PWD -034 Login (MARY) Install from Te	
6	PWD -031 QUICK MENUS forced	
7	PWD -038 Logout (MARY) Install from T	

- 5 To display the details of a specific event, move the cursor to that event and press F5 "DETAIL". The screen below appears.

DETAIL Alarm		JOINT 10%
PWD -035 Login (BOB) Install from CRT/Keyboard		
Password Log		
1	PWD -035 Login (BOB) Install from CRT	
2	PWD -031 QUICK MENUS forced	
3	PWD -039 Logout (BOB) Install from CRT	
4	PWD -052 Pwd Timeout (MARY) from Teach	
5	PWD -034 Login (MARY) Install from Tea	

9.18.8 Screen Restrictions According to Password Level

For each login level, the screen restrictions below are enabled.

- C = The screen can be displayed, and the data on it can be changed and operated on.
- D = The screen can only be displayed. (No data on it can be changed.)

Table 9.18.8 "Screen restrictions according to password level" lists the screen restrictions according to password level.

Table 9.18.8 Screen restrictions according to password level

Menu	Teach pendant screen	Password level			
		Install	Setup	Program	Operator
UTILITIES	Hints	D	D	D	D
	Prog Adjust	C	D	D	D
	Prog Toolbox	C	C	C	D
	Prog Adjust	C	C	D	D
	Program Shift	C	C	D	D
	Mirror Image Shift	C	C	D	D
	Tool offset	C	C	D	D
	Frame offset	C	C	D	D
	Group Exchg	C	D	D	D
TEST CYCLE	TEST CYCLE	C	C	C	D
MANUAL FCTNS	Manual Weld	C	C	C	C
	Macros	C	C	C	C

Menu	Teach pendant screen	Password level			
		Install	Setup	Program	Operator
ALARM	Alarm Log	C	D	D	D
	Motion Log	C	D	D	D
	System Log	C	D	D	D
	Appl Log	C	D	D	D
	Recovery	D	D	D	D
	Password Log	C	D	D	D
	Comm Log	C	D	D	D
I/O	Cell Interface	C	C	D	D
	Weld Interface	C	C	D	D
	Spot Equip	C	C	D	D
	Digital	C	C	D	D
	Analog	C	C	D	D
	Group	C	C	D	D
	Robot	C	C	D	D
	UOP	C	C	D	D
	SOP	C	C	D	D
	Interconnect	C	C	D	D
	PLC I/O	D	C	D	D
	DeviceNet	C	C	D	D
	PMC	C	C	D	D
	PMC Display	C	C	D	D
SETUP	Cell	C	C	D	D
	Prog Select	C	C	D	D
	Spot Equip	C	C	D	D
	Spot Adv Fctns	C	C	C	D
	Spot Config	C	C	D	D
	TP Hardkeys	C	C	C	D
	Servo Gun	C	D	D	D
	General	C	C	D	D
	Coll Guard	C	D	D	D
	Frames	C	C	D	D
	Macro	C	C	D	D
	Ref Position	C	C	D	D
	Port Init	C	C	D	D
	Ovrd Select	C	C	D	D
	User Alarm	C	C	D	D
	Error Table	C	C	D	D
	iPendant Setup	C	D	D	D
	BG Logic	C	C	D	D
	Resume Tol.	C	D	D	D
	Coord	C	D	D	D
	Stroke Limit	C	D	D	D
	Motion DO	C	D	D	D
	Host Comm	C	C	D	D
	Passwords	C	C	C	C
FILE	File	C	C	C	D
	File Memory	C	C	C	D
	Auto Mackup	C	D	D	D
SOFT PANEL	SOFT PANEL	C	C	C	D
USER	USER	C	C	C	D
SELECT	SELECT	C	C	C	D
EDIT	EDIT	C	C	C	D

Menu	Teach pendant screen	Password level			
		Install	Setup	Program	Operator
DATA	Registers	C	C	C	D
	Position Reg	C	C	C	D
	KAREL Vars	C	C	C	D
	KAREL Posns	C	C	C	D
	Pressure	C	D	D	D
	Manual Bkup	C	D	D	D
	Distance	C	D	D	D
STATUS	Production	D	D	D	D
	Robot ready	C	C	C	C
	Servo Gun	D	D	D	D
	DisableFault	C	C	D	D
	Axis	C	D	D	D
	Version ID	D	D	D	D
	Safety Signal	D	D	D	D
	Exec-hist	C	D	D	D
	Memory	D	D	D	D
	Prg Timer	C	D	D	D
	Sys Timer	C	D	D	D
	Remote Diag	C	D	D	D
	Condition	C	D	D	D
	Appl-status	D	D	D	D
POSITION	POSITION	D	D	D	D
SYSTEM	Clock	C	C	D	D
	Variables	C	C	C	D
	Gun Master	C	D	D	D
	Master/Cal	C	C	D	D
	OT Release	C	C	C	C
	Axis Limit	C	C	D	D
	Config	C	C	D	D
	Motion	C	C	D	D
USER2	USER2	C	C	C	D
BROWSER	Browser	D	D	D	D
DEP	Dep	C	C	C	D
	Display	C	C	C	C
	Status	C	C	C	C
	Program	C	C	C	D

9.18.9 Password Auto Login Function

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 system 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 it 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.
		AUTO	PWD-72 Protected in AUTO mode
	OFF	T1/T2	PWD-71 Protected in TP disable
		AUTO	PWD-72 Protected in AUTO mode
_AUTOLOGIN_1	ON	Not applicable	No restrictions.
	OFF	Not applicable	PWD-71 Protected in TP disable
_AUTOLOGIN_2	Not applicable	T1/T2	No restrictions.
		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"

9.19 MIXED LOGIC INSTRUCTION

Overview

The Mixed Logic Instruction allows the use of various operator and data combinations in assignment statements, relational statements, and wait command statements in TP programs. The mixed logic instruction supports the NOT operator "!" and parentheses "()".

Mixed Logic Instructions can be specified on the [Register] menu, [I/O] menu, [IF/SELECT] menu, and [WAIT] menu.

Mixed Logic Instructions must be specified in parentheses, as shown below.

- DO[1]=(DI[1] AND !DI[2])
- IF (DI[1]) JMP LBL[1]
- WAIT (DI[1])

If not enclosed in parentheses, they are executed in the same way as other operation commands.

The mixed logic instruction now supports Boolean data type variable flags and markers and the new command TC_Online.

The mixed logic instruction enables background operations. Using background operations, it is possible to execute up to eight TP programs in which only assignment statements are taught simultaneously, continuously, and repeatedly.

Data types

Mixed logic instructions can use the data types below.

Table 9.19 (a) Data types

Type	Value	Data
Numeric	Numeric values can be handled as data. Both integer and real numbers can be used.	Register, constant, GI/O, AI/O, position register element, argument, system variable
Boolean	Data can assume either ON or OFF values.	DI/O, RI/O, UI/O, SI/O, WI/O, ON, OFF, flag, marker

**CAUTION**

Position data and palletizing data cannot be used with mixed logic instructions.

Operators

Mixed logic instructions can use the operators below.

Table 9.19 (b) Arithmetic operators

Operator	Operation
+	Addition of the left side and the right side
-	Subtraction of the right side from the left side
*	Multiplication of the left side and the right side
/	Division of the left side by the right side
MOD	Remainder of the division of the left side by the right side
DIV	Integer part of the quotient of the division of the left side by the right side

- The arithmetic operators can be used with numeric data only. If an attempt is made to use arithmetic operators with Boolean data, "INTP-203 Variable type mismatch" is generated.
- The output data of an arithmetic operator is always of numeric type.

Table 9.19 (c) Logical operators

Operator	Operation
AND	Logical product of the left side and the right side
OR	Logical sum of the left side and the right side
!	Logical negation of the left side and the right side

- The logical operators can be used with Boolean data only. If an attempt is made to use logical operators with numeric data, "INTP-203 Variable Type Mismatch" is generated.
- The output data of a logical operator is always of Boolean type.

Table 9.19 (d) Relational operators

Operator	Operation
=	Returns an ON value if the left side is equal to the right side. Otherwise, returns an OFF value.
<>	Returns an ON value if the left side is not equal to the right side. Otherwise, returns an OFF value.
<	Returns an ON value if the left side is less than the right side. Otherwise, returns an OFF value.
>	Returns an ON value if the left side is greater than the right side. Otherwise, returns an OFF value.
<=	Returns an ON value if the left side is equal to or less than the right side. Otherwise, returns an OFF value.
>=	Returns an ON value if the left side is equal to or greater than the right side. Otherwise, returns an OFF value.

- "=" and "<>" can be used with both numeric type data and Boolean type data.
- "<", ">", "<=", and ">=" can be used with numeric data only. If an attempt is made to use them with Boolean data, the error "INTP-203 Variable Type Mismatch" occurs.

The table below indicates the priority of operators.

Table 9.19 (e) Relational operators

Priority	Operator
High	!
	*, /, DIV, MOD
	+, —
Middle	<, >, <=, >=
	=, <>
	AND
Low	OR

Expressions

Mixed logic instructions can be used with assignment statements, conditional branch commands, and wait commands.

Assignment statements

Mixed logic instruction assignment statement examples are given below.

```
R[1] = ((GI[1] + R[1]) * AI[1])
DO[1] = (DI[1] AND (GI[1] = GI[2]))
```

- The leftmost = is for an assignment statement. The other = signs are for relational statements. The result of the expression on the right side is assigned to the data on the left side.
- The output data of an arithmetic operator is always of numeric type.
- If the data on the left side is of Boolean type and the result of the expression on the right side is of numeric type, the data on the left side is OFF if the value on the right side is less than 1 and greater than -1 and is ON if the value on the right side is greater than 1 or less than -1. This operation is the same as that for an ordinary assignment statement.
- If the data on the left side is of numeric type and the result of the expression on the right side is of Boolean type, the data on the left side is 0 if the value on the right side is OFF and is 1 if the value on the right side is ON. This operation is the same as that for an ordinary assignment statement.
- If a real number is assigned to a GO, AO, or integer type system variable, the fractional part is truncated.
- "Pulse" cannot be specified with a mixed logic instruction. To specify "Pulse", an ordinary operation command must be used.
- Position data and palletizing data cannot be specified on the left or right side of a mixed logic instruction. To specify position data or palletizing data, an ordinary operation command must be used.
- The maximum number of items (data items or operators) that can be used in an assignment statement is about 20. The exact maximum number of items that can be used depends on the data type.

The data below can be specified on the left side of an assignment statement.

Table 9.19 (f) Assigning data

Type	Data
Boolean	DO, RO, UO, SO, WO, flag, marker
Numeric	Register, GO, AO, position register element, system variable

Conditional statements

The following shows examples of using mixed logic instructions with conditional branch instructions.

```
IF (R[1] = (GI[1] + R[1]) * AI[1]) JMP LBL[1]
IF (DI[1] AND (!DI[2] OR DI[3])) JMP LBL[1]
```


- A mixed logic instruction expression can be used in the conditional statement of a conditional branch command.
- The result of a conditional statement must be of Boolean type.
- If the result of a conditional statement is ON, the executable statement (JMP LBL, for example) of the conditional branch command is executed.
- If a mixed logic is used in a conditional statement, the statements below can be used as an executable portion of a conditional branch command.

```
JMP LBL[ ]
CALL
MIXED LOGIC STATEMENT
PULSE STATEMENT
```

- A mixed logic assignment statement and a pulse statement can be specified in the executable statement of a conditional branch command only if the conditional statement contains a mixed logic expression. See the examples below.

```
IF (DI[1]), DO[1]=(On)
IF (DI[2]), DO[1]=Pulse
```

- The maximum number of items (data items or operators) in a conditional statement is about 20. The exact maximum number of items depends on the data type.

Wait commands

The following shows an example of using mixed logic commands with a wait command.

```
WAIT (DI[1] AND (!DI[2] OR DI[3]))
```

- A mixed logic expression can be specified in the conditional statement of a wait command.
- The result of a conditional statement must be of Boolean type.
- A wait command waits until the result of the expression becomes ON.
- It is not possible to specify "On+", "Off-", or "ERR_NUM" with a mixed logic command. To specify these, it is necessary to use ordinary operation command.
- The maximum number of items (data items and operators) in the conditional statement of a wait command is about 20. The exact maximum number of items depends on the data type.

Adding mixed logic commands

Editing mixed logic commands is complicated than editing ordinary commands. The reason for this is that they can use various data types and operators and there can be various combinations of them.

The functions below are provided to facilitate editing.

- To start mixed logic command teaching, a statement containing parentheses must be selected first.
- On the item selection menu, the items available according to the location of the statement appear.
- If a combination of items is invalid (for example, there are adjacent operators), an empty item is automatically inserted, prompting the user to select an item.
- When an item is selected, any related items are automatically deleted. For example, if an operator is deleted, the subsequent operated item is deleted at the same time.
- If the cursor is on an item in a mixed logic expression, and the expression is invalid, an error message appears on the prompt line. The message below appears.

Table 9.19 (g) Mixed logic error messages

Error message	Explanation
Parentheses mismatch	The number of opening parentheses is not equal to the number of closing parentheses.

Error message	Explanation
Invalid index	Invalid index number.
Variable type mismatch	The data type does not match the operator.
Invalid parameter name	Invalid system variable name.
Untaught element	There is an un-taught item (...).
Invalid motion group	A specific operation group with a PR[] cannot be used in a program.
Invalid item for output	The item on the left side of the assignment statement is invalid.
Invalid item for Mixed Logic	The item cannot be used in a mixed logic expression.
Syntax error	Invalid statement.

To add a mixed logic command to a program, use the procedure below.

Procedure 9-21 Adding a mixed logic command

Step

- 1 In the [Edit] menu, press [F1(INST)] to move to the [Instruction] menu.
- 2 Select the type of the command to be added (Register, I/O, IF/SELECT, or WAIT).
- 3 From selections, select a mixed logic statement that contains parentheses (...)
- 4 Prepare a command in an ordinary way.
- 5 To change an item in the mixed logic statement, press [F4 (CHOICE)] while the cursor is positioned on that item. Available items appear.
- 6 Except on the left side of an assignment statement, items can be inserted into any mixed logic statement. Press [F1 <INSERT>].
"..." is inserted before the cursor, and an item selection menu appears.
Select an operator, and an empty item is inserted after that operator, and a data menu appears.
- 7 Except on the left side of an assignment statement, items can be deleted from any mixed logic statement. Move the cursor to an item and press [CHOICE]. A <DELETE> key appears in [F2]. If an operator is deleted, the subsequent data item is also deleted.
- 8 Except on the left side of an assignment statement, to add or delete the NOT (!), press [F5 (!)] while the cursor is on a digital I/O item in a mixed logic expression. A negation operator (!) is added or deleted.
- 9 If the right side of an assignment statement contains a mixed logic command, to change the left side of the assignment statement, move the cursor to an item on the left side and press [F4 (CHOICE)]. A menu containing the items that can be specified on the left side of the assignment statement appears.

Background operation

Overview

By using a background operation, it is possible to execute a TP program containing assignment statements only in the background. The program is executed repeatedly from the beginning to the end. This execution is not influenced by emergency stops, holds, or alarms.

A background operation can be in three execution modes, standard mode, high-speed mode and high-level mode. High-level mode has been supported in 7DA5 or later.

- In standard mode, all mixed logic commands can be executed in the background. In each ITP, 300 item are scanned. (One ITP is usually eight milliseconds.) "Item" as used here refer to data items and operators. If there are more than 300 items, scanning time increases. For example, if there are 800 operations, processing time is 24 milliseconds. If programs of high-level mode are executed, number of items can be scanned in each ITP is (300 – number of high-level mode item).
- In high-speed mode, only I/O data and logical operators can be executed. In every eight milliseconds, 8000 item are processed. The maximum allowable number of items in high-speed mode is 8000. In high-speed mode, an indirect specification of an index (for example, DO[R[1]]) is not possible.
- In high-level mode, all mixed logic commands can be executed in the background. Programs of high-level mode can be executed certainly in every ITP. (One ITP is usually eight milliseconds.) In

each ITP, 270 items can be scanned. Please share some statements from standard mode program and execute by high-level mode, if these statements must be execute at high frequency.

For more information on the execution modes for a background operation, see Table 9.19 (h).

To execute a background operation, use the procedure described in "Setting up a background operation".

Table 9.19 (h) Execution modes for a background operation

Mode	Maximum allowable number of items	Scanning time	Available data	Available operators
Standard mode	No restriction	(Number of items in standard mode / 300 – Number of items in high-level mode) * ITP). The number of items as used here refers to the total number of items in each mode background operation programs. One ITP is usually eight milliseconds.	F[], M[]*, DI[], DO[], AI[], AO[], GI[], GO[], SI[], SO[], UI[], UO[], RI[], RO[], WI[], WO[], ON, OFF R[], PR[i.j], AR[]	(,), !, AND, OR, =, <>, <, <=, >, >=, +, —, *, /, DIV, MOD
High-level mode	270	Eight milliseconds	Constant Parameter Timer, timer overrun JMP(7DA5 or later) LBL(7DA5 or later)	
High-speed mode	8,000	Eight milliseconds	F[], M[]*, DI[], DO[], SI[], SO[], UI[], UO[], RI[], RO[], WI[], WO[]	(,), !, AND, OR

M[] cannot be specified on the left side of an assignment statement in a background operation.

- Up to eight programs can be executed at the same time as a background operation.
- If the program contains other than assignment statements, "INTP-443 Invalid item for Mixed Logic" is generated at execution time.
- While a program is being executed in the background, the program cannot be edited or executed as an ordinary task. If the program is not executed in the background, the program can be executed as an ordinary task.
- If a program is being executed in the background, a program with the same name cannot be loaded in overwrite mode.
- If, when the power is interrupted, a program is executed as a background operation, the program is automatically executed in the same execution mode the next time the power is turned on.
- The execution of a background operation is started before the execution of an ordinary program. The execution of a background operation takes one millisecond per ITP. The execution of a background operation may influence the cycle time of an ordinary program.

- To reduce the execution time of a background operation, reduce the value of \$MIX_LOGIC.\$ITEM_COUNT. \$MIX_LOGIC.\$ITEM_COUNT defines the number of items to be processed per ITP. The standard value is 300.
- In a background operation, assignment statements with conditional expressions can be executed. If the result of a conditional expression is OFF, the corresponding assignment statement is not executed. In the example below,

IF (DI[1]), DO[1]=(DI[2])

If DI[1] is ON, DI[2] is assigned to DO[1]; and if DI[1] is OFF, DO[1] is not changed.

In high-speed mode, conditional expressions cannot be executed and, therefore, programs containing conditional expressions are always executed in standard mode.

- In a background operation, pulse commands can be used. By combining a pulse command with a conditional expression, it is possible to create a off delay timer. An example is given below.

IF (DI[1]), DO[1]=Pulse 1sec

If DI[1] remains ON for longer than one second, DO[1] is ON for one second. If DI[1] becomes OFF before one second elapses, DO[1] becomes OFF immediately. While DI[1] is OFF, this statement does not set DO[1].

In high-speed mode, pulse commands cannot be executed and, therefore, programs containing pulse commands are always executed in standard mode.

To keep DO[1] ON for one second even if DI[1] becomes OFF within one second, use the statement below.

F[1]=(DI[1] OR (F[1] AND DO[1]))

IF (F[1]), DO[1]=Pulse 1sec

- If no time is specified with a pulse command, one scan pulse is assumed in the execution of a background operation. An example is given below.

IF (DI[1]), DO[1]=Pulse

In this case, if DI[1] is changed from OFF to ON, DO[1] becomes ON for one scan. This can be used for edge detection.

- If executed in an ordinary way, a time-less pulse command uses the time set for \$DEFPULSE, but if the command is executed in the background, a different time is used.
- JMP statement has been supported in 7DA5 or later. In or before 7DA4, if JMP statements are used in background program, the program cannot start and alarm (INTP-443 Invalid item for Mixed Logic) is occurred.
- Target LBL of JMP statement must be set at line after the JMP statement. If the target LBL is before JMP, the program cannot start and alarm (INTP-443 Invalid item for Mixed Logic) is occurred.
- When programs included JMP statement is executed in background logic, scan time of background program may be changed by branch of program execution.

- If number of items in high-level mode is increased, standard mode requests longer scan time.
- If number of high-level mode item exceeds maximum number, alarm (INTP-651 High level mode programs too big) is occurred and the background program stops. In this case, please decrease number of high-level mode item.
- High-Level mode has been supported in 7DA5 or later. In or before 7DA4, high-level mode cannot be used.

Using the [Background Logic] screen, it is possible to set up and execute a program as a background operation. For information on each of the items on the [Background Logic] screen, see Table 9.19 (i). Table 9.19 (j) lists the operations on the [Background Logic] screen.

Table 9.19 (i) Items on the [Background Logic] screen

Item	Explanation
[PROGRAM]	Enter the name of the program to be executed as a background operation.
[STATUS]	The statuses of the background operation programs appear. • [Stop]: The program is stopped. • [Running]: The program is executed in standard mode. • [Running(Fast)]: The program is executed in high-speed mode. • [Running(High)]: The program is executed in high-level mode.
[MODE]	This item is used to select an execution mode. • [Auto]: Executes the program in high-speed mode if it can be executed in high-speed mode. Executes the program in standard mode if it cannot be executed in high-speed mode. • [Fast]: Always executes the program in high-speed mode. If the program contains any item that cannot be used in high-speed mode, the error message "INTP-444 Invalid item for fast mode" is generated at execution time. • [Normal]: Always executes the program in standard mode. • [High]: Always executes the program in high-level mode. The standard value is [Auto]. If the program is to be executed in high-speed mode but is executed in standard mode, set this item to [Fast]. The numbers of the lines invalid in high-speed mode are indicated in error messages.

Table 9.19 (j) Operations on the [Background Logic] screen

Function key	Explanation
[RUN]	Press this key to execute a program as a background operation. An error occurs if the program contains any statement that cannot be used in a background operation.
[STOP]	Press this key to stop the execution of a program in the background.

Procedure 9-22 Setting up a background operation

Step

- 1 Press [MENUS].
- 2 Press [6 (SETUP)].
- 3 Press [F1 (TYPE)].
- 4 Select [BG Logic]. The screen below appears.

Background logic		JOINT 10%
Normal mode scan time: 8 msec		1/8
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 Enter the name of the program to be executed as a background operation in the [PROGRAM] column. In the [STATUS] column, the statuses of background operation programs appear.
- [Stop] : The program is stopped.
 - [Running] : The program is executed in standard mode.
 - [Running(Fast)] : The program is executed in high-speed mode.
 - [Running(High)] : The program is executed in high-level mode.
- 6 In the [MODE] column, select a mode.
- [Auto] : Executes the program in high-speed mode if it can be executed in high-speed mode. Executes the program in standard mode if it cannot be executed in high-speed mode.
 - [Fast] : Always executes the program in high-speed mode. If the program contains any item that cannot be used in high-speed mode, the error message "INTP-444 Invalid item for fast mode" is generated at execution time.
 - [Normal] : Always executes the program in standard mode.
 - [High] : Always executes the program in high-level mode.

**CAUTION**

The standard value is [Auto]. If the program is to be executed in high-speed mode but is executed in standard mode, set this item to [Fast]. The numbers of the lines invalid in high-speed mode are indicated in error messages.

- 7 To execute the program as a background operation, press [F2 (RUN)].
- 8 To stop the program in the background, press [F3 (STOP)].

Flags

A flag (F[]) is an internal I/O port that can be read and set up. A flag is not connected to an actual I/O device, and is similar to a Boolean type variable.

The value of a flag can be recovered with the power failure recovery function (hot start). This is true of other output ports (DO, for example).

All flags are set to OFF in the events below.

- Cold start
- Control start
- INIT start
- Hot start with the I/O allocation being changed
- Hot start with the I/O configuration being changed

By setting DI, DO, UI, UO, GI, and GO as those with rack 34, slot 1, start points 1 to 1024, they can be allocated to flags.

**WARNING**

If a flag is allocated to UI, dedicated signals are controlled by changing the value of the flag from the teach pendant and a program, thereby making it possible to start the program and perform other operations. If allocating a flag to UI, design the system by satisfactorily considering safety. Otherwise, unit damage and personal injury may result.

To display the [Flag] screen, use the procedure below.

- 1 Press [MENUS].
- 2 Press [5 (I/O)].
- 3 Press [F1 (TYPE)].
- 4 Select [Flag]. The screen below appears.

Flag			JOINT 10%
# STATUS			1/1024
F[	1]	ON	[]
F[	2]	OFF	[]
F[	3]	OFF	[]
F[	4]	ON	[]
F[	5]	ON	[]
F[	6]	ON	[]
F[	7]	ON	[]
F[	8]	ON	[]

On this screen, the values of the flags can be changed.

- 5 To display the [Port Detail] screen, press [F2 (DETAIL)]. The screen below appears.

Outputs		JOINT 10%
Port Detail		1/1
Flag	[1]	
1 Comment : []		

On this screen, the comment on a flag can be changed.

Edge detection, counter, and timer examples with a high-speed mode background operation

Edge detection, counter, and timer examples with a high-speed mode background operation

In high-speed mode, processing speed has improved over that in standard mode and, therefore, does not influence the processing speed of standard program execution. Note, however, that in high-speed mode, numeric operations and pulse commands cannot be used.

Example 1: Edge detection

The program below detects an edge of DI[1]. Only if DI[1] is changed from OFF to ON, DO[1] becomes ON.

```
1: DO[1]=(DI[1] AND !F[1])
2:F[1]=(DI[1])
```

F[1] contains the value of DI[1] obtained in the previous scan. If DI[1] is ON and the previous value of DI[1] is OFF, DO[1] becomes ON.

Example 2: Counter

A example of a DI[1] edge 4-bit counter is given below. The counter value is set in F[41] to F[44]. If GI[1] is allocated as that with rack 34, slot 1, start point 41, and number of points 4, the counter value can be read from GI[1].

```

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[2] AND !F[3]) OR (!F[2] 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]) ;

```

DI[1] for edge detection and F[2] on the first two lines become ON in the first scan if DI[1] is changed from OFF to ON.

Lines 4 to 6 calculate the first bit of the counter (F[41]). F[3] is a working variable for retaining the original value. On line 5, if F[2] is ON, F[41] is inverted, and if F[2] is OFF, F[41] is not changed. On line 6, if the original value of F[41] is OFF, F[2] becomes OFF, which means that no carry over has taken place.

Lines 8 to 10 are for the second bit (F[42]), lines 12 to 14 are for the third bit (F[43]), and lines 16 to 18 are for the fourth bit (F[44]).

Example 3: Timer

In high-speed mode, scan time is always eight milliseconds and, therefore, a timer can be created using a counter. The following shows an example of 80-millisecond pulses. The operation of this program is the same as that of "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 used as a working variable for retaining the value of DI[1].

F[2] is a working variable. If F[2] is ON, the counter increments.

If 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 that the counter does not increment until DI[1] becomes OFF.

Markers

On the [Marker] screen, the statuses of markers can be monitored.

A marker (M[]) is similar to a flag, but the value of a marker cannot be edited directly. If a marker (M[]) is specified on the left side of an assignment expression (=) in a TP program, and the statement is executed, the expression is internally defined as a background operation and is executed repeatedly. The marker (M[]) always contains the result of the expression.

As standard, the marker function is disabled and the [Marker] menu does not appear, so that M[] cannot be taught in a 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 an ordinary TP program (not in a background operation), M[1] always contains the result of the expression on the right side. If both DI[1] and DI[2] are ON, M[1] is ON; otherwise, M[1] is OFF.

- If the marker assignment statement is executed in an ordinary TP program, the statement is registered as a background operation. The statement is executed as a background operation until another expression redefines the marker. Because the statement is a background operation, the execution of the statement is not stopped even if the program is paused or stopped.
- As standard, there are eight markers (M[1] to M[8]). The number of markers can be changed with the system variable "\$MIX_LOGIC.\$NUM_MARKERS". The maximum allowable number of markers is 100. One marker occupies a permanent memory area of 300 bytes.
- The scan time in which to calculate marker assignment statements is the same as that in a background operation in standard mode. If marker assignment statements are defined, the scan time in a background operation is influenced.
- To stop calculation, clear the defined marker expression. To clear a defined expression, either execute an [CLEAR] operation on the [Marker detail] menu or execute the line below in a TP program.

```
M[1]=()
```

- If a marker is not allocated in an expression, and the marker is used in another statement, "INTP-438 I/O Read I/O value failed" is generated at the time when the marker value is read.
- It is not possible to specify M[] on the left side of an assignment statement in a background operation.

To display the [Marker] screen, use the procedure below.

- 1 Press [MENUS].
- 2 Press [5 (I/O)].
- 3 Press [F1 (TYPE)].
- 4 Select [Marker]. The screen below appears.

Marker			JOINT 10%	
	#	STATUS	1/8	
M [	1]	ON	[	]
M [	2]	OFF	[	]
M [	3]	OFF	[	]
M [	4]	ON	[	]
M [	5]	ON	[	]
M [	6]	ON	[	]
M [	7]	ON	[	]
M [	8]	ON	[	]

On this screen, the values of the markers can be changed.

- 5 To display the [Port Detail] screen, press [F2 (DETAIL)]. The screen below appears.

Port Detail		JOINT 10%
Marker	[1]	
Comment:	[]	
Expression :		
M[1]=((DI[1] OR DI[2]) AND !DI[3])		
Monitor:		
M [1]	ON	
DI[1]	OFF	
DI[2]	ON	
DI[3]	OFF	
DI[4]	OFF	
DI[5]	ON	

On this screen, the comment on a marker can be changed.

- On the [Port Detail] screen, defined expressions appear. To clear a defined expression, press [F5 (CLEAR)]. When the message below appears,

Clear expression?

Press [F4 (YES)].

- The current values of all the data items in the defined expressions are displayed in the Monitor area.
- On this screen, the comment on a marker can be changed.

TC_ONLINE

TC_ONLINE is similar to a marker. The TC_ONLINE command defines an expression, which is, in turn, calculated as a background operation. While the result of the expression remains OFF, the execution of all TP programs and KAREL programs having motion groups is stopped.

As standard, the TC_ONLINE function is disabled, and the [TC_ONLINE] menu does not appear, so that TC_ONLINE commands cannot be taught in a TP program. To use the TC_ONLINE function, set \$MIX_LOGIC.\$USE_TCOL to TRUE.

The following shows an example.

TC_ONLINE (DI[1] AND DI[2])

While DI[1] or DI[2] remains OFF after this line is executed, all TP programs are stopped. See Table 9.19 (k).

Table 9.19 (k) TC_ONLINE commands

TC_ONLINE (...)	Defines a specified mixed logic command as a TC_ONLINE expression and enables TC_ONLINE.
TC_ONLINE DISABLE*	Disables TC_ONLINE. If TC_ONLINE is disabled, the TP program is not stopped with TC_ONLINE (...).
TC_ONLINE ENABLE*	Enables TC_ONLINE. It is used to enable TC_ONLINE that has been disabled with TC_ONLINE DISABLE.

* As standard, TC_ONLINE DISABLE and TC_ONLINE ENABLE cannot be used. To use these commands, set \$MIX_LOGOC.\$USE_TCOLSIM to FALSE.

- When TC_ONLINE (...) is executed, the specified expression is defined as a TC_ONLINE expression.

While TC_ONLINE is enabled, the defined expression is calculated as a background operation.

While the result of the expression is OFF, all TP programs and KAREL programs having motion groups are stopped.

- If a program is started while the result of the TC_ONLINE expression is OFF, the program is immediately paused. Except [Shift] + [BWD], starts of all types are influenced by TC_ONLINE.
- The only case in which a program can be executed when the result of the TC_ONLINE expression is OFF is when the program is executed backward using [Shift] + [BWD].
- Even if the scan time in a background operation is longer than an ITP, TC_ONLINE expressions are calculated in each ITP. (One ITP is usually eight milliseconds.)
- A program that does not have a motion group and for which "ignore pause" is enabled is not paused even if the result of the TC_ONLINE expression is OFF.
- If \$MIX_LOGIC.\$USE_TCOLSIM is TRUE (standard), TC_ONLINE DISABLE and TC_ONLINE ENABLE cannot be taught on the [Edit] menu. It is necessary to change the TC_ONLINE ENABLE/DISABLE setting on the [TC_ONLINE] menu. In this case, when the execution of an operation statement is completed, TC_ONLINE is automatically enabled. This means that TC_ONLINE is disabled only in those brief cases in which the operator operates the robot.
- If \$MIX_LOGIC.\$USE_TCOLSIM is FALSE, TC_ONLINE DISABLE and TC_ONLINE ENABLE can be taught on the [Edit] menu. The TC_ONLINE ENABLE/DISABLE setting is not automatically changed even after the execution of an operation statement is completed.

To add a TC_ONLINE command, use Procedure 9-23. To display the [TC_ONLINE] menu, use Procedure 9-24.

Procedure 9-23 Adding a TC_ONLINE command

Step

- 1 On the [Edit] menu, press [F1 (INST)] to move to the [Instruction] menu.
- 2 Select [TC_ONLINE].

If \$MIX_LOGIC.\$USE_TCOLSIM is **TRUE**, select items and create statements in an ordinary way.

If \$MIX_LOGIC.\$USE_TCOLSIM is **FALSE**, follow the steps:

- a Select [1(...)].
 - b Select items and create statements in an ordinary way.
- 3 To change a TC_ONLINE statement, press [F4 (CHOICE)] at the first "(" of the TC_ONLINE statement, and select [2 (ENABLE)]].

Procedure 9-24 Displaying a TC_ONLINE command

Step

- 1 Press [MENUS].
- 2 Press [5 (I/O)].
- 3 Press [F1 (TYPE)].
- 4 Select [TC_ONLINE]. The screen below appears.

TC_ONLINE		JOINT 10%
		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	

The "Enable" line indicates whether TC_ONLINE is currently enabled. This item can be changed on this menu.

In the "Status" line, the status of TC_ONLINE is indicated. This is the result of a defined expression.

The defined expression is indicated in the Expression space.

- 5 To clear the defined expression, move the cursor to [Monitor].
 - a Press [F5 (CLEAR)]. The prompt below appears.

Clear expression?

- b Press [F4 (YES)].

Saving/Loading

All data for mixed logic commands are saved as described below.

- Mixed logic commands are saved to TP files, together with programs.

- Background operations are saved to TP files.
- Settings on the "Background Logic" menu are saved to MIXLOGIC.SV. In MIXLOGIC.SV, the values of the system variables are saved.
 - \$MIX_LOGIC
 - \$MIX_BG
 - \$MIX_MKR
 - \$DRYRUN
 - \$DRYRUN_PORT
 - \$DRYRUN_SUB
- Comments on flags and markers are saved to DIOCFGSV.IO.
- If DI/O, UI/O, or GI/O are allocated to flags , their allocations are saved to DIOCFGSV.IO.

9.20 PMC MONITOR FUNCTION

The PMC monitor function enables the program of the built-in PMC to be displayed on the teach pendant.

* The built-in PMC is an option.

PMC ladder display

It is possible to display a PMC ladder program using the procedure below.

- 1 Press Screen select (menu).
- 2 Select I/O.
- 3 Press F1 [TYPE].
- 4 Select PMC Display and press the ENTER key.

A screen show as that shown in the figure below appears.

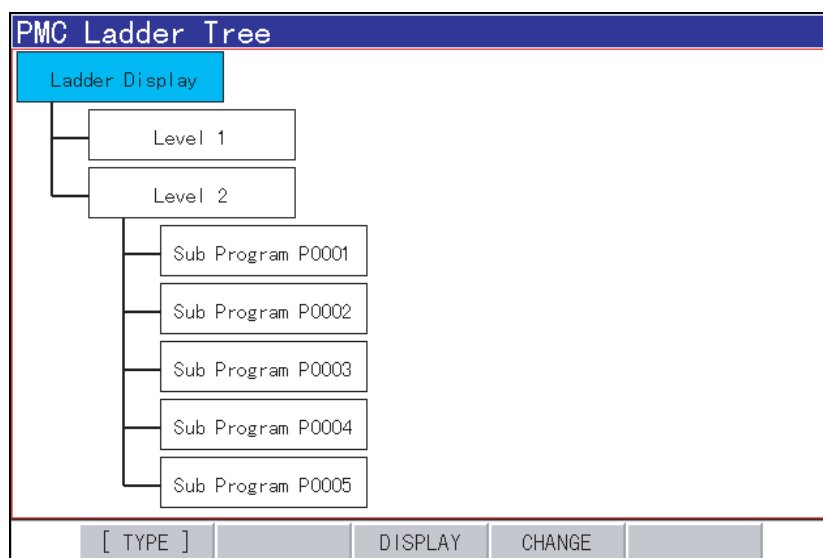


Fig. 9.20 (a) PMC display

- 5 Use the ↑ or ↓ key to move the cursor to the desired level or sub-program. Press F3 [DISPLAY] or the input key, and a screen such as that shown in Fig. 9.20 (b) appears.

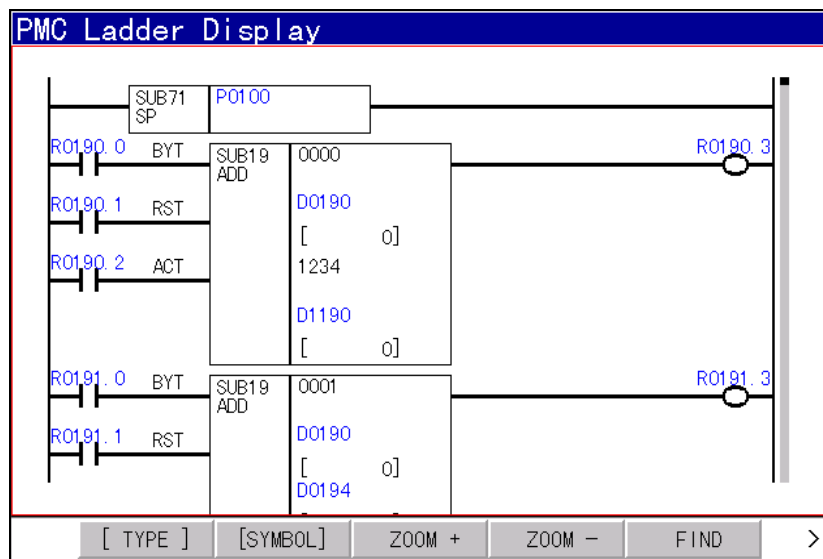


Fig. 9.20 (b) PMC sub-program



CAUTION

If a specific level or sub-program is selected, the selected program is displayed, but it is possible to display all programs by scrolling.

Switching between address and symbol displays

By pressing F2 [SYMBOL] and selecting one from 1 to 4 below, it is possible to switch between address and symbol displays.

1	Address	-	Displays only addresses above relays or coils. (Example: X1000.2)
2	Symbol	-	Displays only symbols above relays or coils. (Example: U13)
3	Address & symbol	-	Displays addresses above relays or coils and symbols below them.
4	Symbol & address	-	Displays symbols above relays or coils and addresses below them.

Address and function command search

- 1 Press F5 [FIND] and a screen such as that shown in Fig. 9.20 (c) appears.

Fig. 9.20 (c) Search screen

The following explains how to use the menu.

- 1 Select Search kind and press the ENTER key, and a popup window appears. Select which of an address or function command to search for.
- 2 Move the cursor to the Search string field and press the ENTER key, and a popup keyboard appears. Input the character string to search for.
- 3 Use the Select button to specify a search start position. The search starts at the selected level or sub-program.
- 4 By selecting Wildcard search (a check mark is placed), it is possible to make a wildcard search, using a wildcard (*) in the string to search for (for example, F1008.*). Only one wildcard character can be used.
- 5 If a search is made by selecting Search result list display (a check mark is placed), a list of all nets containing the search string is created. If a check mark is not placed, a net containing the search string appears. Press F5 [FIND] to search for another net containing the search string.
- 6 If the cursor is moved to the location in which to specify a search range, the selections below can be used.
 All - All relays, coils, and function commands
 Select - Selected relays, coils, and function commands
 Write coils - Coil only
- 7 Press F3 [CANCEL] or the [PREV] key to return to the PMC ladder display screen.
- 8 Press F4 [CLEAR] and the search string is deleted.
- 9 Press F2 [NEXTSRCH] to start a search and display the results.

9.21 PMC EDIT FUNCTION

The PMC edit function enables editing of PMC ladder programs on *iPendant*. It enables editing of relay, coil, and function instructions in all levels and all subprograms of the PMC ladder programs located on the controller.

It enables changing of addresses, symbols, and all nets in a ladder program. The edit screen can be used from all three window frames of *iPendant*, and can also be used via remote communication by accessing CGTP.HTM or ECHO.HTM. By protecting the edit function with a password, it is possible to make it unusable via remote communication. Like the PMC display function, this function enables address and symbol searching.

The PMC edit function does not permit the following:

- Addition and deletion of nets
- Addition and deletion of relays, coils, and connection lines
- Changing of parameter #1 of the function instruction COD
- Changing of parameters #1 and #2 of the function instruction CODB
- Changing of parameter #1 of the function instruction COM
- Changing of the parameters of the function instructions SP, SPE, JMP, JMPE, JMPB, LBL, JMPC, COME, CALL, and CALLU

Using the PMC edit function

- 1 Press the menu.
- 2 Select I/O.
- 3 Press F1 [TYPE].
- 4 Select PMC Display, and a screen such as that shown in Fig. 9.21 (a) appears.

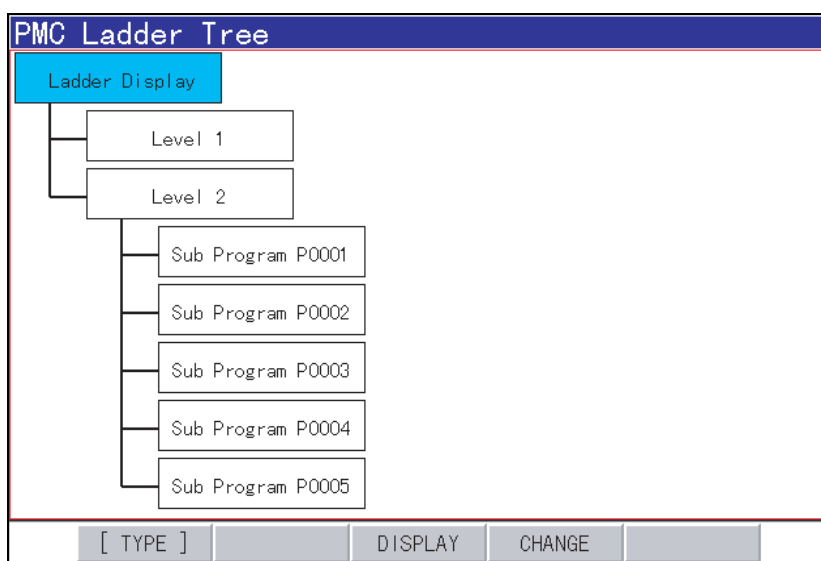


Fig. 9.21 (a) PMC ladder tree

For example, when a ladder program at level 1 or 2 is selected, ladder components are displayed.

- 5 Use the ↑ or ↓ key to move the cursor to the level or subprogram to be displayed and press the F4 [CHANGE] or ENTER key, and a screen such as that shown in Fig. 9.21 (b) appears.

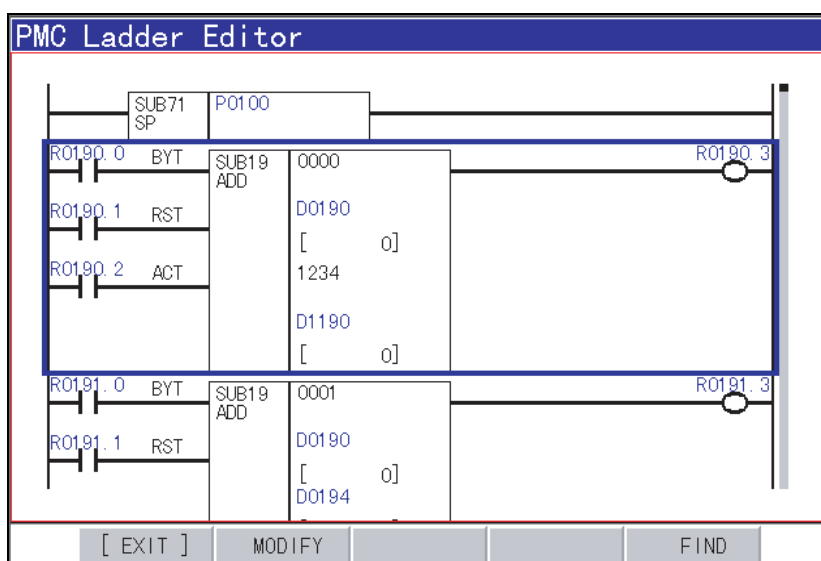


Fig. 9.21 (b) PMC edit screen

⚠ CAUTION

It is possible to start at a specific level or subprogram by selecting it, but the cursor passes through all ladder programs and moves down.

- 6 The cursor changes to a blue one with a 2-pixel width, enclosing a net. The cursor can be used with the arrow keys.

⚠ CAUTION

To search for an address or a function instruction, refer to the section on address and function instruction search.

To edit a net, position the cursor on the next and press the F2 [MODIFY] or ENTER key. A screen such as that shown in Fig. 9.21 (c) appears.

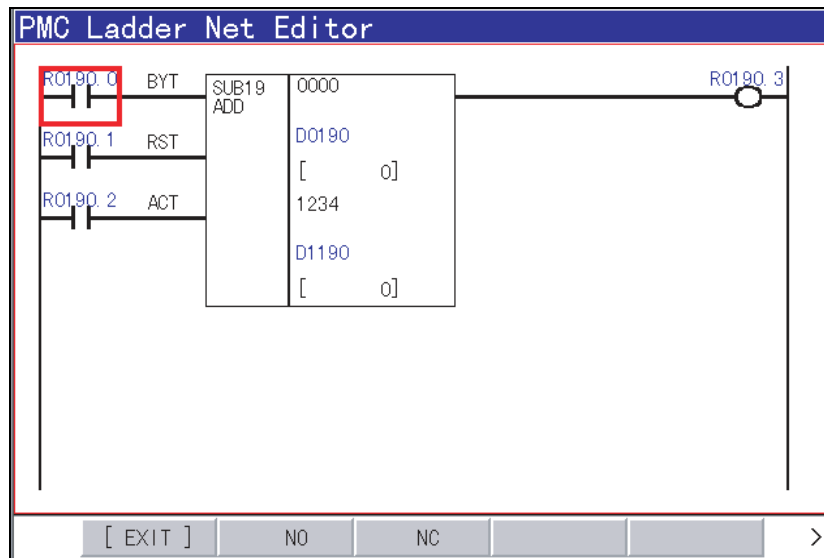


Fig. 9.21 (c) PMC ladder net editing

- 7 To set an element on contact A, press F2 [NO].
- 8 To set an element on contact B, press F3 [NC].
- 9 To modify a parameter of a function instruction, move the cursor to the upper left corner of the function instruction and press the ENTER key. The function instruction parameter screen appears. Use the arrow keys to position the cursor and press the ENTER key, so that parameters can be edited.
- 10 When the cursor is at the rightmost position, the following can be performed:
 - Press F2 [WRT] to write a coil.
 - Press F3 [WRTNT] to write an inverted coil.
 - Press F3 [SET] to set a coil.
 - Press F4 [RESET] to reset a coil.
- 11 To change a specific address or symbol, position the cursor on a relay and press the ENTER key. The PMC connection setup screen appears, enabling editing of addresses and symbols.
- 12 When editing is finished, press F1 [EXIT] and select SUBMIT (which transfers the changes to the editor buffer). To cancel the changes, press F1 [EXIT] and select CANCEL. To continue editing, select CONTINUE.
- 13 When editing of all necessary nets is finished, press F1 [EXIT] on the PMC ladder edit screen and select SUBMIT. The PMC ladder tree screen reappears, and the changes made to the PMC ladder program are saved.

9.22 OPERATION LOG BOOK

9.22.1 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.

To use function it is necessary to order the Operation Log Book option (J695).

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 above. 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. See following examples.
 - Selecting a link on a web page
 - Operations in screens for *i*Pendant only. For example, operations in browser, panel setup, *i*Pendant setup screen.
- 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. Refer to Subsection 9.22.4.
- ‘y’ of “x” is selected in ‘y’ window” is based on the 1st line of prompt window. If the 1st line is blank, y contains nothing. Please refer to screen image for analysis.
- 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.
- “x” is saved” and “x” is loaded” may be recorded by internal process.

9.22.2 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 “9.22.4 Operations” later in this section.

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
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

Message	Event	Parameter (Default)	Save Screen (Default)
'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.		
	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
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

Message	Event	Parameter (Default)	Save Screen (Default)
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
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

9.22.3 Setting Up Book

A system can have 16 books at a maximum. User can setup each book by system variable. Followings are items user can setup.

- Title
- Size
- Memory to store buffer
- Visible/invisible

System variable name	Type	Default	Range	Description
\$LOG_BUF[1-16]. \$TITLE	String	[1-16] "	Max 13 chars	Title of every buffer. This string is displayed on the top line of LogBook menu. It also be displayed in pop up menu of F2([BOOK]) in LogBook menu. If the value is "", the default title is displayed. Default title of BOOK1 is 'Operation'. Default title of 'BOOK2 is 'I/O'. Default title of BOOK3 is 'BOOK 3'
\$LOG_BUF[1-16]. \$SIZE	Integer	[1] 50 [2] 100 [3-16] 0	0, 2147483647	Size of the buffer to save log. Unit is K Bytes. One record takes about 300 bytes. If this value is too big, maximum available size of Log Book buffer is allocated automatically. If there is not enough memory to make Log book buffer, SYST – 188 WARN "book(%d) was not created" is posted and the book is not created. Cycle power is needed to enable the change of this variable.
\$LOG_BUF[1-16]. \$MEM_TYPE	Integer	[1] 0 [2] 1 [3-16] 0	0, 2147483647	Memory type of every buffer. 0: In SRAM, Log is kept at power down. 1: In DRAM, Log is cleared at power down.
\$LOG_BUF[1-16]. \$VISIBLE	Boolean	[1] TRUE [2] FALSE [3-16] TRUE	FALSE, TRUE	If FALSE, the buffer is not displayed in pop up menu of F2([BOOK]) in LogBook menu.

9.22.4 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                                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

```

- 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%
Operation      (20.0k)                  1/790
*SHIFT,F5(TOUCHUP) is pressed, line 2/3
02-JUN-11 14:17:36
--Screen image -----
PNS0001          LINE 0          \T2\ ABORTED
PNS0001\//////////\JOINT\10%\
                                   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                                JOINT 10%
Operation      (20.0k)                  5/790
SRV0-001 Operator panel E-stop

02-JUN-00 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.

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.

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

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.

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
```

9.22.5 Extended Alarm Log

9.22.5.1 Setup

This example name BOOK3 as “ALARM” and use it for only for record of alarms. This configuration uses PERM memory of 500kbytes. The number of alarms that can be stored in the book depends on contents of alarms. Suppose user alarm with undefined alarm message occurred. The book can record the alarm about 5000 times.

NOTE

Because size of book is large, please check the rest of PERM memory before using this setting. The rest of PERM memory should be more than 150KB after the book is established. If the book uses 500KB, the rest of PERM memory should be more than 650KB(500KB + 150KB).

System variable

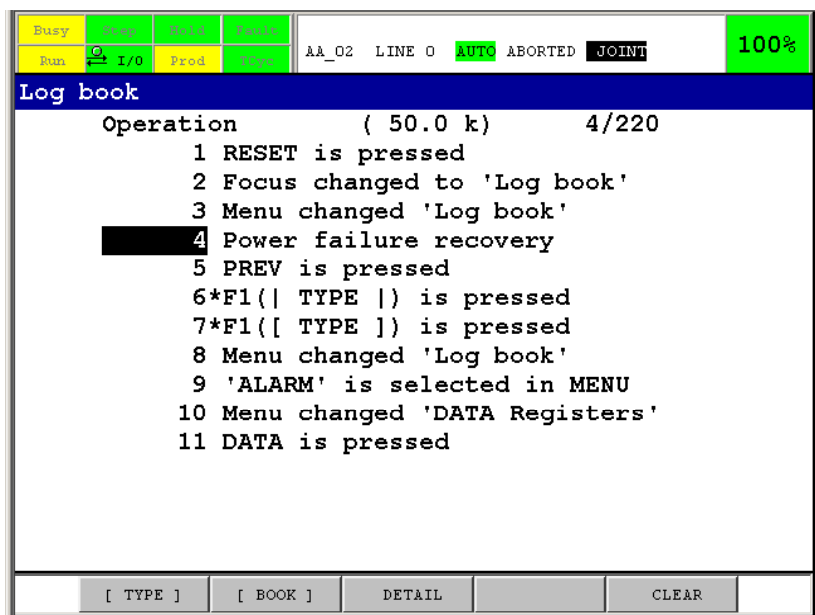
System variable	Type	Default value	Value	Comment
\$LOG_BUF[3].\$TITLE	String	“	ALARM	Title of book
\$LOG_BUF[3].\$SIZE	Integer	0	500	Size of book is 500kbytes. The number of alarms that this book can store depends on contents of alarms.

System variable	Type	Default value	Value	Comment
\$LOG_BUF[3]. \$MEM_TYPE	Integer	0	0	Buffer of this book is made on permanent pool. Cycle power doesn't clear records.
\$LOG_BUF[3]. \$VISIBLE	Boolean	TRUE	TRUE	The book of "ALARM" is displayed in F2 ([BOOK]) pull up menu. Please note that this book is not always displayed as the 3 rd item on the pull-up menu
\$LOGBOOK. \$LOG_ER	Integer	1	3	Alarms are recorded in book3.
\$LOG_ER_SEV	Integer	6	7	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. 7 = warn, pause and abort errors are logged (All alarms are recorded).

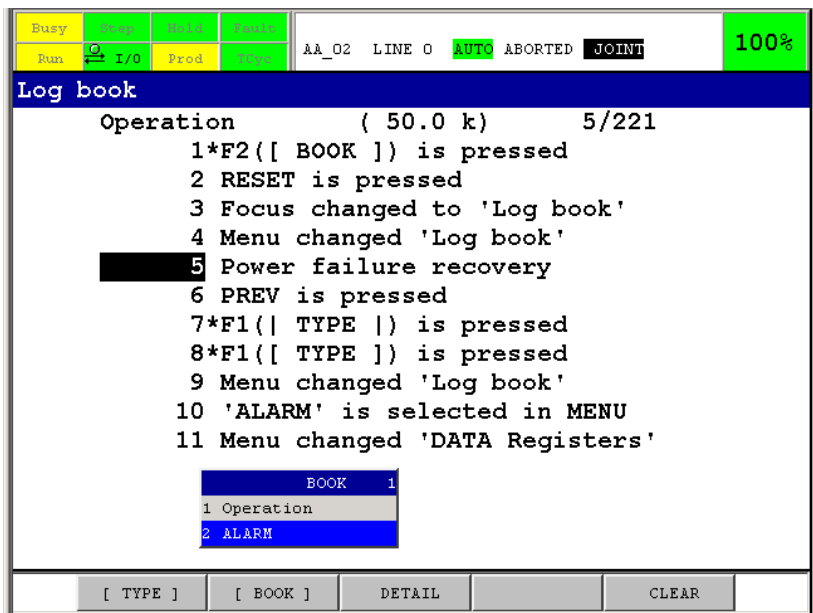
Cycle power is needed to enable these changes.

9.22.5.2 How to display alarm log

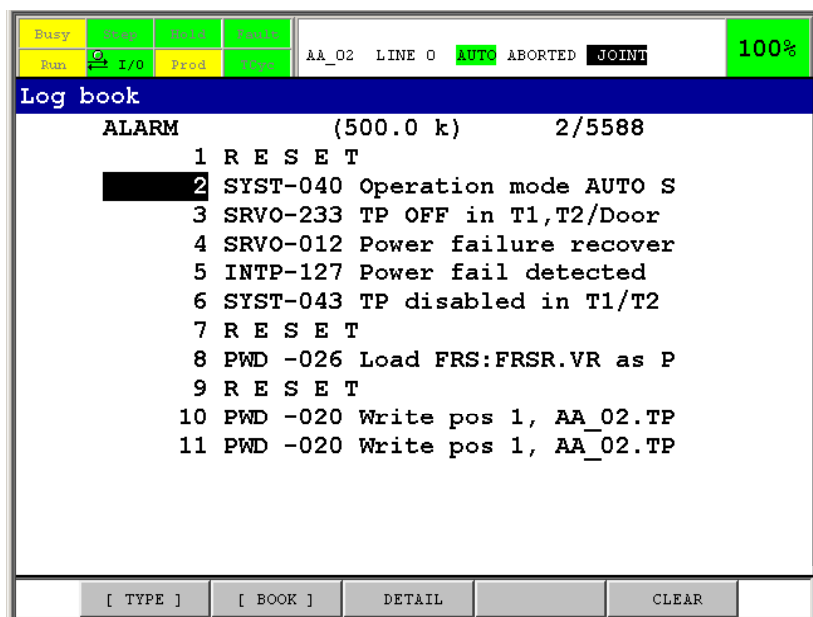
- 1) Press MENU
Select ALARM
Press F1 and select "Log book". Logbook screen is displayed



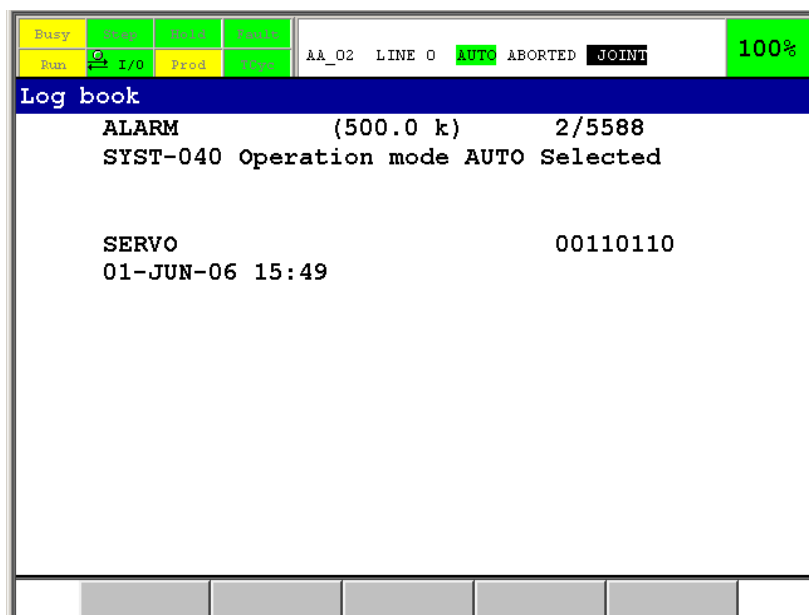
- 2) Press F2 and select “ALARM”



- 3) Contents of book “ALARM”, which is made by setting of this example, is displayed.



- 4) F3 shows detail screen.

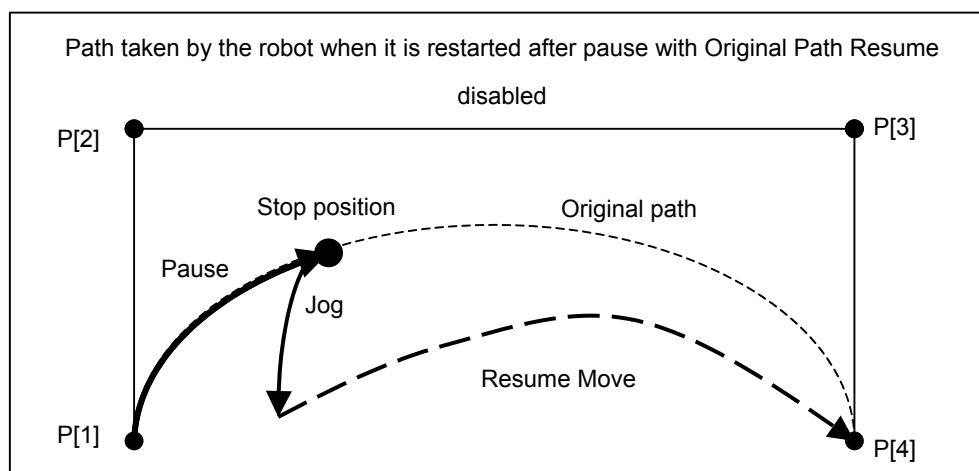


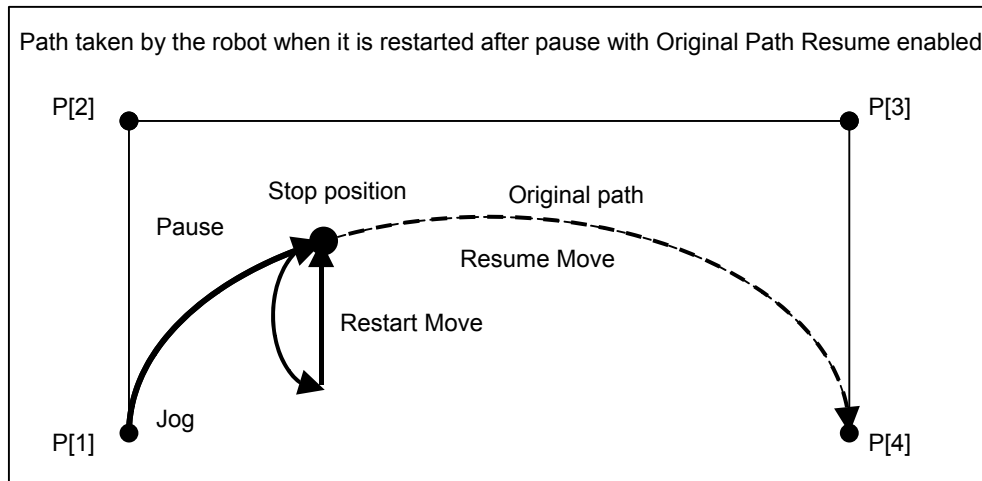
9.23 ORIGINAL PATH RESUME

Overview

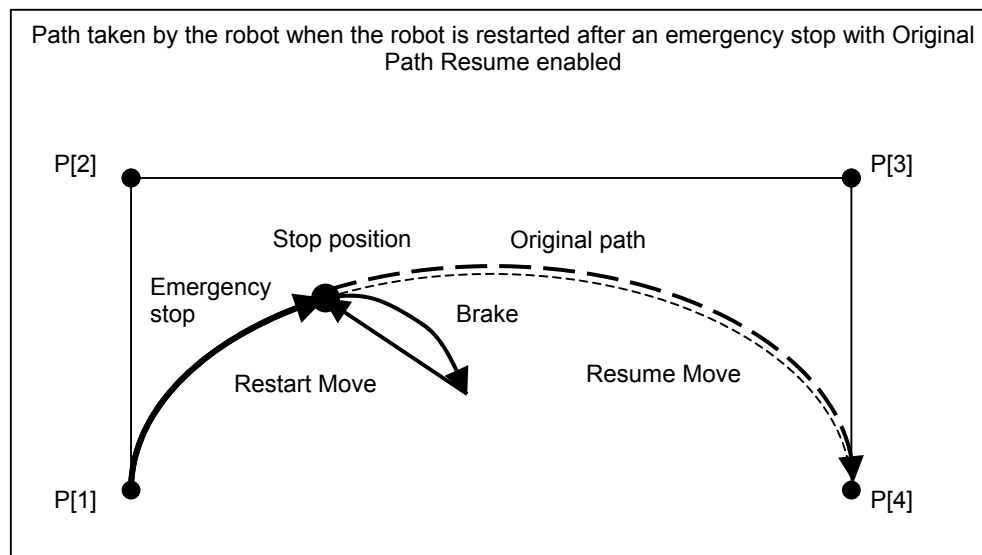
Original Path Resume works as follows: If an attempt is made to restart the program after the robot has paused or been brought to an emergency stop, the program restarts only after the robot has returned to its stop position even when the robot moves after a stop. The stop position can be the position where the robot pauses or stops at an emergency. The move of the robot from the current position to the stop position is called the “Restart Move”.

If an alarm that causes the robot to stop occurs, for example, the robot decelerates to a stop, and the TP program is caused to pause. The user might then jog the robot to another position to remove the cause of the alarm. If the TP program is restarted with Original Path Resume enabled, the robot first executes a Restart Move to the stop position before the program is restarted.





When the robot is brought to an emergency stop, the servo is turned off and the brakes are applied, possibly causing the robot to deviate from the normal path depending on the robot's payload mass, speed, and the direction of gravity. If Original Path Resume is used in restarting the TP program, the robot executes a Restart Move to the stop position before the program is restarted.



System variables

The following variables are used to enable or disable Original Path Resume.

`$SCR.$ORG_PTH_RSM = TRUE` (for enabling Original Path Resume)

`$SCR.$ORG_PTH_RSM = FALSE` (for disabling Original Path Resume)

`$MCR_GRP[].$RSM_MOTYPE` determines what restart move type to use.

- 1: Axis-specific
- 2: Straight line
- 3: Program (move type in use when the robot pauses)

`$MCR_GRP[].$RSM_SPEED` determines the Restart Move speed (in mm/s or %) that will be suitable for the move type.

`$MCR_GRP[].$RSM_ORIENT` determines the posture the robot has when it is restarted.

- 1: With wrist posture
- 3: Without wrist posture

Limitations

- If the Constant Path option is disabled, the path the robot takes during Resume Move may differ from the original path. The stop position of the robot will get close to the original path but is inaccurate.
- If the Constant Path option is enabled, the path the robot takes during Resume Move coincides with the original path.
- The Restart Move is subject to the same limitations as the other move types. If an alarm occurs during Restart Move related to, for example, “stroke limit,” restarting the program requires an operator’s intervention.
- If there is anything between the current and stop positions of the robot, the robot may collide against it because the Restart Move begins at the stop position. For example, the robot may have been jogged to a position where the robot will collide against the tool when it is restarted.

Resume Offset

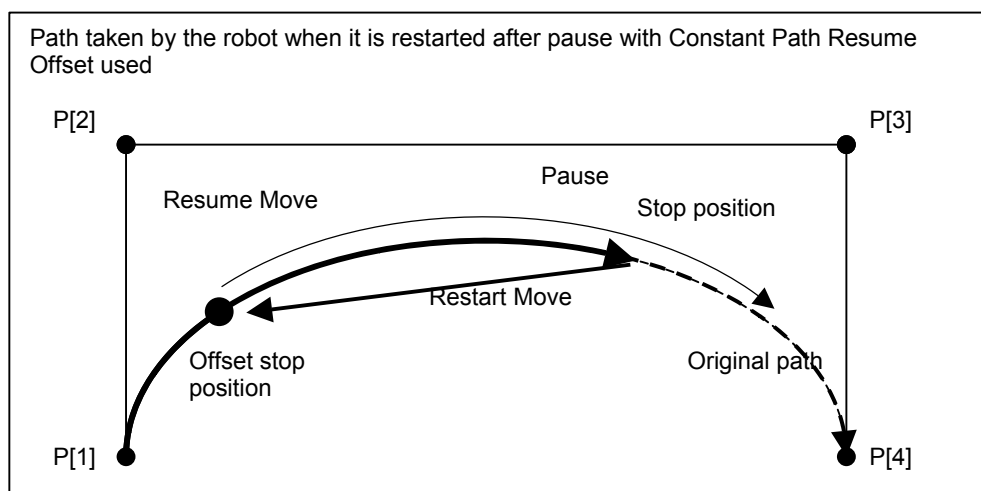
Resume Offset works as follows: When the robot is restarted, it moves back through “offset” from the stop position on the path. This option can be used in processes that are continuous along the path, such as cutting, welding, or dispensing. For example, sealant dispensing applications are sensitive to the TCP speed, so Resume Offset is used to allow the time needed for the robot to accelerate along the path before reaching the original stop position, thus resuming the normal sealant flow.

Limitations of Resume Offset

- Resume Offset requires Original Path Resume. It is subject to the limitations of Original Path Resume.
- Resume Offset uses the same Restart Move as Original Path Resume. Restart Move causes the robot to move directly to the offset stop position, not backward along the original path.
- The actual Resume Offset position gets close to the desired offset rather than the exact specified position.

Constant Path Resume Offset

Constant Path Resume Offset works in conjunction with the Constant Path option, which holds the TCP path in “memory,” so that at the end of pause the stop position can be offset to the position held in memory. The Resume Move causes the robot to accelerate from the offset stop position, follow the identical path to the original stop position, and then continue execution normally.

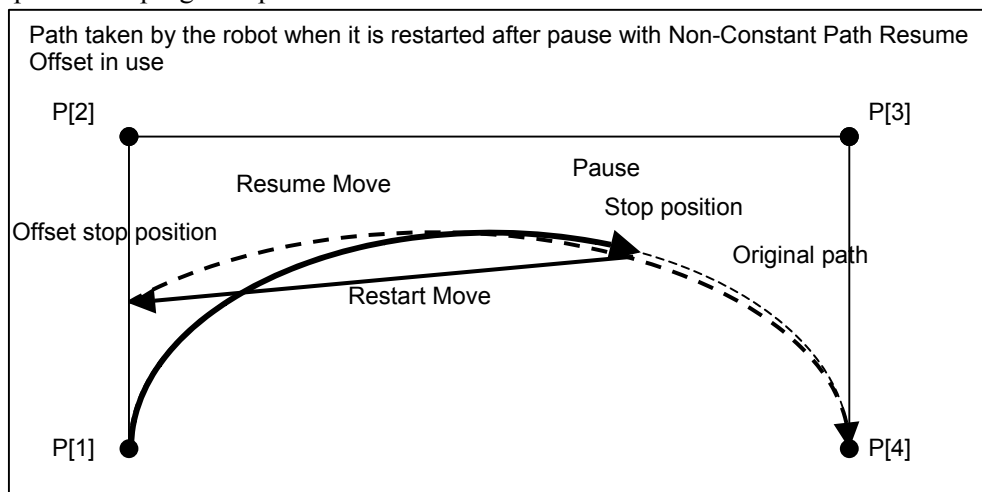


Limitations of Constant Path Resume Offset

- 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 before the robot pauses. Changing the offset distance after the robot pauses but before Resume Move is executed has no effect.
- Constant Path Resume Offset does not allow the robot to offset to a position where it is before positions are held in memory. If an attempt is made to move the robot to that position, the robot is moved to the earliest position to be held in memory. The amount of data that can be held in memory can be changed using the Resume Offset menu.

Non-Constant Path Resume Offset

Non-Constant Path Resume Offset computes the offset in reference to the current move statement. The computed position will not be on the original path. This is used in situations where the Constant Path option is not supported. The Resume Move path will deviate from the original path. However, it will be close to the original path if the program speed is low or its smoothness is at a low rate.



Limitations of Non-Constant Path Resume Offset

- Non-Constant Path Resume Offset records a stop position on the taught move statement rather than on the path. This can lead to path deviation when the path is resumed.
- Non-Constant Path Resume Offset cannot move the robot to an “offset” position where the robot is before the start position of the current move statement is reached. If an attempt is made to move the robot to that “offset” position, the start position of the current move statement is set as the actual offset position.

Selecting Resume Offset type

Table 9.23 Selecting Resume Offset type

\$pcfg.\$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

If the Constant Path option is loaded, Constant Path Resume is used. The value of \$PCCFG.\$RESUME_OFST.\$RO_ENABLE determines whether to use Resume Offset.

If the Constant Path option is not loaded, Non-Constant Path Resume is used. The value of \$MCR_GRP[.].\$RSM_OFFSET determines Resume Offset.

Some options do not support Constant Path Resume. Non-Constant Path Resume is used with those options, which include:

- Touch Sensing
- Line Tracking
- Servo Gun
- Weaving

Resume Offset setup menu

Selecting [MENU]→[SETUP]→[Resume Offset] displays this menu.

The Resume Offset menu is used to configure the memory to be used with Constant Path Resume. This configuration has a direct effect on the way the system memory is used.

Resume Offset Status indicates whether Constant Path Resume Offset is enabled. It is equivalent to the system variable \$CPCFG.\$RESUME_OFST.\$RO_ENABLE.

Max. Res. Offset Dist specifies the maximum amount of offset that might be used. The larger the value, the more memory is required.

Nominal Process Spped specifies a reference speed at which the Resume Offset process will be run. The smaller the value, the more memory is required.

Once the configuration is changed, it is necessary to turn the power off and on again, so the new configuration is enabled. The display given below appears after the configuration is changed. There are two choices: Turn the power off and on again to enable the new configuration or keep the power on to preserve the old configuration.

System variables

\$CPCFG.\$RESUME_OFST.\$RO_ENABLE

If the Constant Path option is loaded, Constant Path Resume is enabled.

\$MCR_GRP[.].\$RSM_OFFSET

Resume Offset distance (mm) to use

\$MOR_GRP[.].\$OGDST_RATIO

Actual Resume Offset ratio used in attaining the target Resume Offset

1.0 = 100%

0.0 = 0%

-1.0 = yet to be initialized

9.24 PROGRAM TOOLBOX

The Program Toolbox has the following features. Installing the Program Toolbox option software makes available the following Program Toolbox options:

- Soft limit setting

9.24.1 Soft Limit Setting

Using the soft limit setting option enables the software axis limits for a robot to be determined automatically; so the hard stop locations for the J1 axis can be determined. Setting up axis limits without using the soft limit setting option requires executing all programs step by step, recording the maximum value for each angle before the axis joint limits can be set up manually. It also requires jogging the robot to the position limits specific to the J1 axis and determining the best hardware stop position for the J1 axis manually.

The soft limit setting option 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 uses the information to set up the specified joint limits automatically while taking a user-specified limit buffer into

account. The soft limit setting option also reports the appropriate locations for the J1 axis hard stop according to the same maximum taught joint angle for all programs.

By default, software axis limits are set up only for the J1, J2, and J3 axes. If necessary, they can be set up for other axes.

A limit buffer is added to the detected limits to allow tolerance for motion between positions when a program is executed. When the robot moves from one position to another, the motion of the robot between the positions might get out of the axis limits. The limit buffer is applied to the detected maximum and minimum taught points to ease the axis limits so that a joint limit error will not occur on the robot motion between the taught points.

A limit buffer can be set anywhere between 0 and 50 degrees. A limit buffer of 10 degrees is set up by default. In many cases, this limit buffer degree provides adequate ease. If many limit errors occur during program execution after the soft limit setting option is used, increase the limit buffer and try again.

Before limits are set up, axes to which the limits are to be applied must be selected and, and if necessary, the limit buffer value must be re-set. Factory-set limits can be restored if newly set values become unnecessary.

Using the limit setting option

Table 9.24.1 lists each soft limit setup item. Use Procedure 9-26 to calculate mounting locations for limit blocks on the robot flange.

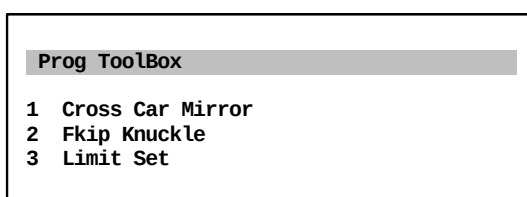
Table 9.24.1 Toolbox soft limit setup menu items

Item	Description
Axis	This item is the number assigned to the axis for which limits can be set up.
Set Limit	This item indicates whether a limit has been set up.

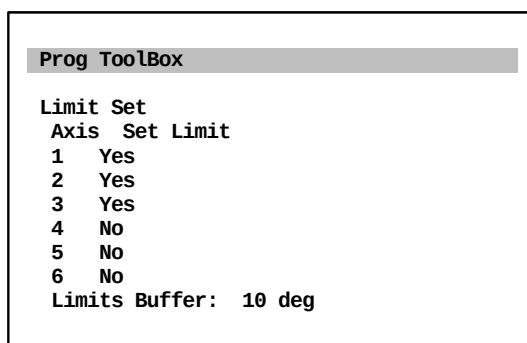
Procedure 9-25 Using the soft limit setting option to set up software axis limits

Step

1. Press [MENUS].
2. Select UTILITIES.
3. Press [F1, [TYPE]].
4. Select Prog ToolBox. The following menu appears.



5. Move the cursor to [Limit Set] and press [ENTER]. The following menu appears.



6. To select an axis for limit setting, follow the steps below:
 - a. Move the cursor to the number assigned to a desired axis.
 - b. Press [F4, Yes] to select the axis. If [F5, No] is pressed, no limit is set up for the axis,
 7. To specify a limit buffer, move the cursor to Limits buffer, enter a degree, and press [ENTER].
 8. To set up the axis limits, press [F3, EXECUTE].
- When limit setting is completed, the following information is displayed on the screen.

Prog ToolBox			
New Limits			
Axis	LOWER	UPPER	
-J1	-57	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
Min Hole# for J1: -4 offset toward - 3			
Min Hole# for J1: 4 offset toward 4			
Min Hole# for J2: -3 Max Hole# 0			
Max Hole# for J3: 0 Max Hole# 2			
YOU MUST COLD START TO TAKE EFFECT			

The displayed mechanical stop numbers indicate the locations of the limit blocks on the J1, J2, and J3 axes. Mounting the limit block on the J1 axis requires aligning the hole at the center of the limit block with the reported J1 axis mechanical stop hole on the base of the robot.

Additional information

The displayed mechanical stop numbers indicate the positions of the center bolt hole on each mechanical stop.

Note: The current version does not support the mounting positions of mechanical stops.

Additional information

The following operation is not supported on some robot models.

9. To reset the axis limits to their factory settings, press [F2, DEFAULT].

Additional information

Enabling new software axis limit settings always requires cold-starting the controller.
10. After axis limit setting is completed, enable the new software axis limit settings by cold-starting the controller according to the following steps.
 - a. If the controller is already on, turn it off.
 - b. On the teach pendant, press and hold down the [SHIFT] and [RESET] keys.
 - c. While still holding down the teach pendant keys, turn the power on.
 - d. After the teach pendant has displayed its menu, release the teach pendant keys.

- Additional information

If many joint limit errors occur during program execution, increase the limit buffer and re-run the program.

9.25 ADVANCED CONSTANT PATH

9.25.1 Overview

In addition to the Constant Path feature, the Advanced Constant Path package consists of the features for easy teaching and cycle time reduction of the following functions.

- Linear Distance
- Corner Region
- Process Speed
- Max Speed

It requires that the Constant Path option be loaded and enabled.

9.25.2 Linear Distance

Linear Distance is a robot motion feature that is useful for pick and place material handling applications such as palletizing.

Fig. 9.25.2 (a) PICK and PLACE Application 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.

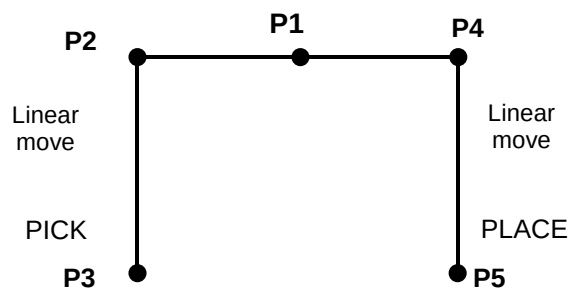


Fig. 9.25.2 (a) PICK and PLACE application

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 Fig. 9.25.2 (b) PICK and PLACE Application with CNT100.

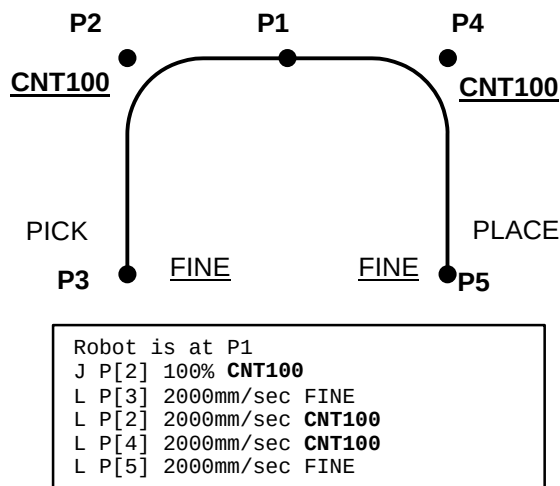


Fig. 9.25.2 (b) PICK and PLACE Application with CNT100

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 Fig. 9.25.2 (c) Adjusting P3 and P5 with Linear Distance .

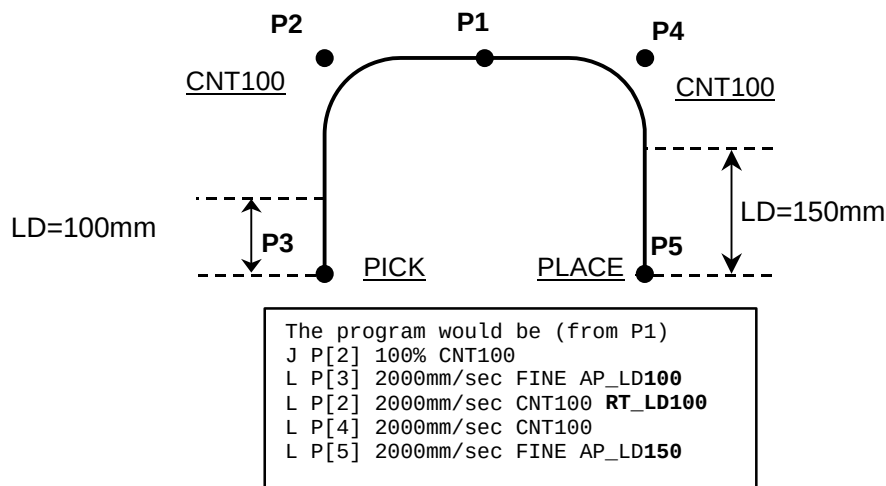


Fig. 9.25.2 (c) Adjusting P3 and P5 with Linear Distance

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.

9.25.2.1 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]

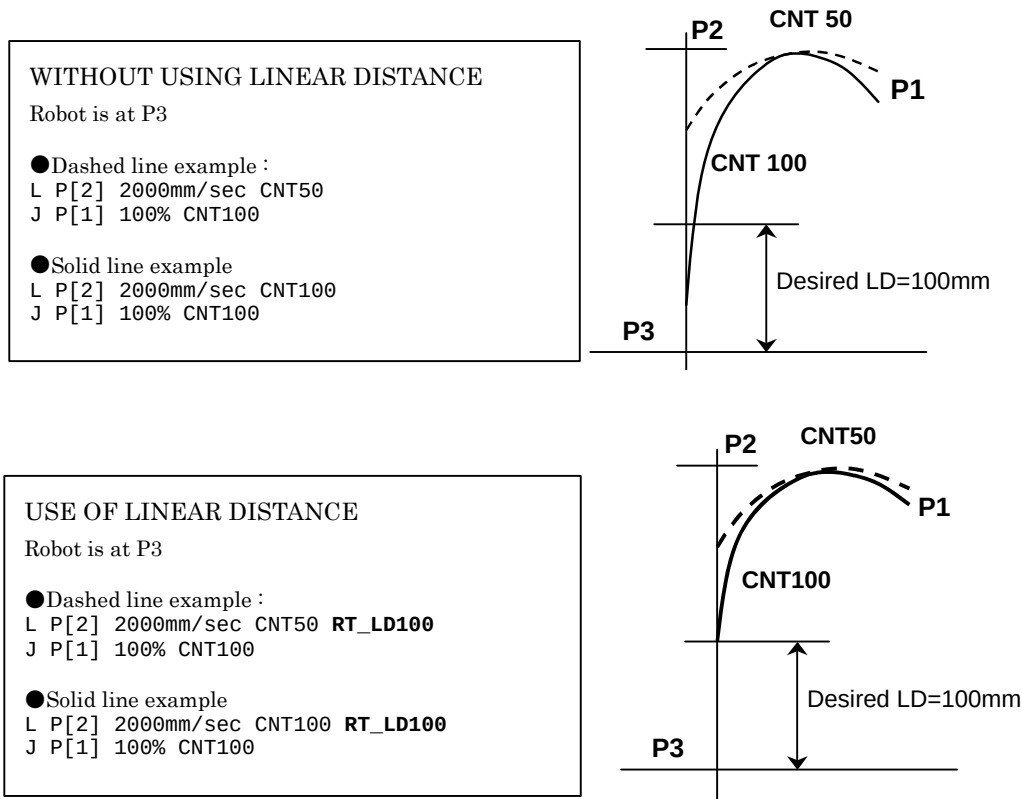
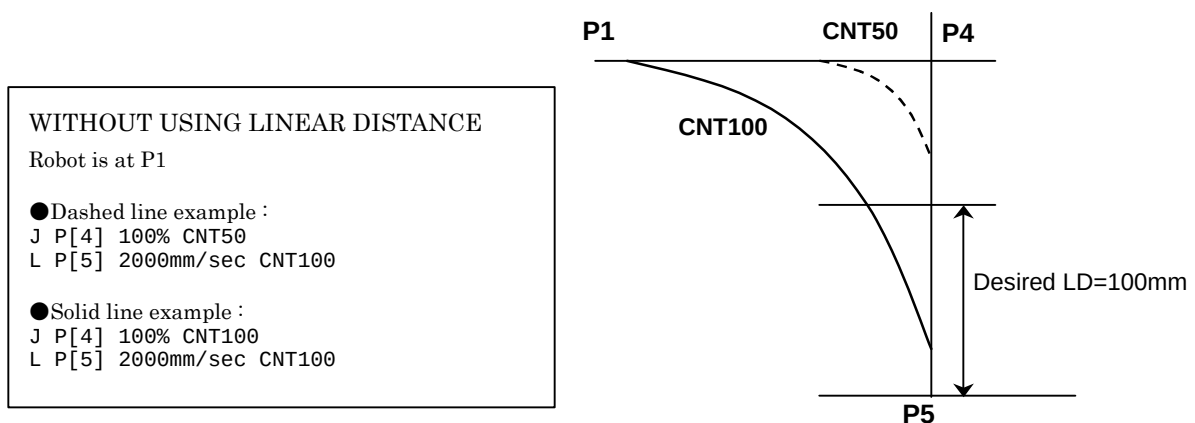


Fig. 9.25.2.1 (a) RT_LD:Effect of CNT Value

The RT_LD value affects the corner of P3-P2-P1 in Fig. 9.25.2.1 (a) RT_LD: Effect of CNT Value . 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.



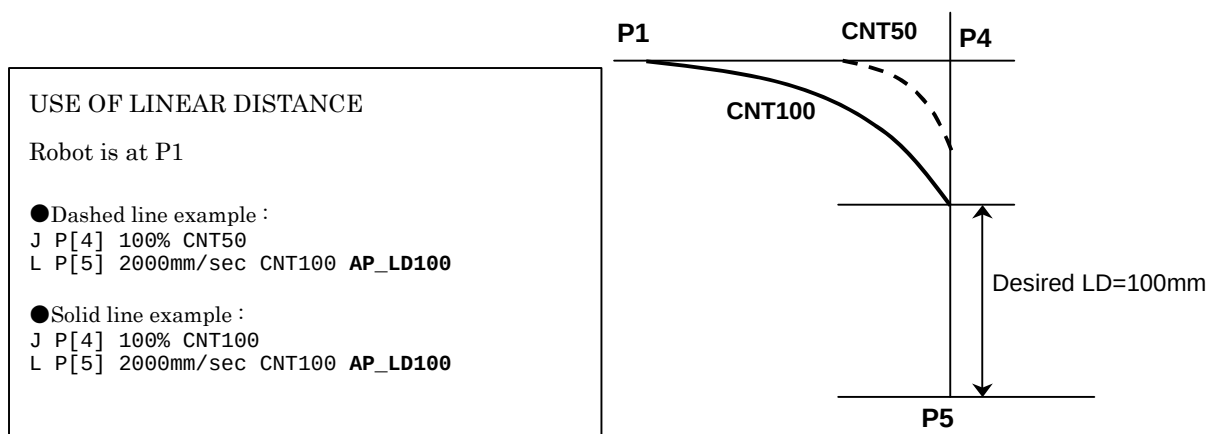


Figure 9.25.2.1 (b) Place Motion: Two Possible Traces

The AP_LD value affects the corner of P1-P4-P5 Figure 1.5 Place Motion: Two Possible Traces . 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.

9.25.2.2 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.

9.25.2.3 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.

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 touch up and modify motion instructions, move the cursor to the line number of the motion instruction you want to modify.

- 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 2
1 Retract LD
2 Approach LD
3 Tool Offset
4 Tool Offset PR[ ]
5 TIME BEFORE
6 TIME AFTER
7 DISTANCE BEFORE
---next page---

```

- 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.

9.25.3 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).

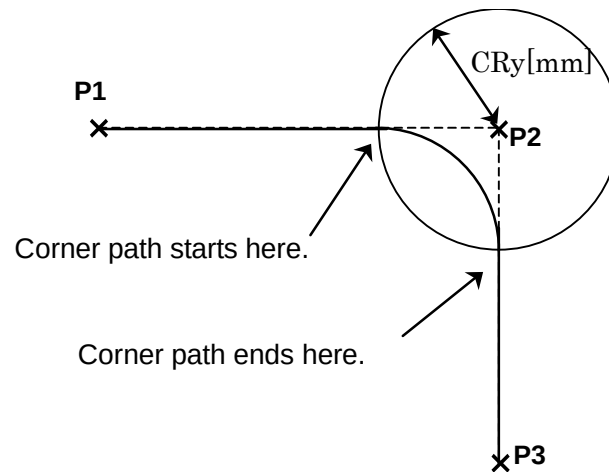


Fig. 9.25.3 Corner Path

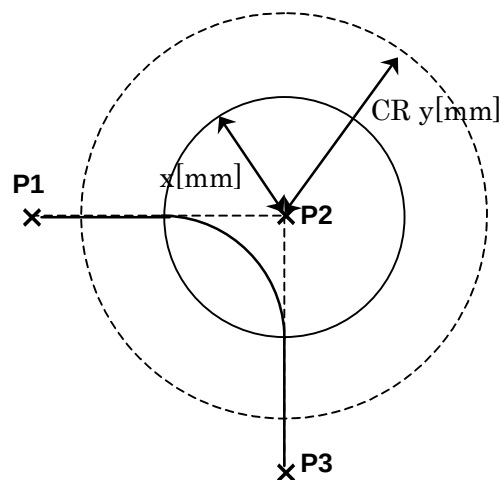
When you set corner region, use the following guidelines:

- 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

9.25.3.1 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.

CR y[mm] > half segment distance
Actual CRx=half segment distance



Unequal segment lengths case
Actual CR=half segment distances x_1 , x_2

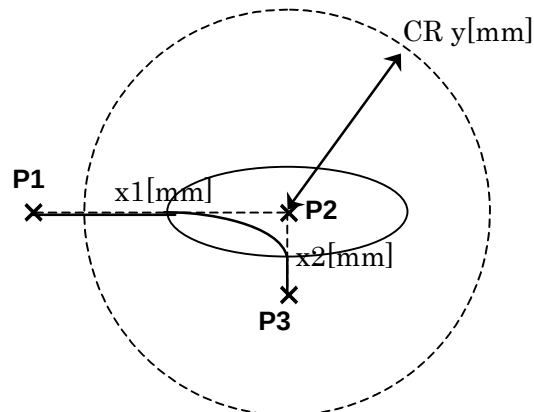


Fig. 9.25.3.1 Half Segment Length

9.25.3.2 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

9.25.3.3 Constant path with respect to program speed changes

Program section with consecutive CRy – path could be maintained much better compared to other termination types even when program speed changes, as shown in the following figure.

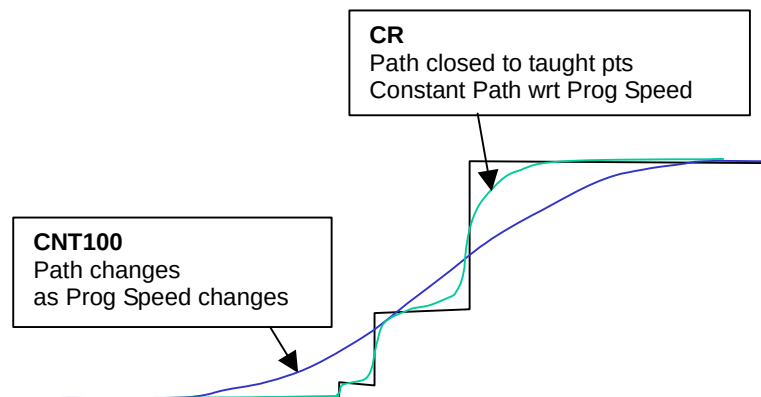


Fig. 9.25.3.3 Program Speed Changes

9.25.3.4 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
- COORD (not in the first release)
- RTCP
- Line Tracking

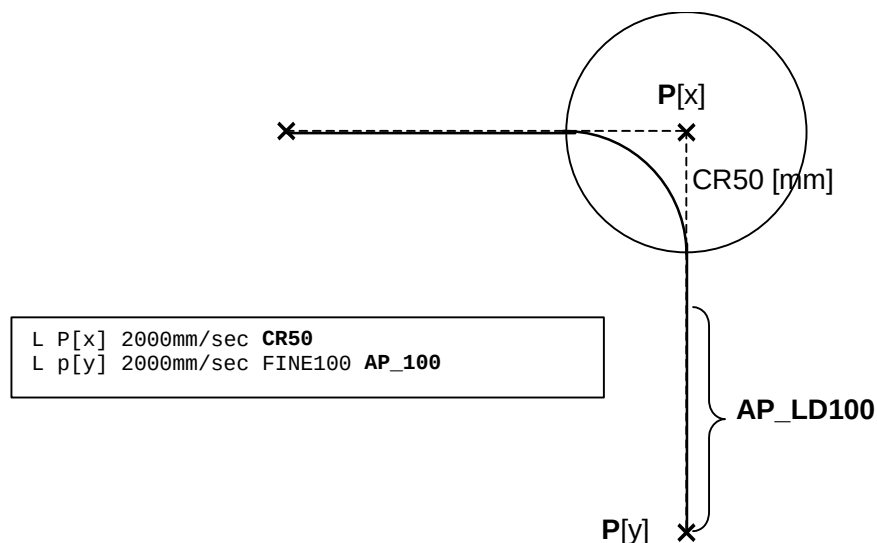
CRy has no obvious geometric meaning (corner path would be naturally generated by motion blending) during the following transitions

- between COORD and non-COORD (if supported)
- between RTCP and non-RTCP (if supported)
- between Tracking and non-tracking (if supported) .

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 the corner angle between two motion lines is almost zero, CNT100 will be used regardless of the specified CR value.

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.



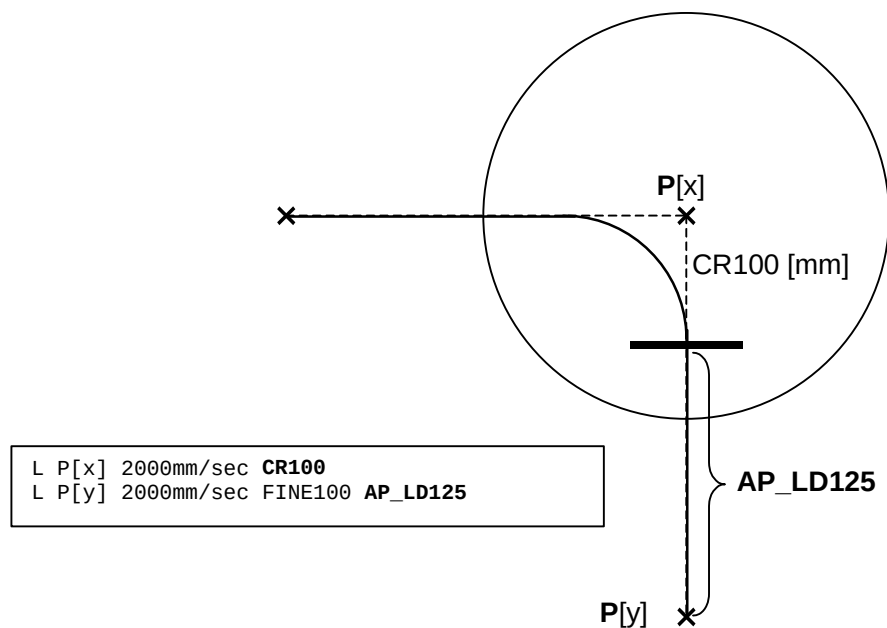


Fig. 9.25.3.4 Corner Path determined by CRy if Linear Dist is satisfied

9.25.3.5 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 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.

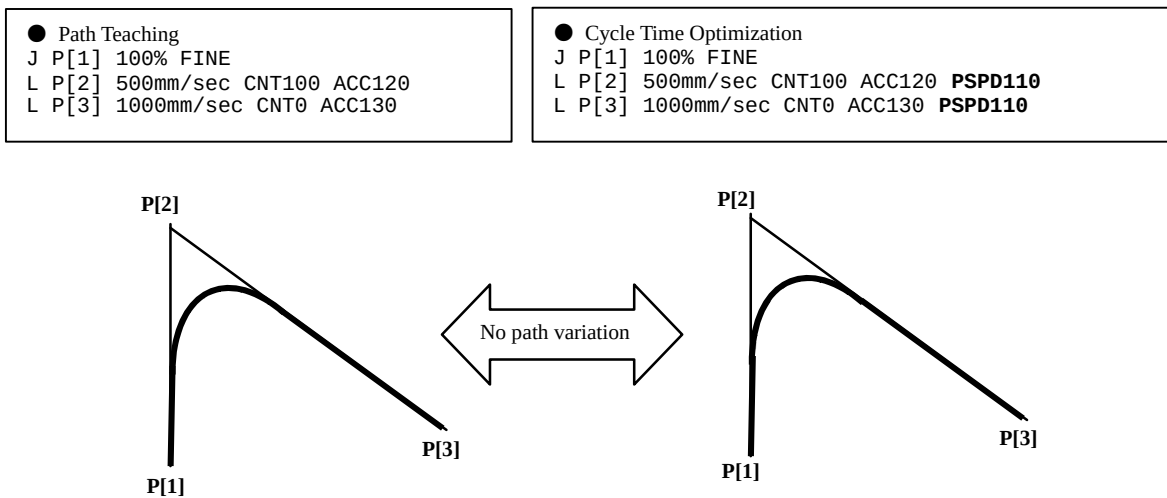


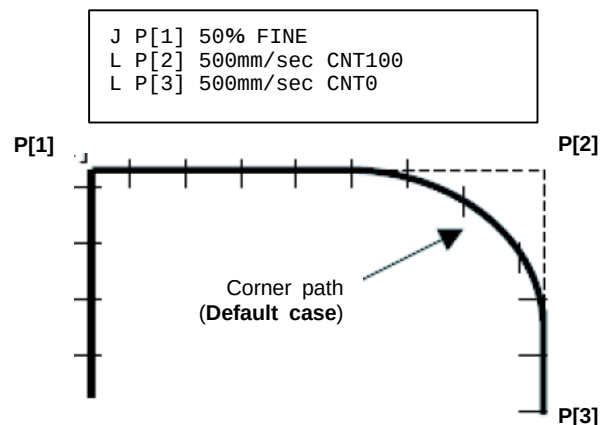
Fig. 9.25.3.5 Process speed tweaking

**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 : *** = 100, motion behavior is exactly the same as 100% speed override in the default case, as though there were no PSPD100
- Case 2 : *** > 100, the speed will be faster than 100% speed override in the default case
- Case 3 : *** < 100, the speed will be slower than 100% speed override in the default case
- Default case :

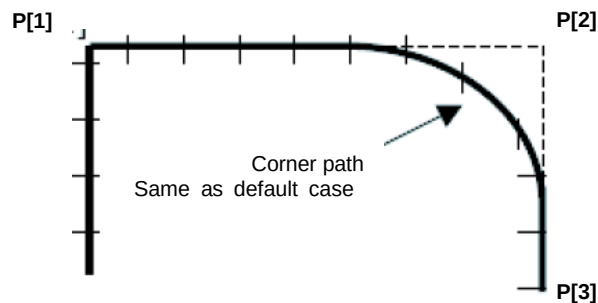


- Faster motion case (path is the same, cycle time is shorter):

```

J P[1] 50% FINE PSPD110
L P[2] 500mm/sec CNT100 PSPD110
L P[3] 500mm/sec CNT0 PSPD110

```

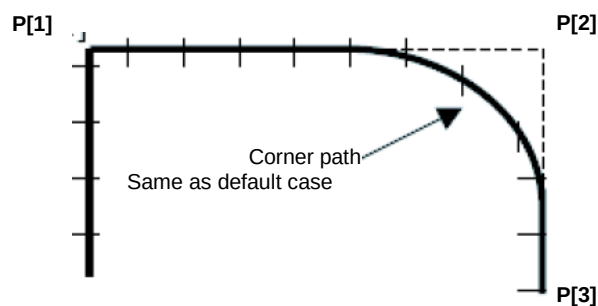


- Slower motion case (path is the same, cycle time is longer):

```

J P[1] 50% FINE PSPD50
L P[2] 500mm/sec CNT100 PSPD50
L P[3] 500mm/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.

9.25.3.6 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;
- With large PSPD value or very short segments, the actual corner path might deviate from the one without PSPD option.

9.25.4 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.

9.25.4.1 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.

<pre>LINEAR_MAX_SPEED = 1200 L P[1] max_speed CNT1000</pre>

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 Max speed feature):
 - Line Tracking
 - RTCP function
 - Multi group motions

9.26 HIGH PERFORMANCE SOFTFLOAT FUNCTION

9.26.1 Outline

This function makes you able to use not only current softfloat but also “pushout softfloat” with the same user interface.

9.26.2 Relation to Current Softfloat Function

Soft float which is able to be used in current “softfloat function” option. (There are several robot type that do not support all softfloat type)

- Joint softfloat
- Cartesian softfloat

Soft float which is able to be used in “high performance softfloat” option. (There are several robot type that do not support all softfloat type)

- Joint softfloat
- Cartesian softfloat
- Pushout softfloat

Each softfloat type has the following features.

Each softfloat schedule can select different softfloat type.

- Joint softfloat
This is current joint softfloat. Softness around robot joint can be specified.
- Cartesian softfloat
This is current Cartesian softfloat. Softness along Cartesian coordinate and around Cartesian coordinate can be specified.
It is good at contouring and face matching with robot motion.
- Pushout softfloat
In this softfloat, force and moment necessary become smaller than current Cartesian softfloat.
Softness around Cartesian coordinate can not be specified.
This softfloat does not work in where $J5 = -3.5 \sim +3.5$.
It is good at following the pushout.

9.26.3 Procedure of operation

Procedure is the same as current softfloat function about “joint softfloat” and “Cartesian softfloat”.

Procedure to show pushout softfloat screen

Select “PUSHOUT” in softfloat detail screen.

- 1 Push MENU key to show screen list.
- 2 Select “6 SETUP”.
- 3 Push [F1] ([TYPE]) key to show screen list.
- 4 Select “softfloat”. Softfloat list screen will be shown.
- 5 Set cursor on schedule No. to set pushout softfloat, and push [F3] (detail) key.

SETUP/SOFTFLOAT				JOINT 10 %
Group1				10/10
No	Type	Start(%)	Comment	
1	CART	*****	[	]
2	CART	*****	[	]
3	CART	*****	[	]
4	CART	*****	[	]
5	CART	*****	[	]
6	CART	*****	[	]
7	CART	*****	[	]
8	CART	*****	[	]
9	CART	*****	[	]
10	CART	*****	[	]
[TYPE] GROUP DETAIL				

- 6 Push NEXT key until “PUSHOUT” are shown on function key line.
- 7 Push [F 4] (PUSH) key. Detail screen of pushout softfloat will be shown.

SOFTFLOAT(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]	
[TYPE] NUMBER LIST JOINT CART >			

“PUSH” are not shown on function key line if high performance softfloat option is not installed.

Procedure to set pushout softfloat

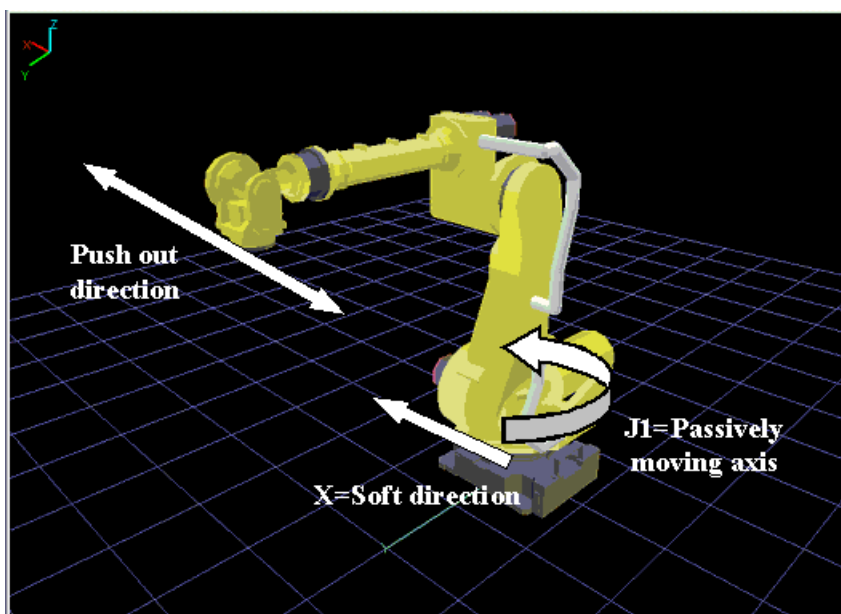
In addition to the setup of Cartesian softfloat, “motion type” and “soft direction” should be set. “Soft rat” and “soft tol” are removed.

- 1 Set “enable/disable” and “coordinate” same as current softfloat.
- 2 Set “soft direction” to the direction which should be soft in the schedule No.
- 3 Set 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.

SOFTFLOAT(PUSHOUT)		JOINT	10 %
		4/6	
Group1			
1	Schedule No.[10]	:[*****]	
2	Enable/Disable	:[ENABLE]	
3	Coordinate	:[USER]	
Motion Type			
4	J1:[PUSH]	J2:[PUSH]	J3:[FREE]
5	J4:[PUSH]	J5:[PUSH]	J6:[PUSH]
6	Soft direction:[Z axis]		
[TYPE] NUMBER LIST FREE PUSH >			

Example)

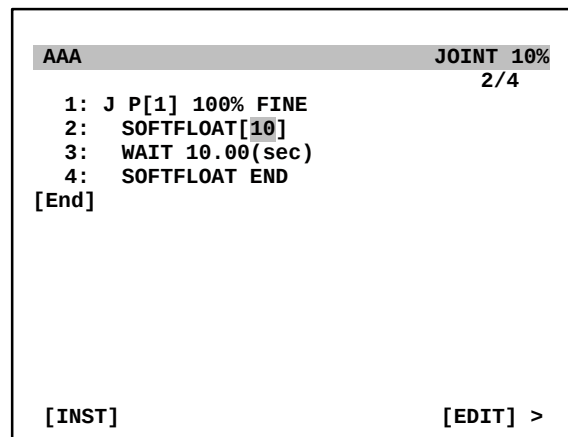
When J1 is 90 degree, to select X axis of world coordinate as “Soft direction”, motion type of J1 should be “FREE”.



Procedure to use pushout softfloat

Procedure is the same as that of current softfloat function.

- 1 In edit screen, open the program to use pushout softfloat.
- 2 Push NEXT key until “INST” is shown on function key.
- 3 Teach softfloat[...] instruction on the line to start pushout softfloat.
- 4 Set softfloat schedule No. of pushout softfloat in the index of softfloat[...] instruction.
- 5 Teach softfloat end instruction on the line to stop pushout softfloat.



9.26.4 Knowhow of Softfloat

Timing to start softfloat

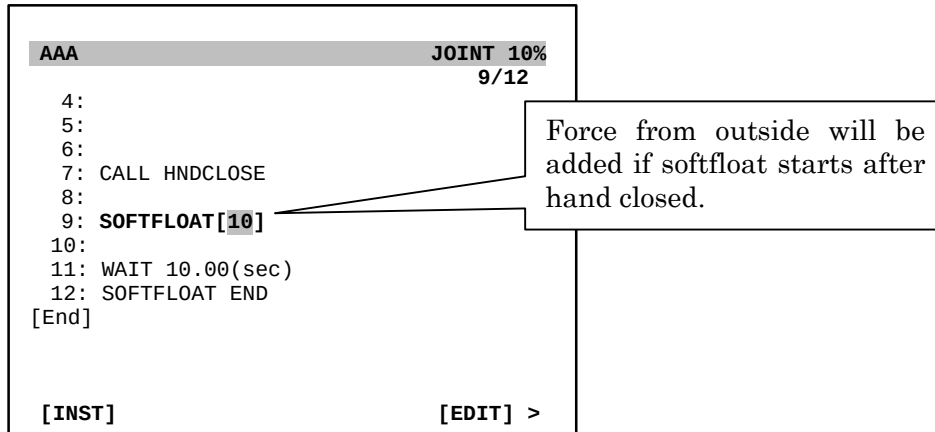
Force from outside should not be added when softfloat starts.

If softfloat starts with not only gravity force but also force from outside, robot will move above or below, and “Stop/Move error excess” or “Softfloat time out” alarm will occur.

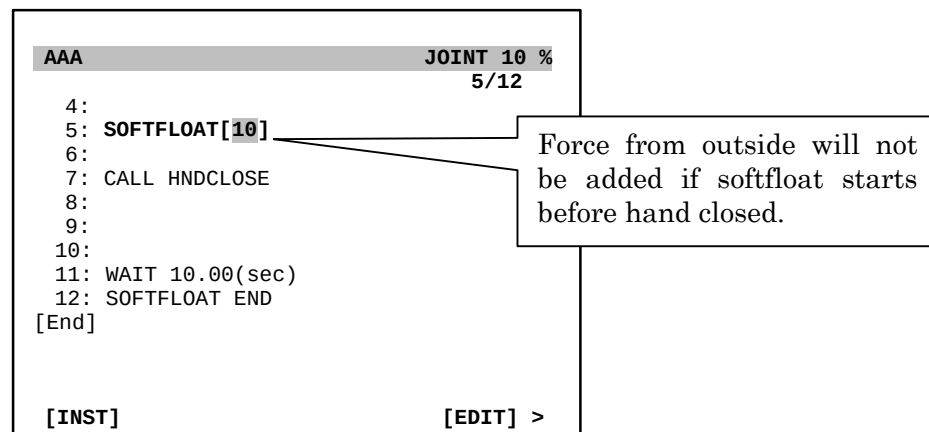
For example in unloading application, you should make program as that softfloat starts before hand closed.

This knowhow is effective in all softfloat of joint, Cartesian and pushout.

Bad example)



Good example)



Knowhow to improve softness

Even with the same softfloat schedule setting, the softness of softfloat differs depending on the direction of the motion just before softfloat starts.

It is caused by the feature of that softfloat 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.

So, you should make program as that just before softfloat starts, move robot a little in the direction opposite to the direction robot should move by softfloat, then move robot back to the position to start softfloat. The distance necessary is 1~2[mm] or longer.

For example, when you want to start softfloat to move in +X direction in P[1], you should add the following motion instruction between the motion instruction to P[1] and softfloat instruction.

To go back to P[1] in the next line, move opposite direction in this line.

```

AAA                                JOINT 10%
                                   6/8
1: J P[1:sflt start pos] 100% FINE
2:
3: L P[2:-x1mm] 100mm/s CNT0 INC
4: L P[3:+x1mm] 100mm/s CNT0 INC
5:
6: SOFTFLOAT[10]
7: WAIT 10.00(sec)
8: SOFTFLOAT END
[End]

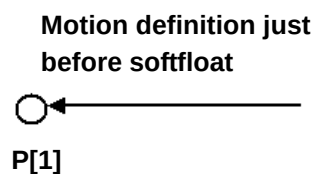
```

Move back to P[1] in the the direction to move by softfloat.

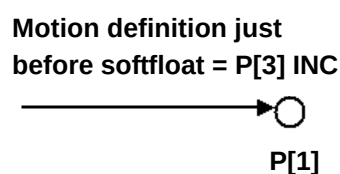
[INST]
[EDIT] >

By this method, robot will become more soft to move +X direction and less soft to move -X direction. This knowhow is effective in all softfloat of joint, Cartesian and pushout.

Bad example)



Good example)



Direction to move by softfloat →

Direction to move by softfloat →

9.26.5 Knowhow of Cartesian Softfloat

Cartesian softfloat is supported both in current softfloat and in high performance softfloat.

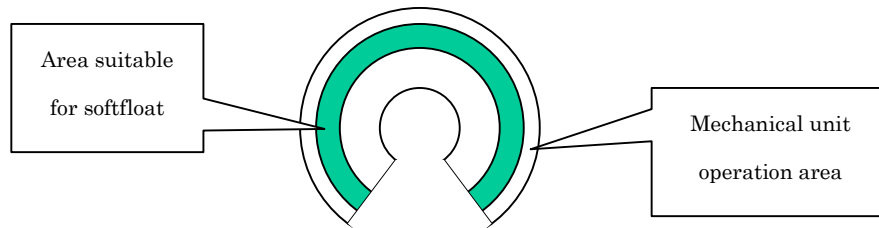
Softness along Cartesian coordinate and around Cartesian coordinate can be specified.

It is good at contouring and face matching with robot motion.

Knowhow to use it effectively will be explained.

All application common knowhow

- Layout and tool selection
Area in 10~20[%] inside from the outside border of mechanical unit operation area is recommended.
And select the tool as short as possible.



- Schedule setup
Set 100[%] in “soft rat” of the direction or rotation that you want to make soft
Adjust “soft tol” of pushing direction to limit force to avoid pushing with too large force.
- Velocity teaching
In pushing application while robot is moving, if the taught speed is slow, robot may not be able to move because it can not overcome the friction.
Teach the velocity more than about 100[mm/s].

Contouring application peculiar knowhow

- Schedule setup
Example) In case pushing Z direction

	Soft Rat		Soft Tol		
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 force.
X rotation	0	%	0	%	Make rotation hard.
Y rotation	0	%	0	%	
Z rotation	0	%	0	%	

- TCP setup
In application which should keep tool direction, TCP could be X, Y and Z =0 when softfloat starts.
By this, difference between soft direction and hard direction appears well.

Example)

AAA
JOINT 10
7/12

```

4:
5: UTOOL_NUM=9
6:
7: SOFTFLOAT[10]
8:
9: UTOOL_NUM =2
10:
11: WAIT 10.00(sec)
12: SOFTFLOAT END
[End]

```

[INST]
[EDIT] >

Tool frame whose X=0, Y=0, Z=0 and W, P, R are the same as actual tool

TCP set as actual tool

Face matching application knowhow

- Schedule setup

Example) In case pushing Z direction

	Soft Rat		Soft Tol		
X direction	0	%	0	%	Make X, Y direction hard.
Y direction	0	%	0	%	
Z direction	0	%	70	%	Limit force to avoid pushing with too large force.
X rotation	100	%	0	%	Make rotation soft.
Y rotation	100	%	0	%	
Z rotation	100	%	0	%	

- TCP setup

Use the tool frame set as actual tool. You don't have to change TCP when softfloat starts.

Select the tool as short as possible. X, Y and Z direction can't become hard enough if tool is long.

Moment of the direction to match face is necessary. Select tool that can generate moment of the direction to match face by pushing even with largest orientation mismatch.

9.27 MULTITASKING FUNCTION

9.27.1 Overview

The multitasking function enables multiple programs to be executed at the same time. A "task" refers to an active program. For example, the multitasking function allows the robot to work based on two programs being concurrently executed, one for controlling the robot and the other for controlling peripherals and extended axis (for multiple groups). Through concurrent execution of programs, the cycle time can be reduced, and the state (of input signals, for example) can be monitored during robot operation.

9.27.2 Functions

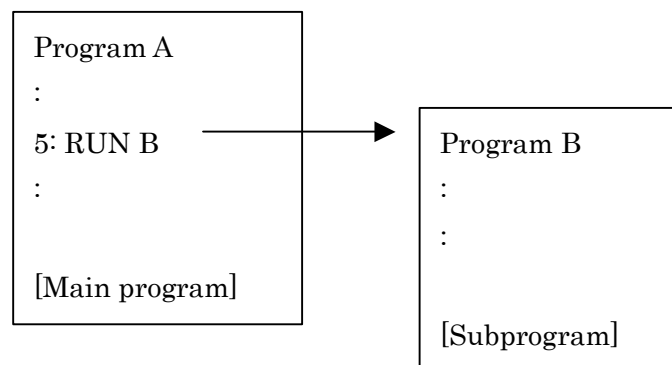
9.27.2.1 Notes on program creation

Create programs as usual. Note the following:

- A signal control program and a program intended only to read data do not use an operation group. Be sure to disable operation groups for these programs.
On the program detail screen, a group mask should be set to [*,*,*,*,*,*,*].
- Programs cannot be executed at the same time if they use the same operation group.
- Programs that they use different operation groups can be executed at the same time.

9.27.2.2 How to start programs in multitasking

One program can start another as a task using the "RUN" instruction. The starter is referred to as a main program, while the started program is referred to as a subprogram.



In the above example, program A starts program B using the "RUN" instruction, and then both programs are executed concurrently. Program A is the main program of program B, and program B is the subprogram of program A.

If program A uses the "RUN" instruction to start program B as in the above example when program B is already active, an alarm is issued, suspending the execution of program A. In this case, program B must be terminated before program A is started.

9.27.2.3 Behavior of main program and subprogram

The main program and subprogram are halted

- When you select the main program and resume the execution, the execution of the subprogram is also resumed.
- When you select the subprogram and resume the execution, only the execution of the subprogram is resumed.
- When you select the main program and start backward execution, the subprogram is also executed in the backward direction.
- When you select the subprogram and start backward execution, only the subprogram is executed in the backward direction.

The main program is active and the subprogram is halted

- You cannot select the main program and resume the execution (in the forward or backward direction) (because the program is already active).
- When you select the subprogram and resume the execution, only the execution of the subprogram is resumed. This operation does not affect the main program execution.
- When you select the subprogram and start backward execution, only the subprogram is executed in the backward direction.

The main program is halted and the subprogram is active

- When you select the main program and resume the execution, the execution of the main program is resumed. The ongoing execution of the subprogram continues.
- When you select the subprogram and resume the execution, the ongoing execution of the subprogram continues. The execution of the main program is not resumed.
- When you select the main program and start backward execution, the main program is executed in the backward direction. The ongoing execution of the subprogram continues.
- When you select the subprogram and start backward execution, the subprogram is not executed in the backward direction. The ongoing execution continues. The main program is not executed in the backward direction.

Execution in single-step mode

- When you execute the main program in single-step mode, the subprogram is also executed in single-step mode.
- When you select the subprogram and resume the execution in single-step mode, only the subprogram is executed in single-step mode.

Halt and forced termination of program execution

- The main program and subprogram are independent of each other for halt and forced termination of program execution.
- Halting or forcibly terminating the main program does not affect the execution of the subprogram.

Execution of the main program in the backward direction

- After the execution of the main program is started in the backward direction, the execution stops at the "RUN" instruction and does not continue further.
- If you want the execution of the main program to continue in the backward direction beyond the "RUN" instruction, move the cursor to the line immediately before the "RUN" instruction.

9.27.2.4 Notes on main program and subprogram

Note the following when using the "RUN" instruction to start a subprogram:

- When the main program is executed in the forward or backward direction, the subprogram is also always executed in the same direction as the main program.
- If you want to execute the main program alone in the forward or backward direction, terminate the subprogram on the monitor screen, and then start execution in the forward or backward direction.
- If you want to execute the subprogram alone in the forward or backward direction, select the subprogram, and then start execution in the forward or backward direction. You do not need to terminate the main program.
- During execution in the backward direction, only motion instructions are executed. If synchronization between the main program and subprogram has been achieved by a register, the synchronization is not provided during the execution in the backward direction; the main program and subprogram are executed independently in the backward direction.

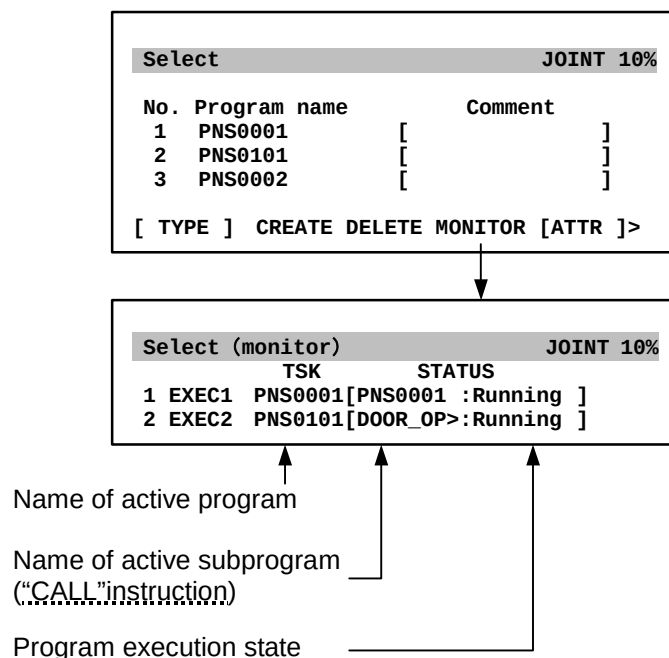
9.27.3 Monitor

On the monitor screen, you can see at a time the state of multiple programs being executed. The active programs and halted programs are listed on the monitor screen.

The monitor screen shows the following information:

- Names of active or halted programs
- Names of active or halted subprogram ("CALL" instruction)
- Execution state (active, halted)

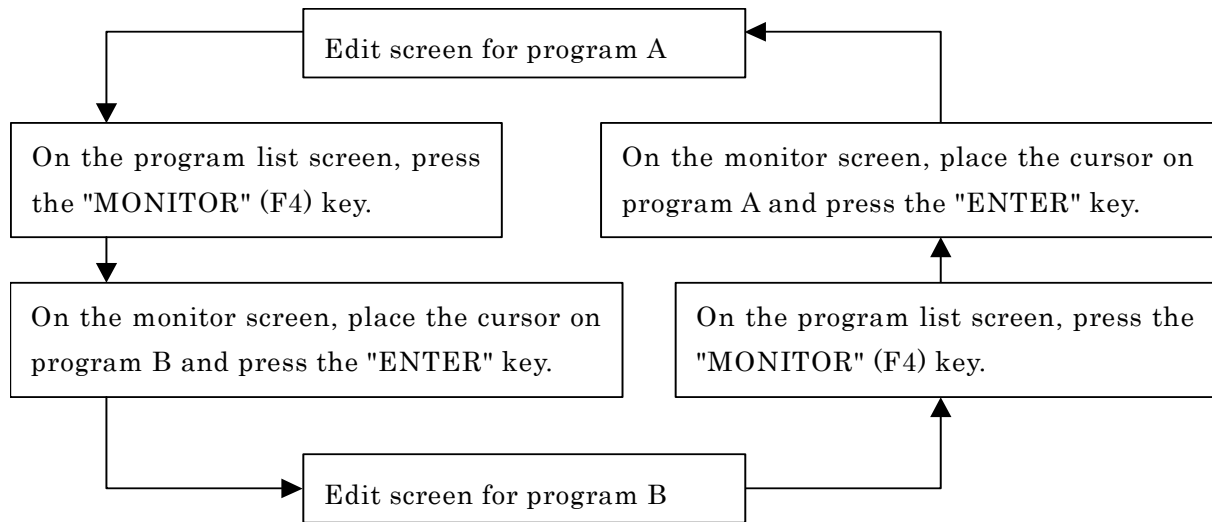
The monitor screen appears when you press the "MONITOR" (F4) key on the program list screen.



Place the cursor on a program you want to monitor, and press the "ENTER" key. The edit screen for the program appears, showing the execution state.

When multiple programs are active and you want to see the editing screen for each program in turn, you can switch among the screens on the above monitor screen with relative ease.

For example, when programs A and B are executed at the same time and you want to check the execution state of the programs on their respective edit screens, you should do the following:



When only one program is executed, pressing the "MONITOR" (F4) key on the program list screen skips the monitor screen and takes you directly to the editing screen. The monitor screen appears only when two or more programs are active or halted.

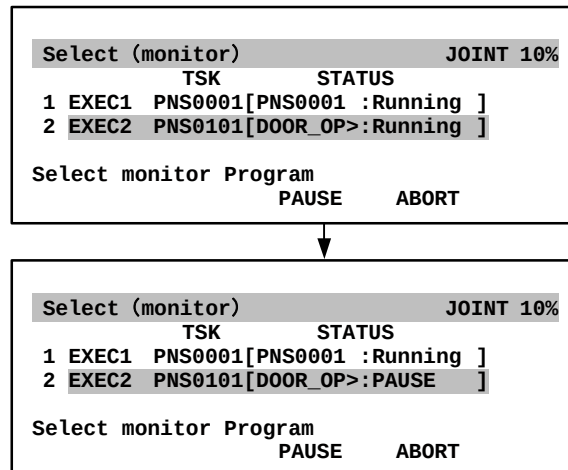
9.27.4 Halt and Forced Termination

If you do one of the followings while multiple programs are being executed, all active programs are halted:

- Press the hold button on the teach pendant or operator panel.
- Press the emergency stop button on the teach pendant or operator panel.
- Teach pendant is enabled when programs are started and executed by anything but teach pendant.
- Teach pendant is disabled, release the shift key, or release the deadman switch when programs are started and executed by teach pendant.
- Turn off the instantaneous stop signal (*IMSTP), halt signal (*HOLD), safety speed signal (*SFSPD), or operation enable signal (*ENBL).

When you select [ABORT] from the function menu, all active or halted programs are forcibly terminated. Among the multiple programs being executed, you can halt or forcibly terminate only the specified program on the monitor screen, as described below.

Press the "MONITOR" (F4) key on the program list screen. The monitor screen appears. Place the cursor on a program you want to halt or forcibly terminate and press the "PAUSE" (F3) key or "ABORT" (F4) key. Only the specified program is halted or forcibly terminated.



9.27.5 Cycle Stop Signal (CSTOPI)

The cycle stop signal can be set to forcibly terminate active or halted programs externally. This setting can be made on the system configuration screen.

Use [6 SYSTEM] to display the system configuration screen. When you set "CSTOPI for ABORT" to "TRUE", the currently selected program is terminated by the cycle stop signal. When you set "Abort all program by CSTOPI" to "TRUE", all programs can be forcibly terminated.

9.28 ERROR SEVERITY TABLE

9.28.1 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.

9.28.2 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 Error Severity Table(Fig.9.28.2).

Error Severity					JOINT 10%
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

Fig. 9.28.2 Error severity table

The default number of entries is 20, but can be extended to 999. Refer to Table9.28.2 for a listing and description of the items on the severity table that you can modify or edit for each kind of error.

Table 9.28.2 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. Refer to for a complete listing of facility names and codes.
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 for a complete listing of error codes and descriptions.

Item	Description
Sever (SeverityLevel)	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. Refer to for information about setting up the error code output 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 Modifying the Error Severity Table(Procedure9-26) to modify the error severity table. Use Procedure Modifying the Number of Entries in the Error Severity Table(Procedure9-27) to modify the number of entries in the error severity table.

Procedure 9-26 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.

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					JOINT 10%
Error Severity Table					
#	FCode	FName	Ecode	Sever	Erlog
1	11	SRVO	6	STOP	DEFAULT
2	11	SRVO	4	STOP	ACTIVE
3	10	FLPY	2	DEFAULT	DEFAULT
4	10	FLPY	3	DEFAULT	DEFAULT
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:**

- 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.
- 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.
- 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	

- Move the cursor to the desired severity level and press ENTER.
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.

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	

- Move the cursor to the desired error logger functionality and press ENTER.
- To verify that the proper error code was entered, press F2, VERIFY.
- To display help information, press F5, HELP. When you are finished, press PREV to return to the Error Severity Table screen.
- 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 9-27 Modifying the number of entries in the error severity table

Steps

- Perform a Controlled start. Refer to for information on performing a Controlled start.
- Press MENUS.
- Press 0, NEXT, and select Program Setup. You will see a screen similar to the following.

Program Limits		CTRL START MENU
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).

9.29 DIAGNOSTIC LOG

Overview

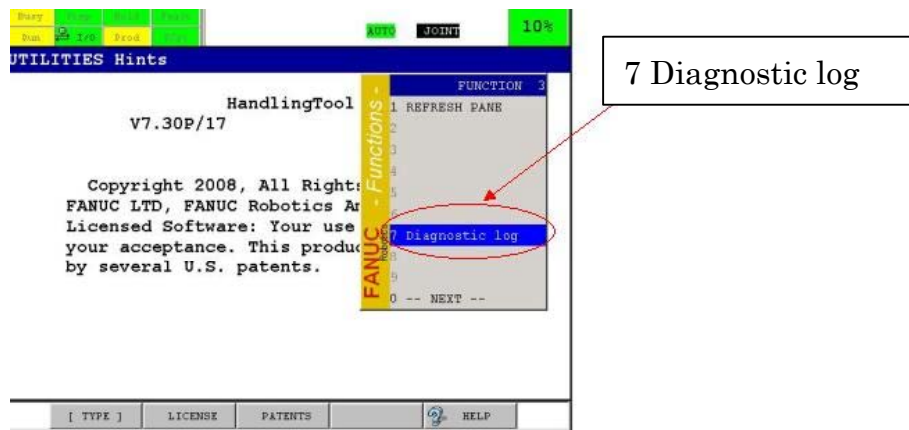
To get necessary data with easy operation while occurring some problems and utilize for investigation. It will be realized early resolution of problems and reduction of investigating time.

Some necessary data to investigate the problems are deleted after power off. This function enables to save these data to FROM.

Operations:

Do below operations **BEFORE POWER OFF** at the time of a trouble occurs.

- 1 Press FCTN key on TP and select “0 – Next –“ -> “0 –Next –“ -> “7 Diagnostic log”
- 2 “Diagnostic log” is displayed on 3rd page in FCTN menu. (See following sample figure)
- 3 Set the cursor onto “7 Diagnostic log” and press Enter key.
- 4 It takes about 1 minute to save diagnostic log.
- 5 There is no need to insert an external memory device such as memory card because data is saved to FROM.



sample figure

Please save image backup after done the diagnostic log. Diagnostic log is included into saved images.

If you can not save the images because it is difficult to cycle power the controller, please save maintenance data by following operation.

1. Insert MC: to the controller for backup.
2. Select menu - file.
3. Select F5[UTIL] - Set Device - Mem Card (MC).
4. Select F4[BACKUP] - Maintenance data.
5. After confirmation message is displayed, select F4[YES].
6. Maintenance data is saved into MNT_DATA¥ under the current directly. For instance, it is saved into MC:¥MNT_DATA¥ in the above case.

Maintenance data is able to be used from 7da3P30, 7da4P12 or 7da5P04.

Limitation:

- Depending on the condition of occurring problem, this function is not available. For example, in the case TP is frozen and any key operation are ignored.
- Data is not saved when there is not enough empty capacity in FROM. Then warning, “ Lack of FRA for saving”, is displayed.
- If you execute this function again, data is overwritten. Diagnostic log is saved only 1 time.

Caution

If you execute this function, verify that the status of robot is STOP. In some situations robot becomes jerky motion while saving log.

9.30 AUTOMATIC ERROR RECOVERY FUNCTION FOR MULTI-TASK

The automatic error recovery function for multi-task is applied to the 7DA4 series or later software.

9.30.1 Overview

This section consists of the following items:

- 9.30.2 Automatic error recovery function
- 9.30.3 Resume program function
- 9.30.4 Fast exit/entry feature
- 9.30.5 RESUME_PROG instruction
- 9.30.6 RETURN_PATH_DSBL instruction
- 9.30.7 MAINT_PROG instruction
- 9.30.8 Setting of the automatic error recovery function
- 9.30.9 Manual operation screen of the resume program function
- 9.30.10 Execution of the resume program from the teach pendant and test mode
- 9.30.11 Changing conditions for executing the resume program
- 9.30.12 Other specifications and restrictions
- 9.30.13 Warnings (Be sure to read this section for safety.)

This function is an optional function. (A05B-2500-J664)

9.30.2 Automatic Error Recovery Function

Robots are sometimes stopped by various alarms even during production. If a robot is stopped, it is necessary to perform recovery operation then resume the program that was originally running. For example, suppose that a robot is performing arc welding. An alarm due to an arc start failure may be issued, stopping the robot. In such a case, the operator must jog the robot to a safe position to, for example, cut the end of the wire or clean the nozzle, then resume the original program.

The automatic error recovery function is provided to support automatic operation of the above sequence.

The automatic error recovery function consists of the following two functions.

- Resume program function
- Fast exit/entry feature

9.30.3 Resume Program Function

This function is a function to remove the cause of the alarm by using the resume program when the robots are stopped by the alarm when being producing, and to resume the original program.

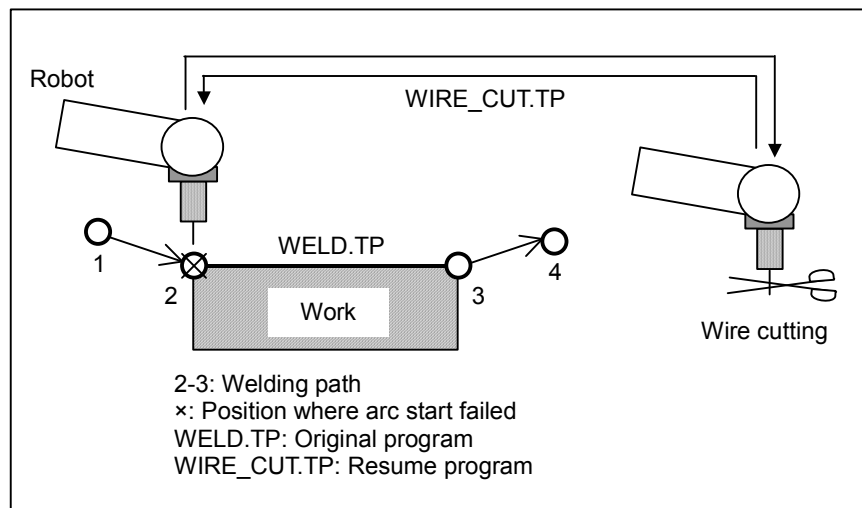


Fig.9.30.3 Resume program function

In the example shown above, the robot is operated by executing WELD.TP to perform welding along the path from 2 to 3.

Assume that an arc start failure occurs at the arc start position 2. At this time, if the resume program function is used, another program called the resume program, which is WIRE_CUT.TP in this case, can be started at the next start signal input. After this program terminates, another start signal input resumes the original program. If the resume operation function is then enabled (which is set on the welding system setting screen), the robot automatically returns to the original position where the robot was stopped, and then the original program is resumed. If the return distance for resume operation is set, the robot returns from the stop position by the set distance, then the original program is resumed. If no arc is produced, a scratch start takes place.

9.30.4 Fast Exit/entry Feature

If an alarm is issued during operation in a complicated environment, the robot moves from the stopped position to the taught point to execute the resume program. In this case, the robot may interfere with part of a workpiece or peripheral devices. After recovery operation, similar interference may occur when an attempt is made to execute the original program. The fast exit/entry feature is provided to avoid the possibility of such interference.

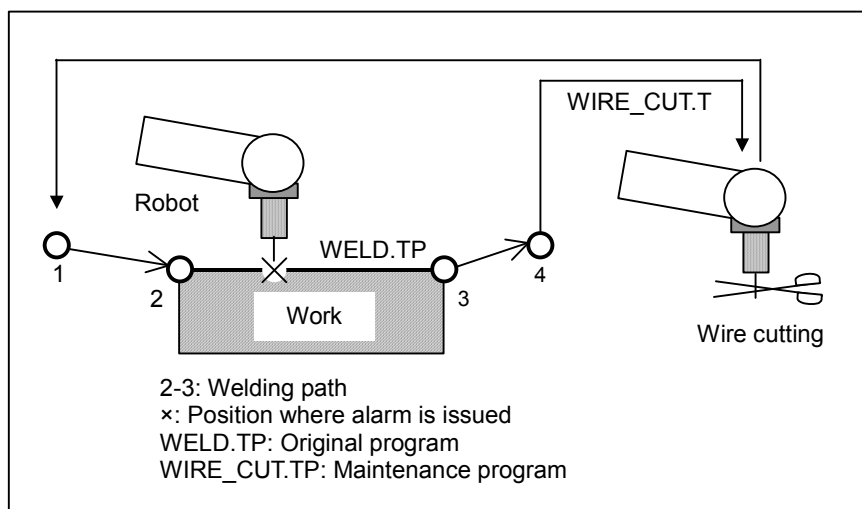


Fig.9.30.4 Fast exit/entry feature

In the example shown above, the robot is operated by executing WELD.TP to perform welding along the path from 2 to 3. At this time, when the alarm of something is issued between positions 2-3 and the robot stops, the fast exit/entry feature automatically does the following operation.

- 1 From the stopped position, disable arc welding, and execute only the move statements of the original program up to the end.
- 2 Execute a maintenance program, which is WIRE_CUT.TP in this case.
- 3 Disable arc welding, execute the move statements of the original program from the beginning to move the robot to the stopped position.
- 4 Enable arc welding, and resume the original program operation.

Thereafter, in this operation manual, above-mentioned operation 1-4 is called FFR sequence (Fast Fault Recovery sequence).

9.30.5 RESUME_PROG Instruction

The resume program function executes a resume program defined in an original program, in lieu of the original program. The resume program instructions to define the resume program are the following five instructions. RESUME_PROG[1] can be used in single-task/multi-task, and RESUME_PROG[2-5] can be used in the multi-task.

- RESUME_PROG[1:Comment] : For Single-task/Multi-task
- RESUME_PROG[2:Comment] : For Multi-task
- RESUME_PROG[3:Comment] : For Multi-task
- RESUME_PROG[4:Comment] : For Multi-task
- RESUME_PROG[5:Comment] : For Multi-task

To erase the defined resume program, use the following instruction.

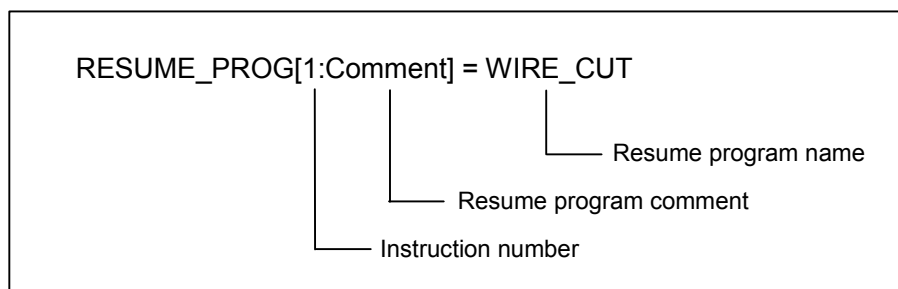
- CLEAR_RESUME_PROG

To teach the resume instructions, click F1 [INST] to display the related submenu, then select [Program control] from the submenu.

Instruction	
1 Miscellaneous	5 Offset/Frames
2 Weave	6 Multiple control
3 Skip	7 Program control
4 Payload	8 ---next page---

RESUME_PROG[1:Comment]

The program name defined by RESUME_PROG[1:Comment] is registered as a resume program when RESUME_PROG[1:Comment] is executed in original program, and when the alarm is issued while defining the resume program, the resume program is executed.



To use the resume program instructions, the following option is needed.

- To use the RESUME_PROG[1-5], the Auto error recovery function (A05B-2500-J664) is needed.
- In addition, to use the RESUME_PROG[2-5], the following options are needed.

RESUME_PROG[2-3]

Multi-Arm Pkg	A05B – 2500 – R671

RESUME_PROG[2-5]

Multi-Arm Pkg	A05B – 2500 – R671
Triple/Quad Arm Pkg	A05B – 2500 – R673

When the instruction number of RESUME_PROG[] is defined, it is necessary to accord the following conditions.

TP Program that can be started by START#1/PROD_STAR#1 of UOP	RESUME_PROG[1:Comment]
TP Program that can be started by START#2/PROD_STAR#2 of UOP	RESUME_PROG[2:Comment]
TP Program that can be started by START#3/PROD_STAR#3 of UOP	RESUME_PROG[3:Comment]
TP Program that can be started by START#4/PROD_STAR#4 of UOP	RESUME_PROG[4:Comment]
TP Program that can be started by START#5/PROD_STAR#5 of UOP	RESUME_PROG[5:Comment]

(To use the START#*/PROD_STAR#*, the Multi-Arm Pkg(A05B-2500-R671) is needed.)

For instance, in multi-task, the RESUME_PROG[3:Comment] can be used only by the program that can be started by START#3/ PROD_STAR#3 of UOP.

The RESUME_PROG[2-5] cannot be used in the single-task because it is for multi-task. Therefore, use the RESUME_PROG[1:Comment] in the single-task.

To define the resume program instructions, there are the following conditions.

- Multiple resume program instructions cannot be defined in the original program.
- The resume program instructions cannot be defined in the resume/maintenance program.
- A resume program instruction which has the same instruction number cannot be executed by the plural. For instance, the RESUME_PROG[5] cannot be executed by other programs while the resume program is already being defined by the RESUME_PROG[5]. However, the RESUME_PROG[1-4] can be executed by other programs.
- The resume program instructions cannot be defined in the programs executed by the RUN instruction.

CLEAR_RESUME_PROG

When the CLEAR_RESUME_PROG is executed, the resume program defined by RESUME_PROG[] is erased. Therefore, even if the alarm is issued after the CLEAR_RESUME_PROG is executed, the resume program is not executed. Moreover, the CLEAR_RESUME_PROG is effective only in the program in which the CLEAR_RESUME_PROG is defined. For instance, in multi task, when the CLEAR_RESUME_PROG is executed in the program in which the RESUME_PROG[1] is defined, only the resume program defined by the RESUME_PROG[1] is erased. At this time, the resume program defined by the RESUME_PROG[2-5] is never erased.

The resume program is erased also when:

- Backward execution is performed in original program.
- The cursor line is changed manually in original program.
- The original program terminates.

**CAUTION**

When the definition of the resume program is erased while the resume program is pausing, the resume program cannot be restarted. The original program restarts from the pause position of the original program if you restart the program at this time.

The following programs are examples that use the RESUME_PROG[2] and the CLEAR_RESUME_PROG.

WELD.TP
Motion group: [*,1,*,*,*,*,*]

WELD	JOINT 10%
	1/7
1: J P[1] 40% FINE	
2: RESUME_PROG[2:WIRE CUT]=WIRE CUT	
3: L P[2] 300mm/sec FINE	
: Arc Start[2]	
4: L P[3] 50mm/sec FINE	
: Arc End[2]	
5: CLEAR_RESUME_PROG	
6: J P[4] 40% FINE	
[End]	

WIRE_CUT.TP
Motion group: [*,1,*,*,*,*,*]

WIRE_CUT	JOINT 10%
	1/8
1: L P[1] 50mm/sec FINE	
2: J P[2] 50% FINE	
3: WO[4]=PULSE, 0.5sec	
4: L P[3] 20mm/sec FINE	
5: L P[4] 50mm/sec FINE	
6: WAIT 0.8sec	
7: J P[5] 50% FINE	
[End]	

← Wire feed

← Wire cut

In the above program example, the WIRE_CUT.TP is taught in the second line of the WELD.TP and is erased in the fifth line. Since the WIRE_CUT.TP is defined as the resume program between the third to fourth lines, it is executed as the resume program. In the sixth and subsequent lines, the resume program has been erased, so the resume program is not executed.

9.30.6 RETURN_PATH_DSBL Instruction

In arc tool systems, the resume operation function is generally enabled. With this function enabled, return to the original stop position is always performed then arc is produced when the original program resumes after the resume program terminates. In some systems, however, return to the original stop position should not sometimes be performed. For example, when the nozzle touch state is input through DI, a resume program is used to relieve the torch slightly in the torch direction. If the resume operation function operates, return to the original stop position is performed even when relieve operation has been performed. As a result, the nozzle touch state is observed again. In such a case, the resume operation function needs to be kept enabled, but it should be disabled only after the execution of the resume program.

This can be performed with the RETURN_PATH_DSBL instruction. By using this instruction within the resume program, the resume operation function can be disabled only when the original program is resumed next. This instruction is valid only when it is executed within a resume program; the instruction is invalid when executed in a program other than the resume program.

The RETURN_PATH_DSBL instruction appears in the menu containing the RESUME_PROG[] instruction.

Instruction	
1 Miscellaneous	5 Offset/Frames
2 Weave	6 Multiple control
3 Skip	7 Program control
4 Payload	8 ---next page---

RETURN_PATH_DSBL

The RETURN_PATH_DSBL is effective only in the resume program in which the RETURN_PATH_DSBL is defined. For instance, in multi task, when RETURN_PATH_DSBL is executed in the resume program defined by RESUME_PROG[1], only the resume operation function of the original program in which RESUME_PROG[1] is defined becomes invalid. The resume operation function of the original program in which other RESUME_PROG[2-5] is defined is effective.

The following programs are examples that use the RETURN_PATH_DSBL. The RETURN_PATH_DSBL instruction is valid only when it is taught within resume program. Use this instruction as shown in the sample program given below. If the instruction is taught as shown below, the resume operation function does not operate when the original program resumes after the resume program terminates, even if the resume operation function is enabled.

WELD.TP
Motion group: [*,1,*,*,*,*,*]

```

WELD                                JOINT 10%
                                1/7
1: J P[1] 40% FINE
2:  RESUME_PROG[2:WIRE CUT]=WIRE CUT
3: L P[2] 300mm/sec FINE
   : Arc Start[2]
4: L P[3] 50mm/sec FINE
   : Arc End[2]
5:  CLEAR_RESUME_PROG
6: J P[4] 40% FINE
[End]

```

WIRE_CUT.TP
Motion group: [*,1,*,*,*,*,*]

```

WIRE_CUT                            JOINT 10%
                                1/9
1: L P[1] 50mm/sec FINE
2: J P[2] 50% FINE
3:  WO[4]=PULSE, 0.5sec           ← Wire feed
4: L P[3] 20mm/sec FINE
5: L P[4] 50mm/sec FINE
6:  WAIT 0.8sec
7: J P[5] 50% FINE                ← Wire cut
8:  RESUME_PATH_DSBL
[End]

```

9.30.7 MAINT_PROG Instruction

The fast exit/entry feature automatically does the following operation.

- 1 From the stopped position, disable arc welding, and execute only the move statements of the original program up to the end.
- 2 Execute a maintenance program.
- 3 Disable arc welding, execute the move statements of the original program from the beginning to move the robot to the stopped position.
- 4 Enable arc welding, and resume the original program operation.

Thereafter, in this operation manual, above-mentioned operation 1-4 is called FFR sequence (Fast Fault Recovery sequence).

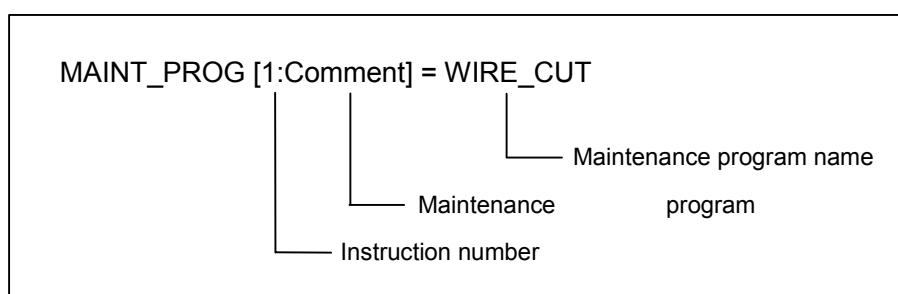
The maintenance program instructions to define the maintenance program are the following five instructions. MAINT_PROG[1] can be used in single-task/multi-task, and MAINT_PROG[2-5] can be used in the multi-task.

- MAINT_PROG[1:Comment] : For Single-task/Multi-task
- MAINT_PROG[2:Comment] : For Multi-task
- MAINT_PROG[3:Comment] : For Multi-task
- MAINT_PROG[4:Comment] : For Multi-task
- MAINT_PROG[5:Comment] : For Multi-task

The MAINT_PROG[] instruction appears in the menu containing the RESUME_PROG[] instruction.

Instruction	
1 Miscellaneous	5 Offset/Frames
2 Weave	6 Multiple control
3 Skip	7 Program control
4 Payload	8 ---next page---

MAINT_PROG[1:Comment]



The program name defined by MAINT_PROG[1:Comment] is registered as a maintenance program when MAINT_PROG[1:Comment] is executed in original program, and when the alarm is issued while defining the maintenance program, the FFR sequence is executed.

To use the maintenance program instructions, the following option is needed.

- To use the MAINT_PROG[1-5], the Auto error recovery function (A05B-2500-J664) is needed.
- In addition, to use the MAINT_PROG[2-5], the following options are needed.

MAINT_PROG[2-3]

Multi-Arm Pkg	A05B – 2500 – R671

MAINT_PROG[2-5]

Multi-Arm Pkg	A05B – 2500 – R671
Triple/Quad Arm Pkg	A05B – 2500 – R673

When the instruction number of MAINT_PROG[] is defined, it is necessary to accord the following conditions.

TP Program that can be started by START#1/PROD_STAR#1 of UOP	MAINT_PROG[1:Comment]
TP Program that can be started by START#2/PROD_STAR#2 of UOP	MAINT_PROG[2:Comment]
TP Program that can be started by START#3/PROD_STAR#3 of UOP	MAINT_PROG[3:Comment]
TP Program that can be started by START#4/PROD_STAR#4 of UOP	MAINT_PROG[4:Comment]
TP Program that can be started by START#5/PROD_STAR#5 of UOP	MAINT_PROG[5:Comment]

(To use the START#*/PROD_STAR#*, the Multi-Arm Pkg(A05B-2500-R671) is needed.)

For instance, in multi-task, the MAINT_PROG[3:Comment] can be used only by the program that can be started by START#3/ PROD_STAR#3 of UOP.

The MAINT_PROG[2-5] cannot be used in the single-task because it is for multi-task. Therefore, use the MAINT_PROG[1:Comment] in the single-task.

When the FFR sequence is completed, the following messages are displayed on the teach pendant.

MOTN -431	Fast fault recovery 1	When the FFR sequence executed by the MAINT_PROG[1] is completed.
MOTN -432	Fast fault recovery 2	When the FFR sequence executed by the MAINT_PROG[2] is completed.
MOTN -433	Fast fault recovery 3	When the FFR sequence executed by the MAINT_PROG[3] is completed.
MOTN -434	Fast fault recovery 4	When the FFR sequence executed by the MAINT_PROG[4] is completed.
MOTN -435	Fast fault recovery 5	When the FFR sequence executed by the MAINT_PROG[5] is completed.

To define the maintenance program instructions, there are the following conditions.

- Multiple maintenance program instructions cannot be defined in the original program.
- The maintenance program instructions cannot be defined in the resume/maintenance program.
- A maintenance program instruction which has the same instruction number cannot be executed by the plural. For instance, the MAINT_PROG[5] cannot be executed by other Programs while the maintenance program is already being defined by the MAINT_PROG[5]. However, the MAINT_PROG[1-4] can be executed by other Programs.
- The maintenance program instructions cannot be defined in the programs executed by the RUN instruction.

The maintenance program is erased also when:

- Backward execution is performed in original program or maintenance program.
- The cursor line is changed manually in original program or maintenance program.
- The original program or the maintenance program terminates.



CAUTION

When the definition of the maintenance program is deleted while executing the FFR sequence, the maintenance program or the original program cannot be restarted by the FFR sequence.

The following programs are examples that use the MAINT_PROG[2].

WELD.TP
Motion group: [*,1,*,*,*,*,*]

WELD	JOINT 10%
	1/7
1: J P[1] 40% FINE	
2: MAINT_PROG[2:WIRE CUT]=WIRE_CUT	
3: L P[2] 300mm/sec FINE	
: Arc Start[2]	
4: L P[3] 50mm/sec FINE	
: Arc End[2]	
5: CLEAR_RESUME_PROG	
6: J P[4] 40% FINE	
[End]	

WIRE_CUT.TP
Motion group: [*,1,*,*,*,*,*]

WIRE_CUT	JOINT 10%
	1/8
1: L P[1] 50mm/sec FINE	
2: J P[2] 50% FINE	
3: WO[4]=PULSE, 0.5sec	← Wire feed
4: L P[3] 20mm/sec FINE	
5: L P[4] 50mm/sec FINE	
6: WAIT 0.8sec	← Wire cut
7: J P[5] 50% FINE	
[End]	

In the above program example, the WIRE_CUT.TP is taught in the second line of the WELD.TP. Since the WIRE_CUT.TP is defined as the maintenance program between the third to final lines, it is executed as the maintenance program. In addition, when the execution of the WELD.TP ends, the definition of the maintenance program is automatically deleted.

9.30.8 Setting of the Automatic Error Recovery Function

On the setting screen of the automatic error recovery function, the following settings can be made:

- Instruction number : Selecting the instruction number

Error recovery function common setup

- Error Recovery Function : Enabling/disabling the automatic error recovery function
- Approval DI Index No.* : Defining the recovery switch DI
- Incomplete End DO Index No.* : Defining the incomplete end DO
- Reset DI Index No.* : Defining the incomplete-end reset DI
- Automatic Start Feature : Enabling/disabling the alarm-time automatic start feature
- Alarm setting screen : Enabling/disabling the alarm code monitoring function
- DI_alarm setting screen : Defining the automatic error recovery alarm conditions

Resume program type recovery

- RESUME comment : Defining the resume program comment
- Status DO Index No.* : Defining the resume program information DO
- Auto Start Max Count : Setting the maximum number of automatic start repetitions
- Auto Start Max Count R[] : Setting the automatic start count register

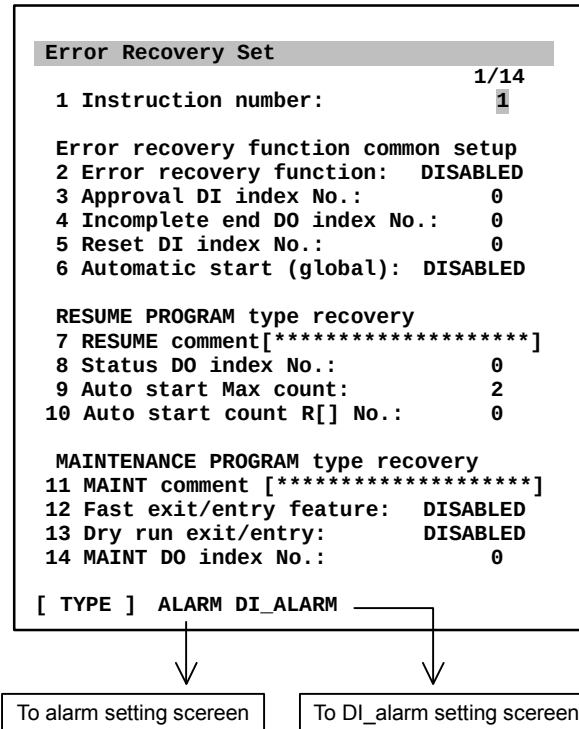
Maintenance program type recovery

- MAINT comment : Defining the maintenance program comment
- Fast Exit/Entry Feature : Enabling/disabling the fast exit/entry feature
- Dry Run Exit/Entry : Enabling/disabling dry run exit/entry operation
- MAINT DO Index No.* : Defining the FFR sequence information DO

(* Set this to zero if you do not want to use this feature.)

To display the setting screen of the automatic error recovery function, first press MENU to display the screen menu, then select "6 Set up". Then, press F1, [TYPE] to display the screen switching menu, then select Err recovery.

[MENU] → 6 Setup → Err recovery



Selecting the instruction number

This screen can be displayed by each instruction number of the resume program instruction and the maintenance program instruction. It is possible to switch to a setup screen of other instruction numbers by changing this item.

For multi-task

For instance, instruction number 1-3 can be selected by this item in the system that can use RESUME_PROG[1-3] and MAINT_PROG[1-3].

Error recovery function common setup

Enabling/disabling the automatic error recovery function

This item enables or disables the automatic error recovery function. When the automatic error recovery function is enabled, and neither monitored alarm codes nor the recovery switches DI are defined, the resume program is always executed at restart from the suspended state (except when the resume program information DO is off). And, when the automatic error recovery function and the fast exit/entry feature are enabled, and neither monitored alarm codes nor the recovery switch DI are defined, the FFR sequence is always executed at restart from the suspended state(except when the FFR sequence information DO is off).

When this item is disabled, the resume program and the FFR sequence is not executed.

For multi-task

This item can be set by each instruction number.

Defining the recovery switch DI

To use the recovery switch DI function, define an DI number. After the number is defined, power must be turned off then back on.

With this function, the operator can choose whether to execute the resume program or the FFR sequence or not at the time of restart from the suspended state by using a peripheral device. When DI is OFF, neither the resume program nor the FFR sequences are executed. If this signal number is not defined, this function is disabled.

The specifications of the recovery switch DI function are listed below.

Table.9.30.8 (a) Specifications of the recovery switch DI function

DI number definition	Recovery switch DI function status	DI status	Execution of resume program or FFR sequence at restart
0	Disabled	—	Executed
Valid number defined	Enabled	On	Executed
		Off	Not executed

**CAUTION**

To continue a resume program at program restart after the resume program is suspended, input the on state of the recovery switch DI. If it is off, the original program is executed.

For multi-task

This item can be set by each instruction number.

Defining the incomplete end DO

When an incomplete end DO number is defined, the incomplete end DO is output if a certain forced termination alarm is issued during execution of the resume program or the maintenance program. The output incomplete end DO is turned off by the next start signal input.

Before inputting the start signal, the operator must check the incomplete end DO signal status. If this signal is on, the resume program or the maintenance program terminates in the middle, so the robot is not in a specified position.

**CAUTION**

When the resume program or the maintenance program terminates in the middle, and the start signal is input, the robot might do action not predictable. Therefore, before inputting the start signal, check the program name and the program start line that will be executed with the next start signal. In addition, if an interfering object exists, jog the robot to a position where interfering object doesn't exist, then input the start signal.

This signal may be added to the PLC start signal acceptance conditions.

If this signal is set to 0, this function is disabled.

For multi-task

This item can be set by each instruction number.

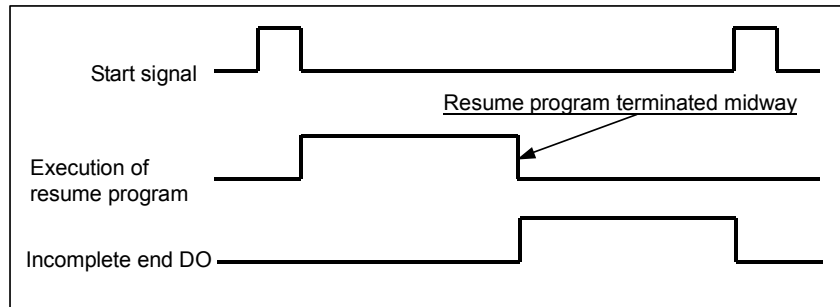


Fig. 9.30.8 (a) Incomplete End DO Output Timing Chart

Defining the incomplete-end reset DI

When the incomplete end DO is included in the PLC start signal acceptance conditions, the operator requires a means to turn off the incomplete end DO externally. Inputting the incomplete-end reset DI signal turns off the incomplete end DO. When the incomplete end DO is output, the operator must first check the program name and the program start line that will be executed with the next start signal. In addition, if an interfering object exists, jog the robot to a position where interfering object doesn't exist, input the incomplete-end reset DI, then input the start signal.

If this signal is set to 0, this function is disabled.

For multi-task

This item can be set by each instruction number.

Enabling/disabling the alarm-time automatic start feature

When this item is enabled, and a alarm defined in the monitored alarm codes is issued, the resume program or the FFR sequence are automatically executed without outputting the alarm signal and stopping the robot. When the resume program has terminated, the original program is resumed automatically by this function (In the fast exit/entry feature, when the maintenance program has terminated, the original program is resumed automatically by FFR sequence.). During then, the operator need not input the start signal.

Because the alarm signal is not output, other robots are not stopped when multiple robots are operating. The robot for which the alarm was issued moves by itself to the recovery station, and after recovery work, the original program is resumed.

When this item is disabled, and a alarm defined in the monitored alarm codes is issued, the program outputs an alarm signal and stops running. Input of the start signal executes the defined resume program or FFR sequence. In the resume program function, after the resume program terminates, another start signal input restarts the suspended original program. Another start signal to resume an original program need not be input for the fast exit/entry feature. The original program is resumed automatically by FFR sequence.

For multi-task

- This item cannot be set by each instruction number. When this item is enabled in either of instruction number, this function of all the instruction numbers are automatically enabled.
- Even if this function is enabled, this function is not executed when the alarm defined in the monitored alarm codes is issued while the alarm is being issued in other robots. In that case, the program outputs an alarm signal and stops running. Input of the start signal executes the defined resume program or FFR sequence.

Enabling/disabling the alarm code monitoring function

When this function is enabled, and only when a alarm defined in the monitored alarm codes is issued, the resume program or the FFR sequence are executed. When the monitored alarm code is defined, and the resume program or the maintenance program are defined, this function is enabled automatically. If no

alarm code is defined, this function is disabled. In this case, before the suspended original program is resumed, the resume program or the FFR sequence are always executed(except when the recovery switch DI function is enabled).

To define alarm codes to be monitored, press the F2 (ALARM) key. A screen for defining alarm codes is displayed.

Error Recovery Setup:1		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	
10 Monitored alarm code	0	
[TYPE]	DONE	HELP

When a defined alarm code is issued, and a program is suspended, the resume program or FFR sequence are executed at restart.

Each alarm code consists of an alarm code ID and alarm number. The alarm code ID indicates the type of alarm. For an arc start failure alarm, for example, the following alarm code is indicated:

<u>ARC</u>	–	<u>013</u>	Arc Start failed	=	<u>53013</u>
ID (53)		Number			ID Number

For alarm numbers, refer to the alarm code table in the operator's manual.

Up to ten alarm codes can be defined as standard. To change the maximum number of alarm codes (up to 20 codes) that can be defined, change system variable \$RSMPRG_SV.\$NUM_ALARM, turn the power off then on.

Instruction number	System variable
1	\$RSMPRG_SV.\$NUM_ALARM
2	\$RSMPRG_SV2.\$NUM_ALARM
3	\$RSMPRG_SV3.\$NUM_ALARM
4	\$RSMPRG_SV4.\$NUM_ALARM
5	\$RSMPRG_SV5.\$NUM_ALARM

Pressing the F5 (HELP) key displays the following screen:

Error Recovery Setup:1	
HELP	Arrows to scroll, PREV to exit
Typical alarm code IDs are specified as follows.	
PROG : 3,	SRVO : 11, INTP : 12
PRI0 : 13,	MOTN : 15, SP0T : 23
SYST : 24,	PALT : 26, LASR : 50
SEAL : 51,	ARC : 53, MACR : 57
SENS : 58,	COMP : 59

If the following conditions are met, the alarm code monitoring function is disabled:

- The recovery switch DI is off.
- There is no alarm code defined.
- The resume program or the maintenance program are not defined in the program currently being executed.
- The resume program or the FFR sequence is executing.

The specifications of the alarm code monitoring function are listed below.

Table.9.30.8 (b) Specifications of the Alarm Code Monitoring Function

Alarm code definition	Alarm code function status	Issuance of defined alarm	Execution of resume program or FFR sequence at restart
0	Disabled	—	Executed
At least one alarm code is defined	Enabled	Issued	Executed
		Not issued	Not executed



CAUTION

Defined alarms must have the suspension alarm attribute. Do not define any warning alarm as an alarm code.

For multi-task

This item can be set by each instruction number.

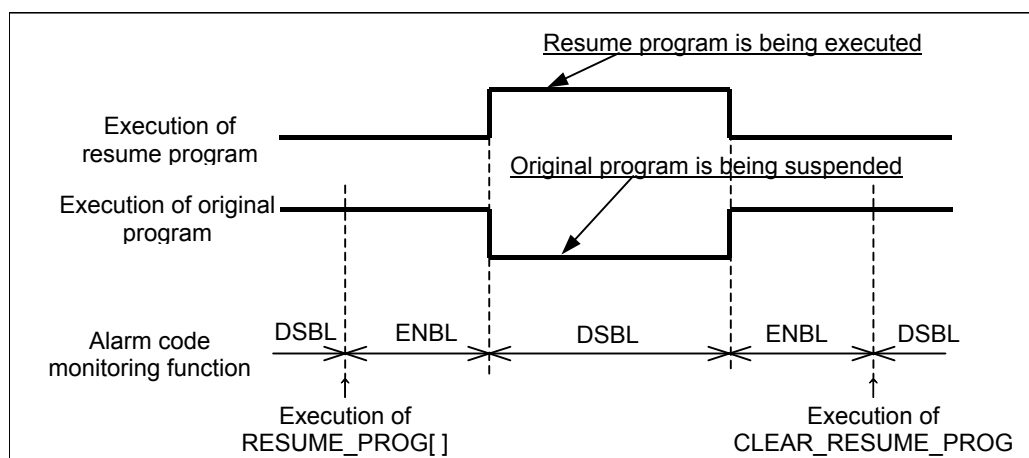


Fig. 9.30.8 (b) Enabling/disabling the alarm code monitoring function in the resume program function

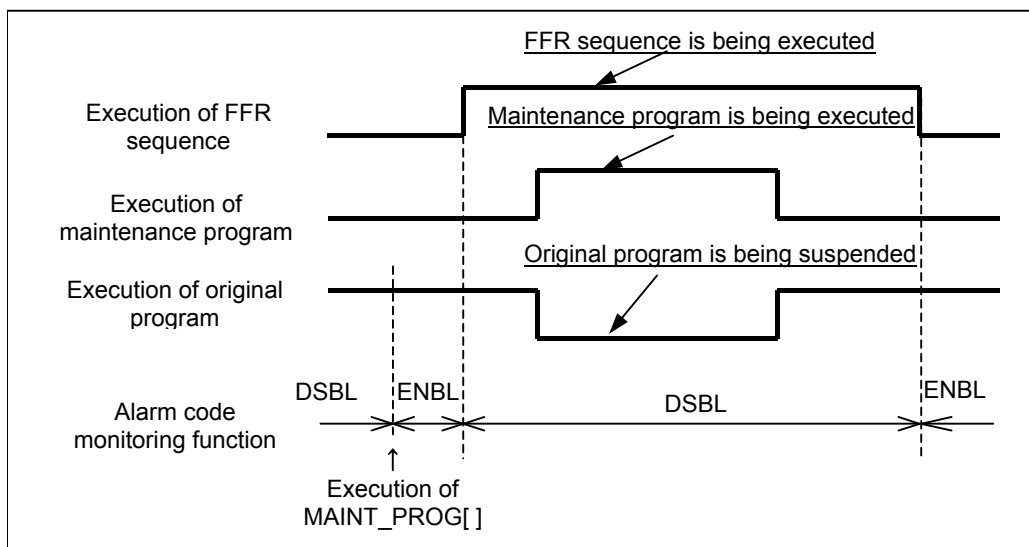


Fig. 9.30.8 (c) Enabling/disabling the alarm code monitoring function in the fast exit/entry feature

Defining automatic error recovery alarm conditions

By inputting a defined digital input signal, an automatic error recovery alarm can be issued. When this alarm is defined for the alarm-time automatic start feature, the resume program or the FFR sequence can be executed automatically by inputting the digital input signal.

As the message for an automatic error recovery alarm, a message defined for a user alarm can be used. The alarm severity can be set to either LOCAL or GLOBAL selectively. When LOCAL is selected, the alarm is issued only for a program that defines the resume program or the maintenance program. The status of a digital input signal to be monitored can be set by selecting the signal type from among DI, RI, and WI, changing the signal number, and selecting the trigger status between on and off.

Define the conditions for issuing an automatic error recovery alarm on the definition screen that is displayed by pressing F3 (DI_ALARM).

Error Recovery Setup				1/3
UALM	Severity	Type	Value	
1 [1]	LOCAL	DI[3]	ON	
2 [5]	GLOBAL	RI[6]	OFF	
3 [10]	LOCAL	DI[2]	ON	
[TYPE]	DONE	HELP		

On this screen, the items shown below can be set. The alarm code of the automatic error recovery alarm is 12278.

- **User alarm number**
When the automatic error recovery alarm is issued, the user alarm message with the set number is displayed as an alarm message. When this item setting has been changed, the new setting becomes effective immediately.
- **Alarm severity**
This item can choose whether the automatic error recovery alarm is a local alarm or global alarm. When LOCAL is set, the automatic error recovery alarm is issued only for the program that defines the resume program or the maintenance program. If there is no program that defines the resume program or the maintenance program, the alarm is regarded as a global alarm. When this item setting has been changed, the new setting becomes effective immediately.

- **Signal type**
Choose the type of the digital signal for issuing the automatic error recovery alarm from among DI, RI, and WI. When this item setting has been changed, the power must be turned off then back on for the new setting to become effective.
- **Signal number**
Set the number of the digital signal for issuing the automatic error recovery alarm. When this setting has been changed, the power must be turned off then back on for the new setting to become effective.
- **Detection signal status**
Set the status of the digital signal for issuing the automatic error recovery alarm to ON (high) or OFF (low). When this setting has been changed, the power must be turned off then back on for the new setting to become effective.

The standard number of automatic error recovery alarm conditions is three. This number can be increased to up to five by changing system variable \$RSMPRG_SV.\$NUM_DI_ALM. After this system variable has been changed, the power must be turned off then back on for the new setting to become effective.

For multi-task

- This item cannot be set by each instruction number. When this item is enabled in either of instruction number, this function of all the instruction numbers are automatically enabled.
- When the alarm severity issues the automatic error recovery alarm of LOCAL alarm when there are multiple programs that define the resume program or the maintenance program, the automatic error recovery alarm of the number of programs is issued.
- Define the automatic error recovery alarm code in all the instruction numbers that can be used when the automatic start feature is enabled, and the automatic error recovery alarm code is defined in the monitored alarm codes. When LOCAL alarm is defined in the automatic error recovery alarm only by the limited instruction number, even if the LOCAL alarm is issued while multiple programs are executed at the same time, the automatic start feature might not be able to be executed.

Resume program type recovery

Defining the resume program comment

A comment can be specified for the resume program instruction. Move the cursor to this item and press the enter key. The screen enters comment input mode. A resume program comment is used to describe additional information to be displayed on the program edit screen together with the resume program instruction.

For multi-task

This item can be set by each instruction number.

Defining the resume program information DO

When the alarm code monitoring function and recovery switch DI function are both disabled, the resume program is always executed at the time of restart after the original program is suspended. When both the functions are enabled, it is difficult to determine whether the original program or resume program is to be executed at restart.

The resume program information DO is on only when the resume program is executed at restart. When the signal is off, the original program is executed at restart. With this function, the operator can know which program is to be executed next.

If the following conditions are met, the resume program information DO goes on:

- The automatic error recovery function is enabled.
- Not the single step mode.
- The resume program is defined in the original program.
- The original program has a motion group.
- The original program is suspended, and the resume program is not yet completed.
- There is no optional function that disables the automatic error recovery function. See “Other specifications and restrictions.”
- The user condition parameter is true. See “Conditions for executing the resume program.”

RESUME_PROG instruction	User condition parameter
RESUME_PROG[1:Comment]	\$AUTORCV_ENB
RESUME_PROG[2:Comment]	\$AUTORCVENB2
RESUME_PROG[3:Comment]	\$AUTORCVENB3
RESUME_PROG[4:Comment]	\$AUTORCVENB4
RESUME_PROG[5:Comment]	\$AUTORCVENB5

- When the teach pendant is enabled:
 - The operation mode (on the automatic error recovery manual operation screen) is TP_TEST.
- When the teach pendant is disabled:
 - The operation mode (on the automatic error recovery manual operation screen) is AUTO.
 - The remote conditions are met when system variable \$RMT_MASTER is 0.
 - There is no alarm code defined. If any alarm code is defined, the alarm code is issued.
 - The recovery switch DI function is disabled. If this function is enabled, the recovery switch DI signal is on.

CAUTION

1. While the resume program is being executed, single step operation cannot be performed.
2. Even if the resume program information DO is on, the resume program is not executed when backward execution of the original program is performed.
3. Backward execution in the resume program is possible.
4. The update cycle period for the resume program information DO is 300 ms. When the conditions listed above have been changed, wait 300 ms before program execution.

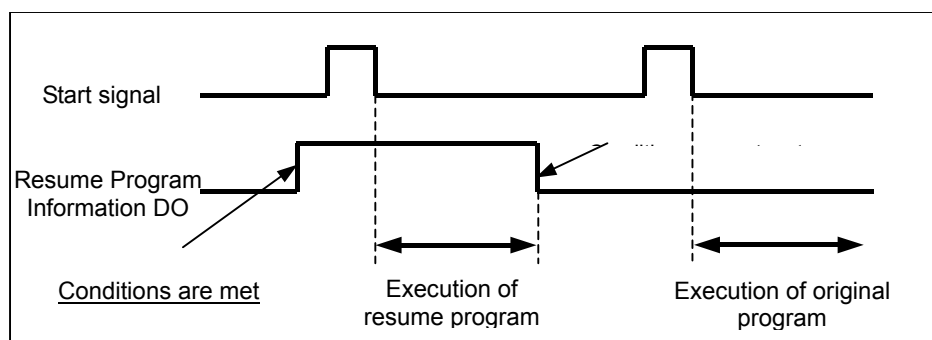


Fig. 9.30.8 (d) Resume Program Information DO Output Timing Chart

For multi-task

This item can be set by each instruction number.

Setting the maximum number of automatic start repetitions

When a defined alarm is issued, the alarm-time automatic start feature automatically executes the resume program, then resumes the original program. If the defined alarm is issued again when the original program is resumed, the automatic start feature functions again. For example, the automatic start feature is activated by an alarm indicating an arc start failure, then the same alarm is issued again when the original program has resumed.

To prevent such an endlessly repeated condition, set the maximum number of automatic start repetitions. The number of times the resume program is started repeatedly is counted internally. If the count exceeds the set value, "INTP-134 Over automatic start Max counter" is issued, and the resume program information DO is turned off at the same time. If this occurs, eliminate the cause of the alarm issued in the original program. Then input the start signal.

**CAUTION**

The number of repetitions counted internally is cleared when the execution of a move statement has terminated and when the CLEAR_RESUME_PROG instruction has been executed.

For multi-task

This item can be set by each instruction number.

Defining the automatic start count register

As mentioned above, the resume program may be executed several times repeatedly by the automatic start feature. When the automatic start count register is defined, a different program can be executed as the resume program each time the resume program is executed. For example, when the resume program is executed for the first time by the automatic start feature, the register value is 1. When an alarm is issued again during execution of the original program, and the resume program is then executed again by the automatic start feature, the register value is 2. By executing a different subprogram in the resume program according to the register value, different resume program operation can be performed each time the repetition count is incremented.

**CAUTION**

When the resume program is executed by other than the automatic start feature, the register value is 0. Therefore, a resume program must be created so that the same subprogram is called when the register value is 0 and when the value is 1.

For multi-task

This item can be set by each instruction number.

Maintenance program type recovery**Defining the maintenance program comment**

A comment can be specified for the maintenance program instruction. Move the cursor to this item and press the enter key. The screen enters comment input mode. A maintenance program comment is used to describe additional information to be displayed on the program edit screen together with the maintenance program instruction.

For multi-task

This item can be set by each instruction number.

Enabling/disabling the fast exit/entry feature

This item enables or disables the fast exit/entry feature.

When the automatic error recovery function and the fast exit/entry feature are enabled, and neither monitored alarm codes nor the recovery switch DI are defined, the FFR sequence is always executed at restart from the suspended state (except when the FFR sequence information DO is off).

When this item is disabled, the FFR sequence is not executed.

For multi-task

This item can be set by each instruction number.

Enabling/disabling dry run exit/entry operation

In the fast exit/entry feature, this item specifies whether exit from the stopped position and return to the stopped position after maintenance program execution are to be performed at dry run speed. This item can be set only for instruction number 1 in the single-task.

For multi-task

This item can be set by each instruction number.

Defining the FFR sequence information DO

When the alarm code monitoring function and recovery switch DI function are both disabled, the FFR sequence is always executed at the time of restart after the original program is suspended. When both the functions are enabled, it is difficult to determine whether the original program or the FFR sequence is to be executed at restart.

The FFR sequence information DO is on only when the FFR sequence is executed at restart. In addition, because the FFR sequence is not executed though the original program is executed at restart when the signal is off, the maintenance program is not executed after the original program is completed. With this function, the operator can know the FFR sequence is to be executed next.

If the following conditions are met, the FFR sequence information DO goes on:

- The automatic error recovery function is enabled.
- Not the single step mode.
- The maintenance program is defined in the original program.
- The resume program information DO is not on
- The FFR sequence is being executed.
- The fast exit/entry feature is enabled.
- When the teach pendant is disabled:
 - The remote conditions are met when system variable \$RMT_MASTER is 0.
 - There is no alarm code defined. If any alarm code is defined, the alarm code is issued.
 - The recovery switch DI function is disabled. If this function is enabled, the recovery switch DI signal is on.

**CAUTION**

- 1 Perform neither backward execution nor the single step operation while the FFR sequence is executed.
- 2 The update cycle period for the FFR sequence information DO is 300 ms.
When the conditions listed above have been changed, wait 300 ms before program execution.
- 3 When the teach pendant is enabled, the FFR sequence is not executed.

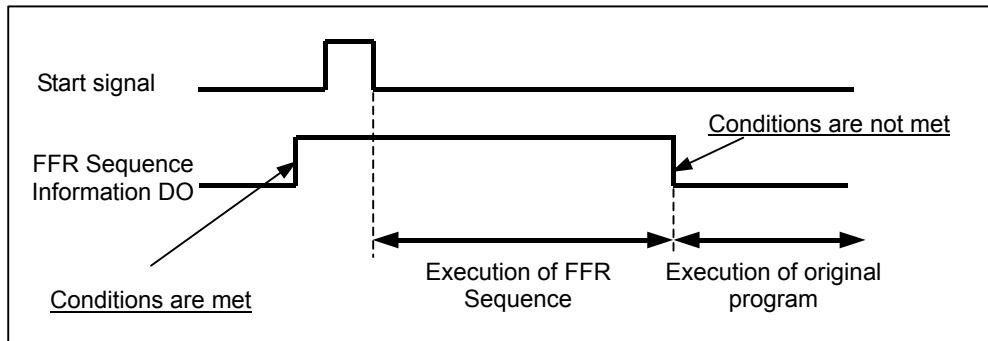


Fig. 9.30.8 (e) FFR Sequence Information DO Output Timing Chart

For multi-task

This item can be set by each instruction number.

9.30.9 Manual Operation Screen of the Resume Program Function

A manual operation screen is supported for the resume program function (The fast exit/entry feature is not supported.). To display the manual operation screen, first press MENU to display the screen menu, then select "3 MANUAL FCTNS". Then, press F1, [TYPE] to display the screen switching menu, then select Err recovery.

[MENU] → 3 MANUAL FCTNS → Err recovery

Error Recovery MNFC 1/1

Instruction number: 1

Error recovery DO status: OFF

Original program:]

[WELD

Defined resume program:]

[WIRE_CUT

RESUME comment: [Wire cutting]

1 Operation mode: **AUTO**

[TYPE] DETAIL NUMBER [CHOICE]

↓

Error Recovery MNFC 1/11

Instruction number: 1

1 Auto error recovery enabled: Yes

2 PAUSED & resume prog incomp: No

3 Program has motion group: No

4 Not in single step mode: No

5 Resume program is defined: No

6 Mode is(AUTO): Yes

7 Approval DI is ON: None

8 Defined alarm occurs: No

9 Remote when \$RMT_MASTER is 0: None

10 No disabled options: None

11 User condition param enable: Yes

[TYPE] DONE

Selecting the instruction number

The instruction number of the resume program instruction is indicated. It is possible to switch to a manual operation screen of other instruction numbers by changing this item.

Resume program information DO status

The resume program information DO status is indicated. Even when the resume program information DO is not defined, its status can be indicated. From this information, the operator can know which program, the resume program or original program, is to be executed.

Original program

The name of the original program that defines the resume program is indicated.

Defined resume program

The name of the resume program defined in the currently selected program is indicated. From this information, the operator can check whether a wrong resume program is defined or not.

**CAUTION**

If a wrong program is defined as the resume program, the robot operation is unpredictable. Therefore, check that the resume program is correct.

Resume program comment

The comment specified for the resume instruction is indicated.

Operation mode

There are three operation modes. The standard setting is AUTO. When the display changes from this screen to another, AUTO is automatically set again.

Operation mode	Explanation
AUTO	<u>This mode should be set when the teach pendant is disabled.</u> When this mode is selected, the resume program is executed according to the status of the alarm code monitoring function and recovery switch DI function. If this mode is selected when the teach pendant is enabled, the resume program is not executed.
NO_EXEC	When this mode is selected, the resume program information DO is always off. Therefore, in this mode, the resume program is not executed.
TP_TEST	<u>This mode should be set when the teach pendant is enabled.</u> When this mode is selected, and when the teach pendant is enabled, the resume program is always executed regardless of the status of the alarm code monitoring function or error recovery switch DI function.

Displaying detail conditions of the resume program information DO

When F2 (DETAIL) is pressed on the manual operation screen of the automatic error recovery function, detail conditions related to the resume program information DO status are displayed. When all items on the detail screen are set to Yes or None, the resume program information DO is turned on. When the resume program information DO is off, and you cannot find the cause of the DO being off, check this screen.

Auto error recovery enabled

This item indicates whether this function is enabled or disabled on the setting screen of the automatic error recovery function.

PAUSED & resume prog incomp

This item indicates the following conditions:

- The original program must be in the suspended state.

- A resume program must be defined in the original program, and the execution of the resume program must not have been completed.

Program has motion group

This item indicates that the selected program has a motion group.

Not in single step mode

This item indicates that the single step mode is not set.

The single step LED on the teach pendant indicates the single step status of the original program. Even when the single step key is pressed while the resume program is suspended, and the single step LED goes on, the resume program information DO is held on. This is because the LED indicates that the original program is in single step mode; the resume program is not in single step mode.

Resume program is defined

This item indicates that a resume program is defined in the original program.

Mode is (xxxx)

This item indicates that the operation mode is suitable for the current status.

For example, when the teach pendant is disabled, "AUTO" is indicated in the portion "xxxx." When the teach pendant is enabled, "TP_TEST" is indicated.

Approval DI is ON

This item indicates the recovery switch DI status. When the DI number is not defined, or when the teach pendant is enabled, "None" is indicated.

Defined alarm occurs

This item indicates that an alarm code is defined, and that alarm is issued.

When no alarm code is defined, or when the teach pendant is enabled, "None" is indicated.

Remote when \$RMT_MASTER is 0

This item indicates that remote conditions are met. This function is enabled only when the teach pendant is disabled, system variable \$RMT_MASTER is 0, and system variable \$RSMPRG_SV.\$CHK_REMOTE is true.

No disabled options

There are options that cannot be used together with the automatic error recovery function. This item indicates whether such options are present or not.

User condition param enable

This item indicates the status of the user condition parameter. For how to use the user condition parameter, see "Changing conditions for executing the resume program."

9.30.10 Execution of the Resume Program from the Teach Pendant and Test Mode

Normally, the automatic error recovery function is used when production is started with the teach pendant disabled. When checking the resume program during teaching, set the operation mode to TP_TEST on the manual operation screen. In TP_TEST mode, the resume program can be executed regardless of the recovery switch DI status and whether a defined alarm is issued or not. A manual operation screen is not supported for the fast exit/entry feature.

9.30.11 Changing Conditions for Executing the Resume Program

To use resume program execution conditions other than alarm codes, use user condition parameter and the status monitoring function.

RESUME_PROG instruction	User condition parameter
RESUME_PROG[1:Comment]	\$AUTORCV_ENB
RESUME_PROG[2:Comment]	\$AUTORCVENB2
RESUME_PROG[3:Comment]	\$AUTORCVENB3
RESUME_PROG[4:Comment]	\$AUTORCVENB4
RESUME_PROG[5:Comment]	\$AUTORCVENB5

For example, to execute the resume program when R[1] is 1, create the following monitor program, and start MONIT1.CH on the system monitor screen.

```
MONIT1.CH
1: WHEN R[1]=1,CALL DO_RESUME
2: WHEN R[1]<>1,CALL NO_RESUME
```

```
DO_RESUME.TP
1: $AUTORCV_ENB=1
2: MONITOR MONIT_3
```

```
MONIT2.CH
1: WHEN R[1]=1,CALL DO_RESUME
```

```
DO_RESUME.TP
1: $AUTORCV_ENB=0
2: MONITOR MONIT_2
```

```
MONIT3.CH
1: WHEN R[1]<>1,CALL NO_RESUME
```

The start conditions can be changed by modifying the monitor program. For how to use the status monitoring function, refer to the operator's manual on the status monitoring function. In this case, the automatic start function is unavailable.

9.30.12 Other Specifications and Restrictions

Automatic error recovery function

- The automatic error recovery function supports the power failure handling function.
- The automatic error recovery function is disabled when one of the following options is loaded:
 - Arc sensor
 - AVC (TIG arc length control)
 - Root path memorization
 - Line tracking
 - Soft float
 - Continuous turn
 - Remote TCP

Resume program function

- While the resume program is being executed, single step operation is not performed. Single step mode is valid only for the original program.
- When the cursor line is changed and executed while the original program is suspended, the resume program is not executed.
- While the resume program is being executed, the resume program execution status cannot be checked on the program edit screen. On the edit screen, the suspended original program is displayed.
- The resume program instructions cannot be defined in the program executed by the RUN instruction.
- When the program execution is done after the cursor line in the original program is moved while the resume program is suspended, the popup menu to confirm the movement of the cursor is displayed. When Yes is selected for the question on the popup menu, the original program is executed starting from the new cursor line. When No is selected, the resume program is restarted.

- Programs other than the original program cannot be selected by PNSTROBE signal while the resume program is executed.
- When PNSTROBE signal has been on when the resume program starts, the alarm is issued. Turn off PNSTROBE signal when the resume program starts.
- When programs other than the original program are selected while executing a original program, the original program is automatically selected again at the same time as the resume program's starting/ending.
- Never teach the arc and weaving instructions in the resume program. If an arc instruction is executed in the resume program while arc welding is being performed by the original program, alarm is issued. In addition, weaving operation is not performed within the resume program.

Fast exit/entry feature

- When an original program is a single step mode, the FFR sequence is not started.
- When the cursor line is changed and executed while the original program is suspended, the FFR sequence is not executed.
- The maintenance program instructions cannot be defined in the program executed by the RUN instruction.
- Programs other than the maintenance program cannot be selected by PNSTROBE signal while the FFR sequence is executed.
- When PNSTROBE signal has been on when the FFR sequence starts/ends, the alarm is issued. Turn off PNSTROBE signal when the FFR sequence starts/ends.
- When programs other than the original program are selected while executing a original program, the original program is automatically selected again at the same time as the FFR sequence 's starting.

9.30.13 Warnings (Be sure to read this section for safety.)

When using the automatic error recovery function, observe the following safety precautions:

- If a wrong program or a program causing wrong operation is defined as a resume program or a maintenance program, the robot moves in a direction the operator cannot predict. Define a correct program.
- Before inputting the start signal and before pressing the execution key on the teach pendant, for safety, check the resume program information DO status to confirm whether the original program or resume program is to be started.
- Before inputting the start signal and before pressing the execution key on the teach pendant, for safety, check the FFR sequence information DO status to confirm whether the original program or maintenance program is to be started.
- If the operation mode is set to TP-TEST on the manual operation screen of the automatic error recovery function, the resume program is started even when a defined alarm is not issued or when the recovery switch DI is off.
- When an operation mode other than AUTO is set on the manual operation screen of the automatic error recovery function, then the display is changed to another screen, the operation mode is set to AUTO again automatically. To use an operation mode other than AUTO, always keep displaying the manual operation screen of the automatic error recovery function.
- When the resume program or the maintenance program terminates in the middle, and the start signal is input, the robot might do action not predictable. Therefore, before inputting the start signal, check the program name and the program start line that will be executed with the next start signal. In addition, if an interfering object exists, jog the robot to a position where interfering object doesn't exist, then input the start signal.
- Perform neither backward execution nor the single step operation while the FFR sequence is executed.

In this case, when DI[1] is ON, DO[1] is turned on, when DI[1] is OFF, DO[1] is not changed.
The assignment statement can use the following items and operators.

Items for left side	Operator for right side or IF condition	Items for right side or IF condition
DO[]	()	ON OFF
RO[]	+	Constant value
WO[]	-	DI[] DO[]
AO[]	*	RI[] RO[]
GO[]	/	WI[] WO[]
SO[]	MOD	AI[] AO[]
UO[]	DIV	GI[] GO[]
F[]	AND	SI[] SO[]
R[]	OR	UI[] UO[]
PR[i,j]	!	F[]
System variables	= (comparison)	M[]
M[]	<	R[]
TC_ONLINE	>	PR[i,j]
	<=	TIMER[]
	>=	System variables
	<>	TCP_SPD[]

Marker assignment and TC_ONLINE can be specified in PS statement.

AR[] (calling parameter) can not be specified in PS statement.

Up to 20 operations can be specified in one motion line as total of right side formula and IF condition.
One motion statement can have up to 20 PS statements.

Case that the trigger point is negative and it is farther than the start position (distance-before type)

If the trigger point is negative and it is farther than the start position, this is triggered immediately.

```
1: L P[1] 200mm/sec FINE
2: L P[2] 200mm/sec FINE DB 100mm, CALL A
```

Contents of sub-program A

```
1: PS -1000mm +0sec, DO[1]=(ON)
```

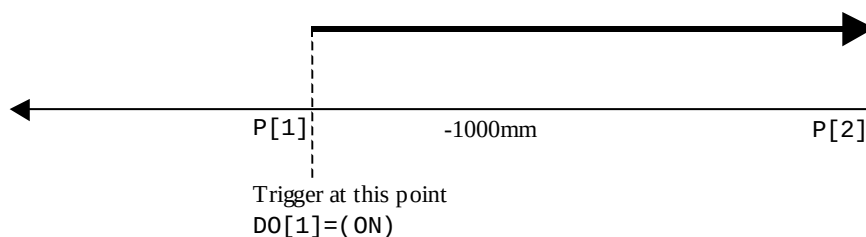


Fig. 9.31 (b) Path switch application

You can choose to cause error at this situation by following parameter setting.

\$PS_CONFIG.\$DB_IMMTRIG (distance-before type)

\$PS_CONFIG.\$DA_IMMTRIG (distance-after type)

Value	Description
0	Trigger PS condition immediately, but post neither alarm nor warning.
1 (default)	Trigger PS condition immediately, and post "INTP-539 PS(program name, line number, distance[mm]) Already in area" (warning).
other than (0, 1)	Not triggered, and post "INTP-540 PS(program name, line number, distance[mm]) Already in area" (PAUSE).

Case that the trigger point is passed by jog while the program is paused

When original path resume is disabled (\$SCR.\$ORG_PTH_RSM=FALSE), if the program is paused during the motion with PS, and jogged by TP, and the trigger point is passed by jog, then the PS condition is triggered immediately when the program is resumed.

When original path resume is enabled (\$SCR.\$ORG_PTH_RSM=TRUE), once back to the paused position on the original motion path and re-start the program. In this case, the PS condition is triggered when the trigger point is passed after resume and re-start.

You can choose to cause error at this situation by above parameter setting.
(\$PS_CONFIG, \$DB_IMMTRIG, \$DA_IMMTRIG)

```
1: L P[1] 200mm/sec FINE
2: L P[2] 200mm/sec FINE DB 100mm, CALL A
```

Contents of sub-program A

```
1: PS -500mm +0sec, DO[1]=(ON)
```

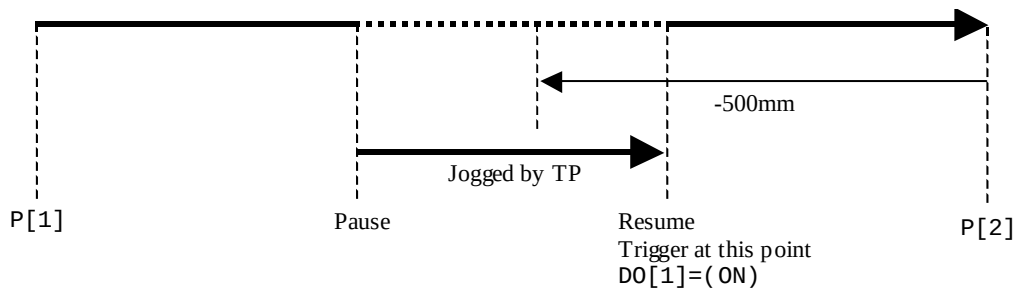


Fig. 9.31 (c) Path switch application (original path resume is disabled)

Case that TCP path moves away without entering the trigger area on CNT motion

In case of CNT motion, teaching point does not on the actual robot path. If the robot path is too far from the trigger area, reminding untriggered distance-before type PS conditions are triggered when the TCP path starts to move away from the teaching point.

```
1: L P[1] 200mm/sec FINE
2: L P[2] 200mm/sec CNT100 DB 100mm, CALL A
3: L P[3] 200mm/sec FINE
```

Contents of sub-program A

```
1: PS -10mm +0sec, DO[1]=(ON) (distance-before type)
2: PS +10mm +0sec, DO[2]=(ON) (distance-after type)
```

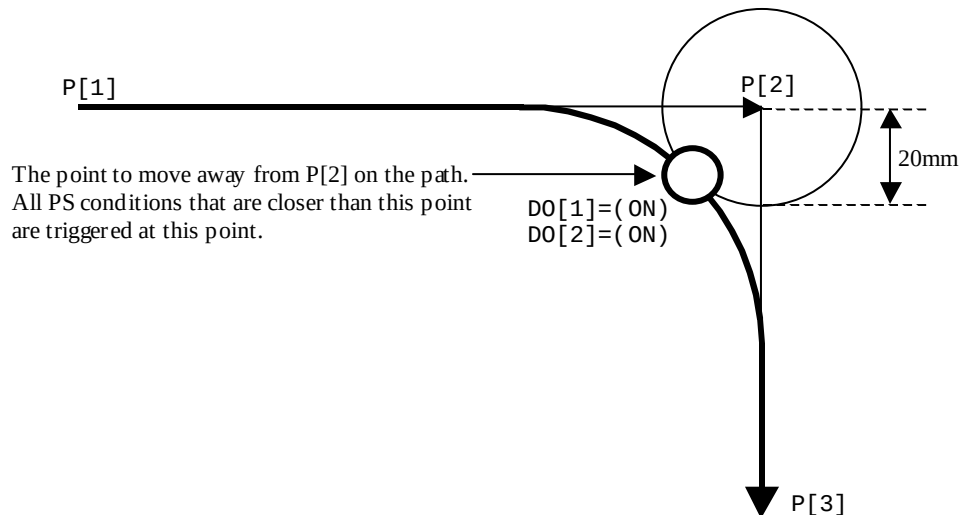


Fig. 9.31 (d) Path switch application

You can choose to cause error at this situation by following parameter setting.

\$PS_CONFIG.\$DB_NOTRIG (distance-before type)
\$PS_CONFIG.\$DA_NOTRIG (distance-after type)

Value	Description
0	Trigger PS condition at the closest point, but post neither alarm nor warning.
1 (default)	Trigger PS condition at the closest point, and post "INTP-541 PS(program name, line number, distance[mm]) Forced trigger" (warning).
other than (0, 1)	Not triggered, and post "INTP-542 PS(program name, line number, distance[mm]) No trigger" (PAUSE).

Case that the trigger point is positive and it is farther than the destination position (distance-after type)

If the trigger point is positive and it is farther than the destination position of the next motion, the PS condition is triggered when the next motion is completed.

You can choose to cause error at this situation by above parameter setting.
(\$PS_CONFIG. \$DB_NOTRIG, \$DA_NOTRIG)

```
1: L P[1] 200mm/sec FINE
2: L P[2] 200mm/sec FINE DB 100mm, CALL A
3: L P[3] 200mm/sec FINE
```

Contents of sub-program A

```
1: PS +1500mm +0sec, DO[1]=(ON)
```

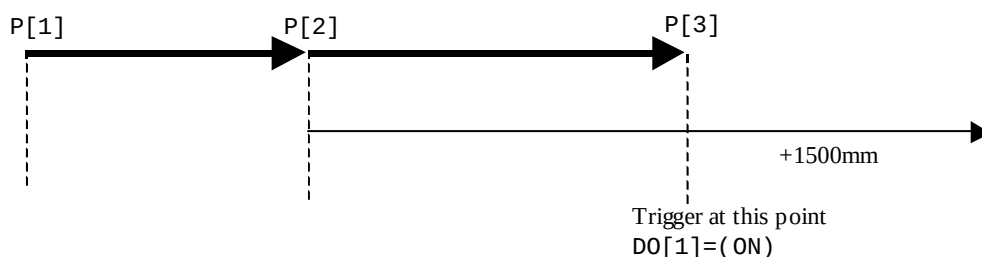


Fig. 9.31 (e) Path switch application

About multi group configuration

In case of multi group configuration, the PS statement is implicitly executed for the first motion group in the group mask of the TP program. For example, if the motion mask is [*, 1, 1, *, *, *, *, *], then the PS statement is executed for group 2.

Limitations

PS statement is not available for the following motion type.

- Weaving motion
- Tracking motion which can not define actual destination point beforehand, as line-tracking, MIG-EYE, Robot-Link.

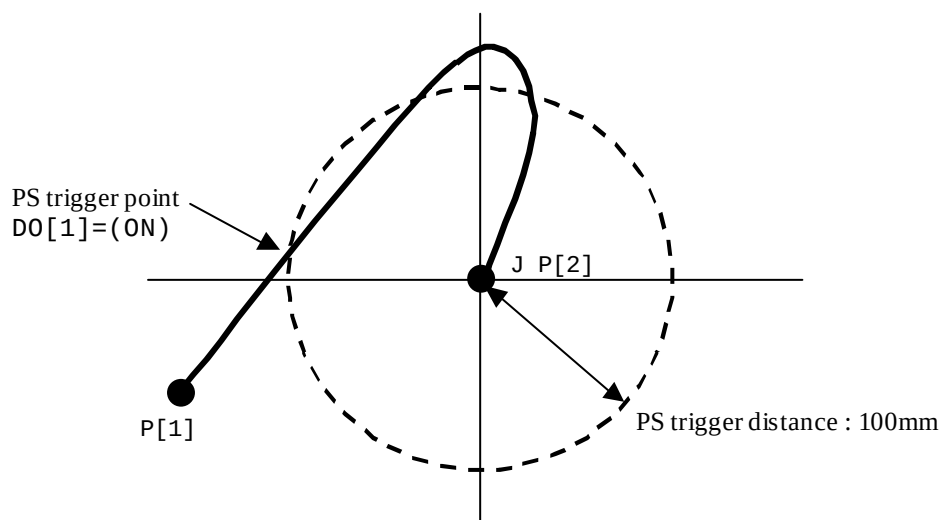
Caution

- PS statement is interpreted in a flash only at the start of the new motion line. When the motion including PS statement is paused, if you edit and change the PS statement in the sub program during this pause status, modified PS commands do not take effect until the motion line is newly executed. After pausing program, to take effect of the modified PS statement immediately, you can BWD the motion line and re-start. Then the modified PS statements will take effect.
- PS command for joint motion is not prohibited. But it must be used carefully. Regardless of the motion type Linear, Circular, or Joint, PS command is checking the linear distance between TCP and destination point as trigger threshold.



NOTE

In cases of PS command's Distance-Before type usage, If TCP once goes into the defined sphere around the destination point, the PS command will be triggered regardless how long the rest of path. Please note the distance which is being checked in PS motion is not the total remaining path length to the destination point. Please see the following figure for example.



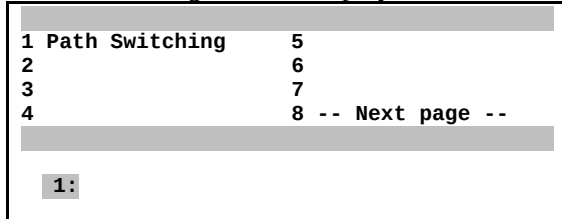
```
1: J P[1] 100% FINE
2: J P[2] 100% FINE DB 100mm, CALL PROG1
```

Contents of sub-program PROG1

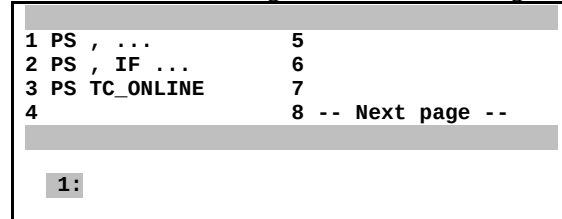
```
1: PS -100mm +0.0sec, DO[1] = ON
```

TP editor operation

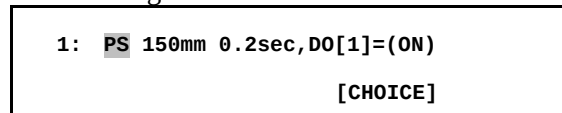
1. "Path Switching" item is displayed in the [INST] menu.



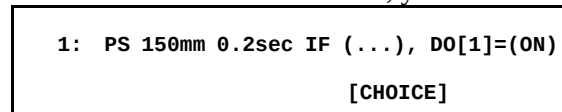
2. Select "Path Switching" item, the following menu is displayed.



3. The distance and offset time can be set, and the assignment statement can be edited as the same way as mixed logic instructions.



4. When the cursor is on 'PS ' item, you can switch "PS ..." to "PS ... IF ..." and vice versa from menu.



9.32 AUXILIARY AXIS SERVO OFF (LOCAL STOP) FUNCTION

9.32.1 Introduction

In the robot system that auxiliary axis controls a jig, there is a requirement to cut off the power of auxiliary axis for safety when an operator do a job with that jig. Auxiliary axis servo off(Local Stop) is a function for that purpose. This function can turn off motor power supply for each auxiliary axis respectively and prevent axis from unexpected movements.

This function has following merits.

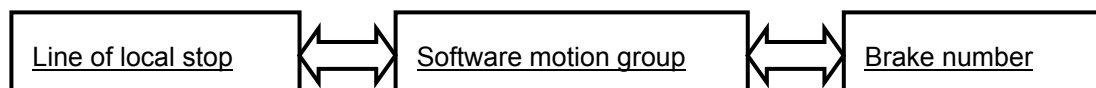
- Keep safety for operator who works at jig (auxiliary axis).
- Improvement of cycle time in comparison with using machine lock. In case of machine lock, cycle time becomes long because power supply to amplifier is controlled(OFF/ON).
- Output servo ON/OFF signal by just using macro program or press button.

This function is an optional. And dedicated hardware (Local Stop Unit) is required.

Please refer to the CONNECTION/MAINTENANCE MANUAL A-83118E(for R-J3iB Controller) OR CONNECTION/MAINTENANCE MANUAL A-95011E(for R-30iA Controller) for the detail of Local Stop Unit.

9.32.2 Specification

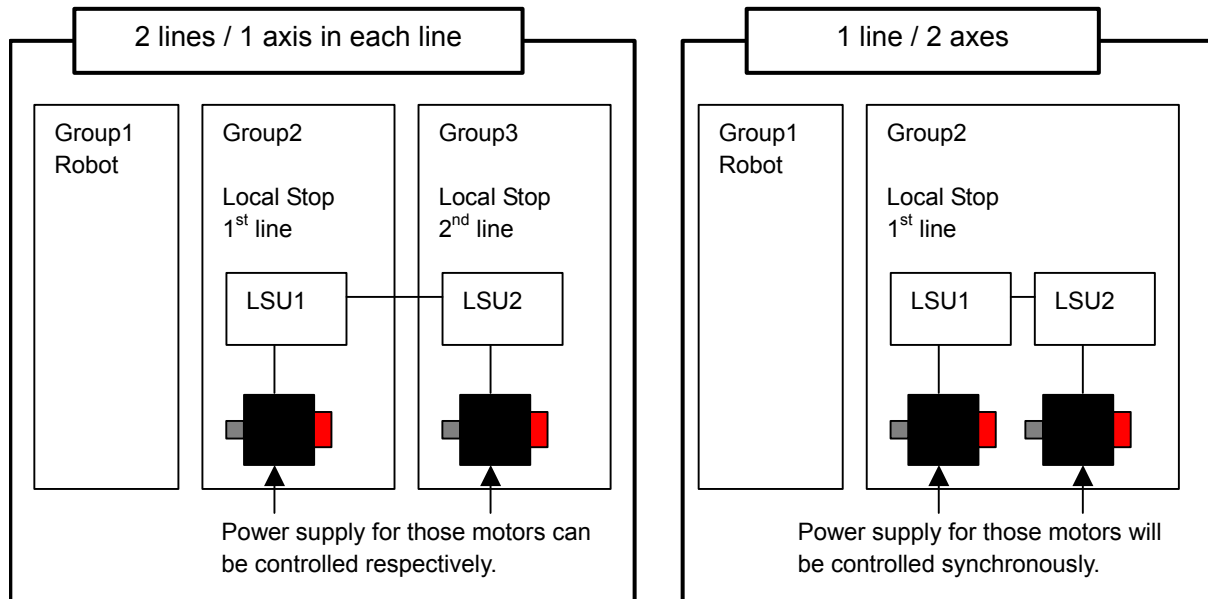
- Order number for software option
R-30iA controller: A05B-2500-J806
- Way to cut off motor power supply
This function cuts off power supply at power cable between servo amplifier and motor. Even in the local stop mode(power supply for motor is cut off), power is supplied to the amplifier.
- Motor for the auxiliary axis
Please be sure to use motor with brake for this function and enable brake control when you set up auxiliary axis.
- Local Stop Unit
In R-J3iB, brake unit and Local Stop Unit are separated.
In R-30iA, brake unit and Local Stop Unit are united.
- Line of local stop
The unit(group) in which power supply of motors is controlled synchronously is called as Line of local stop. Line concept corresponds to the software motion group. Motors in the same line must have the same brake number and motors in the different line must have the different brake number. Software supports 3 local stop lines at maximum.



- Number of axes in the local stop line
If there are multiple axes in the one local stop line, power supply for those motors will be controlled synchronously. In this case, there are multiple axes in one motion group. Power supply for those motors cannot be controlled separately. If you want to, please separate a local stop line.

Configuration of local stop function is determined by the number of lines and the number of axes in one line.

- Example of system configuration
When there are two axes that use local stop function, possible configurations are as follows.



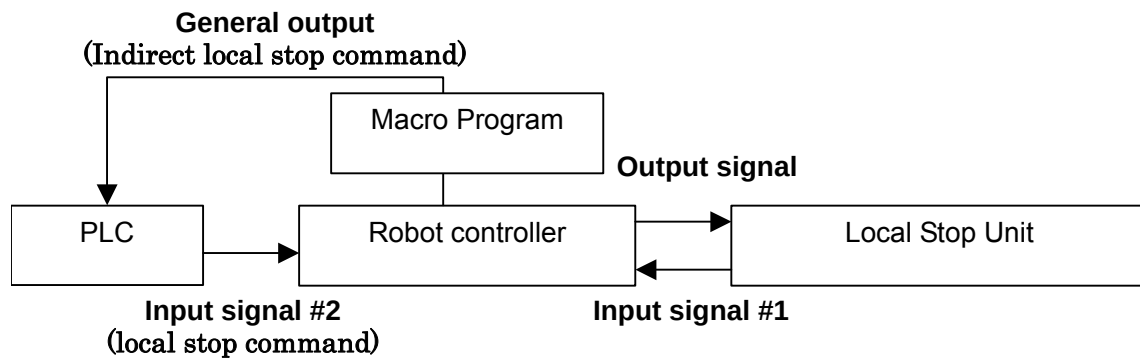
- Necessary signals for local stop function
Two input signals and one output signal are required for each local stop line.
 - Input signal #1
Signal from a Local Stop Unit to a robot controller.
This signal is used by the software internally to check that Local Stop Unit functions normally.
Please do not operate this signal directly.
 - Input signal #2
Local stop command signal by the user.
When this signal is ON, power is supplied to the motor and axis can move.
When this signal is OFF, power is not supplied to the motor and axis cannot move.
 - Output signal
Signal from a robot controller to a Local Stop Unit.
This signal will be output by the software automatically according to the input signal #2.
Please do not operate this signal directly.

For R-J3iB controller, please assign general I/O or robot I/O to all three signals.

For R-30iA controller, please assign general I/O or robot I/O only for input signal #2 (Rest of signals are assigned to the specialized I/O of a robot controller).

- Basic flow of process
When user input the input signal #2, software output signal to the Local Stop Unit and power supply for motor will be controlled. Software checks the status of Local Stop Unit by the input signal #1. Please connect a peripheral device such as push button to the input signal #2.

In case a macro program will input that local stop command, please output general output signal in that program to the peripheral device such as PLC and control input signal #2 according to that output.



9.32.3 Constraints

- Alarm occurs when user command cutting off the power supply while auxiliary axis is moving. Running program will be paused and robot stops when this alarm occurs. Please command power off while axis is not moving.
- Alarm occurs when user tried to move axis while the power supply is cut off. Axis does not move in this case. Please turn on power supply for auxiliary axis before you move axis.

9.32.4 Settings

- Firstly, please connect Local Stop Unit (Please refer to the CONNECTION /MAINTENANCE MANUAL A-83118E(for R-J3iB Controller) OR CONNECTION /MAINTENANCE MANUAL A-95011E(for R-30iA Controller) for the detail of Local Stop Unit.).
- Secondly, please set system variables. For each local stop line, motion group number and setting of input signal #1, input signal #2, and output signal are necessary. Please specify port type and port number for those signals.

Procedure to set up:

- Display the SYSTEM Variables screen.
 - Press the **MENUS** key.
 - Select 0 NEXT and then 6 SYSTEM.
 - Press **F1** (TYPE) key.
 - Select Variables, and then SYSTEM variables screen will be shown.
- Move the cursor to \$LS_IOPORT and press the **ENTER** key.
You will see a screen similar to the following.

SYSTEM Variables		JOINT 10 %
\$LS_IOPORT		1/3
1	[1]	LS_IOPORT_T
2	[2]	LS_IOPORT_T
3	[3]	LS_IOPORT_T

- Select the local stop line(1~3) that needs to be configured and press the **ENTER** key.
You will see a screen similar to the following.

SYSTEM Variables		JOINT 10 %
\$LS_IOPORT[1]		1/10
1 \$MO_GRP_NUM	0	
2 \$SDI1_P_TYPE	11	
3 \$SDI1_P_NUM	12	
4 \$SDI1_P_STAT	FALSE	
5 \$SDI2_P_TYPE	1	
6 \$SDI2_P_NUM	1	
7 \$SDI2_P_STAT	TRUE	
8 \$SDO_P_TYPE	12	
9 \$SDO_P_NUM	14	
10 \$SDO_P_STAT	TRUE	
[TYPE]		

4. Please refer to the below tables and set up the system variables according to the actual I/O connection.

Description of system variable (Following settings are necessary for each local stop line)

Variable name	Description
\$MO_GRP_NUM	Motion group number that uses a local stop function. Valid range is 2-5. Setting this variable to 0 disables local stop function.
\$SDI1_P_TYPE	Port type for input signal #1(signal to check the status of a Local Stop Unit).
\$SDI1_P_NUM	Port number for input signal #1.
\$SDI1_P_STAT	Status of input signal #1. Please do not modify this variable.
\$SDI2_P_TYPE	Port type for input signal #2(local stop command signal by a user).
\$SDI2_P_NUM	Port number for input signal #2.
\$SDI2_P_STAT	Status of input signal #2. Please do not modify this variable.
\$SDO_P_TYPE	Port type for output signal(local stop command signal from a robot controller to a Local Stop Unit).
\$SDO_P_NUM	Port number for output signal.
\$SDO_P_STAT	Status of output signal. Please do not modify this variable.

Values for port type (\$****_P_TYPE) are as follows.

For R-30iA controller, input signal #1 and output signal for local stop function are assigned to the operator panel I/O. Please assign general I/O or robot I/O only for the input signal #2.

Value for port type	Type of signal
1	DI (General input signal)
2	DO (General output signal)
8	RI (Robot input signal)
9	RO (Robot output signal)
11	SI (Operator panel input signal)
12	SO (Operator panel output signal)

Initial values for system variables

System variables are set as follows initially. When you use a local stop function, please change these variables properly according to the actual signal connection.

R-J3iB Controller (It is necessary to set three signals for each line)

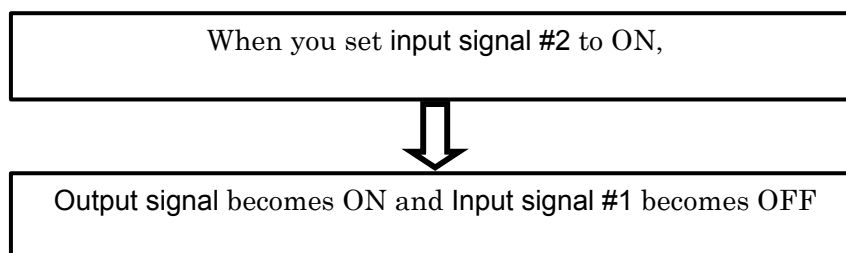
LINE	Role of signal in local stop function	System variable		Signal to be used by this setting
		name	value	
1	Input signal #1	\$LS_IOPORT[1].\$SDI1_P_TYPE	8	RI[7]
		\$LS_IOPORT[1].\$SDI1_P_NUM	7	
	Input signal #2	\$LS_IOPORT[1].\$SDI2_P_TYPE	8	RI [8]
		\$LS_IOPORT[1].\$SDI2_P_NUM	8	
	Output signal	\$LS_IOPORT[1].\$SDO_P_TYPE	9	RO[8]
		\$LS_IOPORT[1].\$SDO_P_NUM	8	
2	Input signal #1	\$LS_IOPORT[2].\$SDI1_P_TYPE	8	RI[5]
		\$LS_IOPORT[2].\$SDI1_P_NUM	5	
	Input signal #2	\$LS_IOPORT[2].\$SDI2_P_TYPE	8	RI[6]
		\$LS_IOPORT[2].\$SDI2_P_NUM	6	
	Output signal	\$LS_IOPORT[2].\$SDO_P_TYPE	9	RO[7]
		\$LS_IOPORT[2].\$SDO_P_NUM	7	
3	Input signal #1	\$LS_IOPORT[3].\$SDI1_P_TYPE	8	RI[3]
		\$LS_IOPORT[3].\$SDI1_P_NUM	3	
	Input signal #2	\$LS_IOPORT[3].\$SDI2_P_TYPE	8	RI[4]
		\$LS_IOPORT[3].\$SDI2_P_NUM	4	
	Output signal	\$LS_IOPORT[3].\$SDO_P_TYPE	9	RO[6]
		\$LS_IOPORT[3].\$SDO_P_NUM	6	

R-30iA Controller (* It is unnecessary to change SI and SO for LINE1,2)

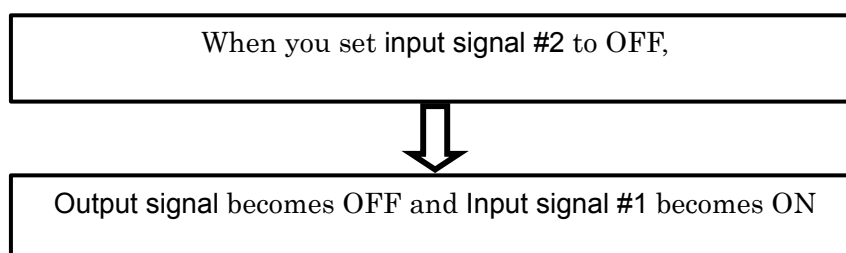
LINE	Role of signal in local stop function	System variable		Signal to be used by this setting
		name	value	
1	Input signal #1 (*)	\$LS_IOPORT[1].\$SDI1_P_TYPE	11	SI[12]
		\$LS_IOPORT[1].\$SDI1_P_NUM	12	
	Input signal #2	\$LS_IOPORT[1].\$SDI2_P_TYPE	1	DI[1]
		\$LS_IOPORT[1].\$SDI2_P_NUM	1	
	Output signal (*)	\$LS_IOPORT[1].\$SDO_P_TYPE	12	SO[14]
		\$LS_IOPORT[1].\$SDO_P_NUM	14	
2	Input signal #1 (*)	\$LS_IOPORT[2].\$SDI1_P_TYPE	11	SI[13]
		\$LS_IOPORT[2].\$SDI1_P_NUM	13	
	Input signal #2	\$LS_IOPORT[2].\$SDI2_P_TYPE	1	DI[2]
		\$LS_IOPORT[2].\$SDI2_P_NUM	2	
	Output signal (*)	\$LS_IOPORT[2].\$SDO_P_TYPE	12	SO[15]
		\$LS_IOPORT[2].\$SDO_P_NUM	15	
3	Input signal #1	\$LS_IOPORT[3].\$SDI1_P_TYPE	1	DI[3]
		\$LS_IOPORT[3].\$SDI1_P_NUM	3	
	Input signal #2	\$LS_IOPORT[3].\$SDI2_P_TYPE	1	DI[4]
		\$LS_IOPORT[3].\$SDI2_P_NUM	4	
	Output signal	\$LS_IOPORT[3].\$SDO_P_TYPE	2	DO[1]
		\$LS_IOPORT[3].\$SDO_P_NUM	1	

5. When you use more than 2 local stop lines, please set system variables for 2nd or 3rd line too.
6. Please do not modify \$LS_CONFIG and \$LS_SYSTEM.
7. That is all for software setup. In order to make settings available, please cycle power once. After cycling power, setting will be available.
8. To confirm settings are proper, please do check by following procedure for each local stop lines.

Confirmation of exiting from the local stop mode



Confirmation of entering into the local stop mode



NOTE

Please perform these checks when the system is NOT in servo off status. If any alarm factor such as panel board e-stop, teach pendant e-stop, deadman switch release, external emergency stop, etc. exists, please clear these alarms first. Then perform a test when the system is in servo on status.

9. If no alarm occurs and signals change like above, local stop function works correctly.
10. Following alarms could occur.
 LSTP-006 DI1 ON timer expired(G:%d) or LSTP-007 DI1 OFF timer expired(G:%d)
 In case that
 LSTP-006 DI1 ON timer expired(G:%d) occurs but Input signal #1 becomes ON or
 LSTP-007 DI1 OFF timer expired(G:%d) occurs but Input signal #1 becomes OFF
 , please increase following system variables
 \$LS_CONFIG.\$SDI_ON_LAG and \$LS_CONFIG.\$SDI_OFF_LAG
 100 by 100 until alarm does not occur. Default value for these variables is 1000. Please set the same value to these variables.

SYSTEM Variables		JOINT 10 %
\$LS_CONFIG		2/6
1 \$IO_SCANRATE	12	
2 \$SDI_ON_LAG	1000	
3 \$SDI_OFF_LAG	1000	
4 \$BRK_ON_LAG	414	
5 \$BRK_OFF_LAG	544	
6 \$LS_DEBUG	0	
[TYPE]		

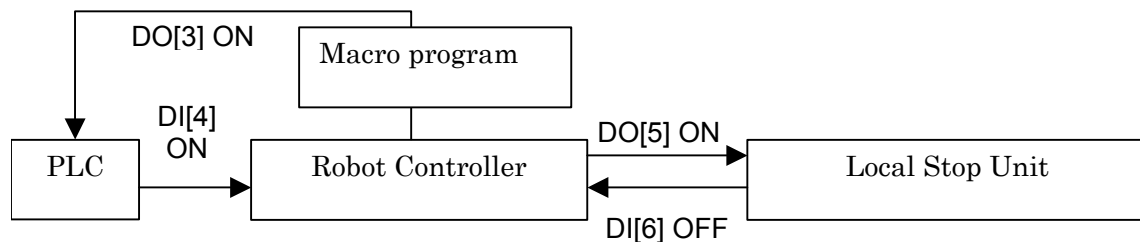
When LSTP-006 DI1 ON timer expired(G:%d) or LSTP-007 DI1 OFF timer expired(G:%d) occurs, it is necessary to cycle power once to clear alarm status.

11. If alarms occur except for the above situation, please refer to Alarm Code List Operator's Manual and check the settings of system variable, setting of brake control, and hardware connections.

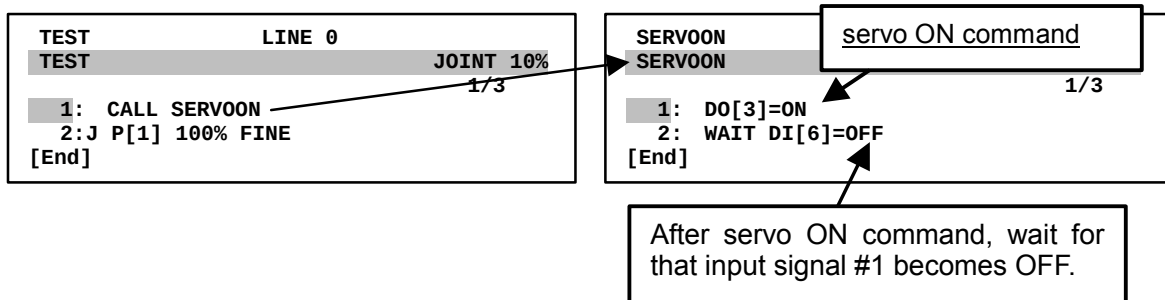
9.32.5 Attention

- Note for moving auxiliary axis right after exiting from local stop mode(servo ON command)
Please do not move auxiliary axis until input signal #1 becomes OFF.

Example) Assuming the system in which I/O has been set up like below.



In this case, please make a TP program like below.



- If you teach motion instruction right after servo ON command without WAIT instruction, *LSTP-011 Motion grp %d is in LSTOP* alarm might occur.

Motion group mask of a TP program in which servo OFF will be commanded

Please set a motion group mask to (*,*,*,*,*,*,*) for a TP program in which servo OFF will be commanded to a certain auxiliary axis group by RO or DO.

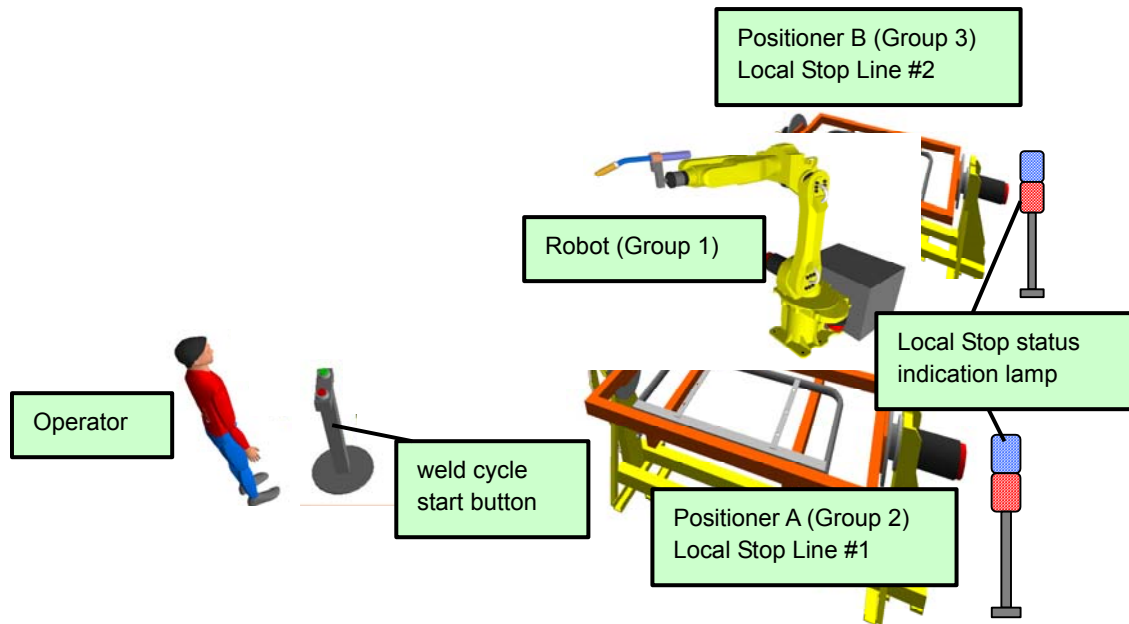
(NOTE: Please set a motion group mask at creating a program)

- When turning on a robot controller, servo ON command will be output once
This is due to the initialization of software internal status. If the input signal #2 is ON, the status will be kept (servo ON). If the input signal #2 is OFF, servo OFF command will be output and auxiliary axis will enter into the local stop mode(servo OFF).

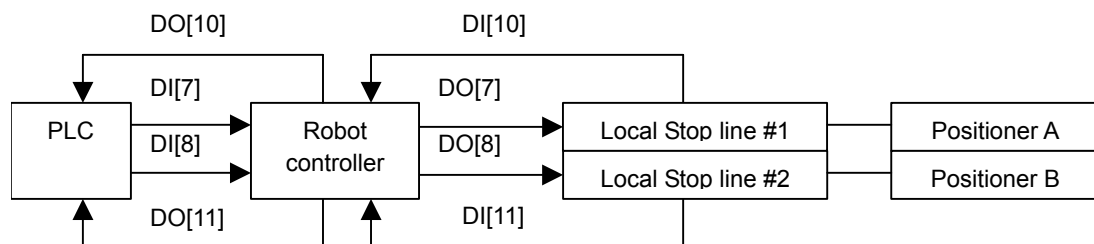
9.32.6 Programming

Suppose a system like below.

- 3 groups welding system that consists of a robot and two positioners (A and B).
- Operator loads/unloads a workpiece at one side of the positioner while robot welds at another positioner.
- Two Local Stop lines are required.

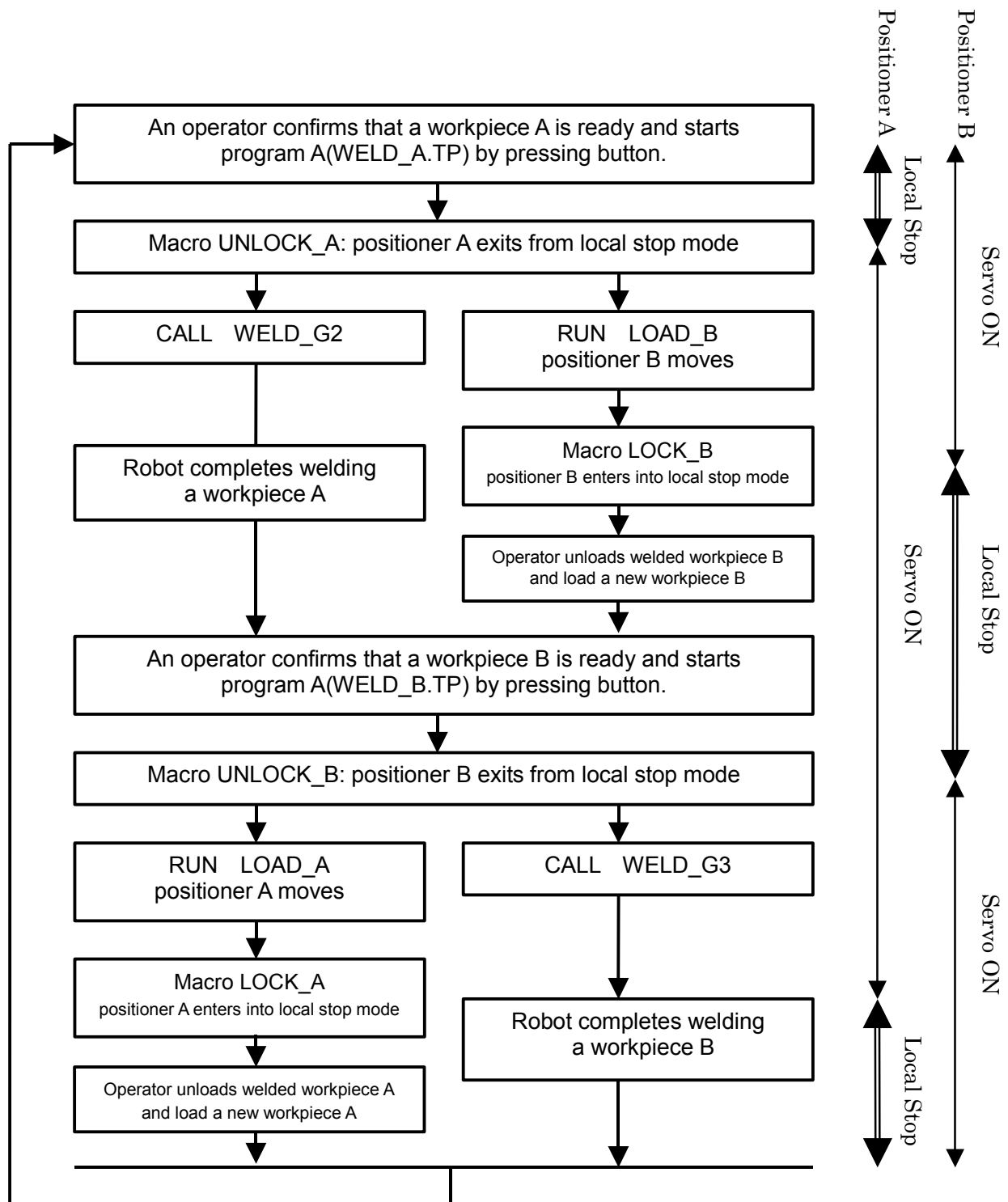


- Assume that I/O is connected like below.



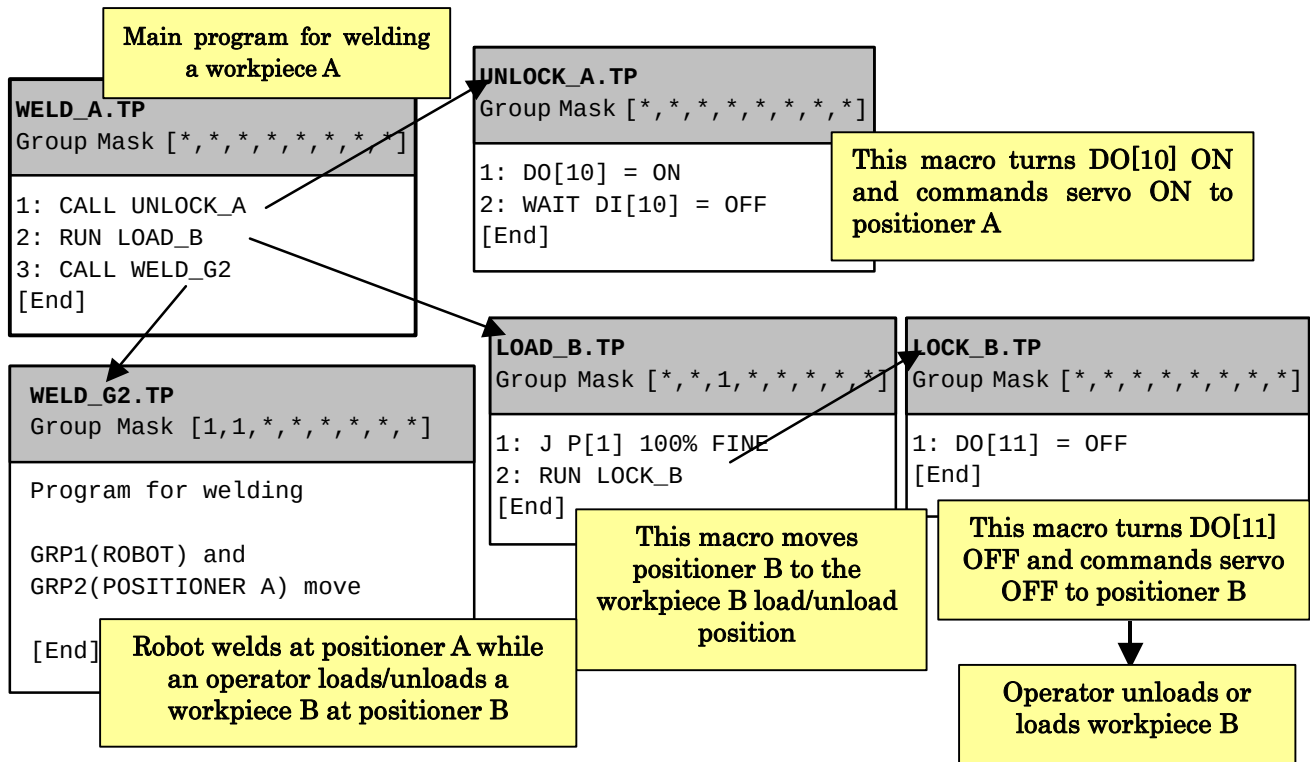
Suppose a welding sequence as follows.

- A) Robot completes welding of a workpiece B at positioner B and returns to a home position.
- B) An operator confirms that workpiece A of positioner A is ready and starts program A by pressing cycle start button.
- C) WELD_A.TP program starts.
 - C.1 In macro UNLOCK_A, positioner A exits from local stop mode.
 - C.2 In LOAD_B.TP, positioner B moves to the position for loading/unloading workpiece B.
 - C.3 In macro LOCK_B, positioner B enters into local stop mode.
 - C.4 In WELD_G2.TP, robot starts to weld a workpiece A at positioner A.
 - C.5 An operator confirms that a lamp(This lamp is connected to the input signal #1) indicates positioner B is in local stop mode. Then an operator unloads welded workpiece B and loads a new workpiece B.
- D) Robot completes welding of a workpiece A at positioner A and returns to a home position.
- E) An operator confirms that a workpiece B is ready and starts next program B by pressing button.
- F) Repeat C)~D) and then go back to program A.

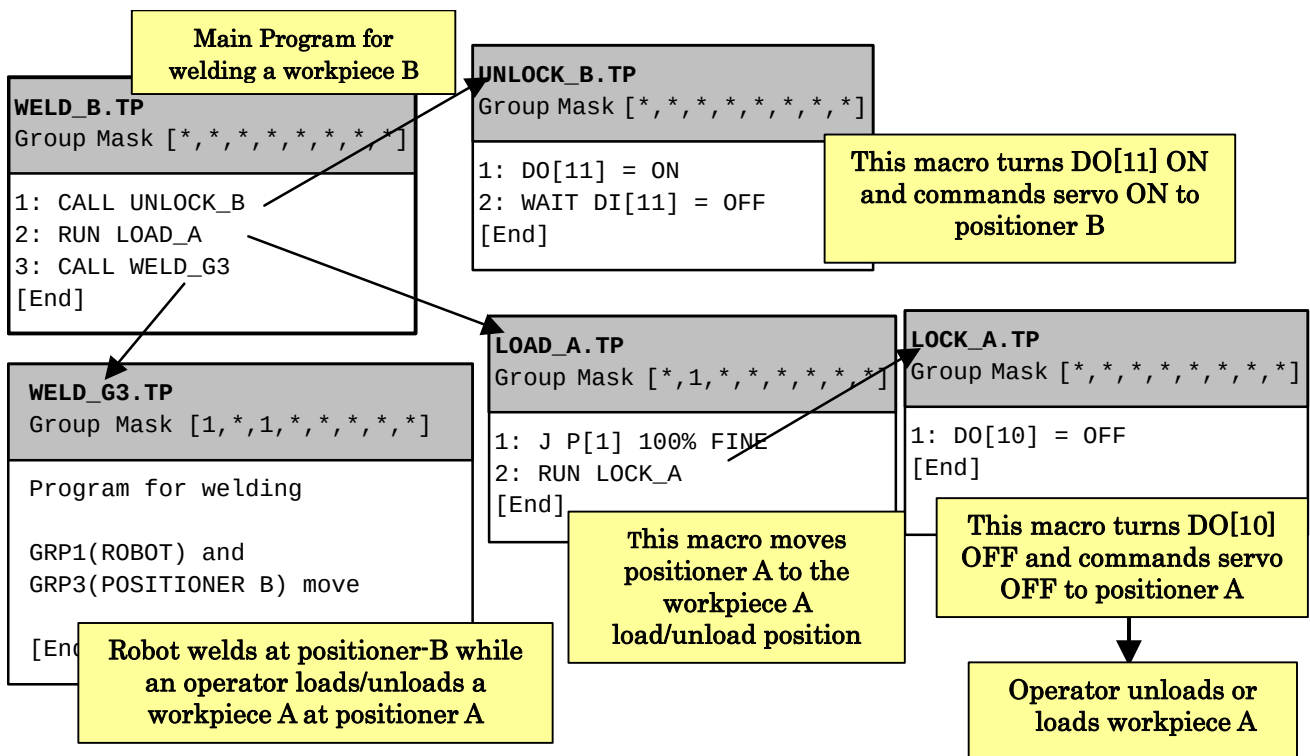
Cycle flow of the above-mentioned example.

For example, following programs accomplish this sequence.

Program A for welding of a workpiece A : WELD A



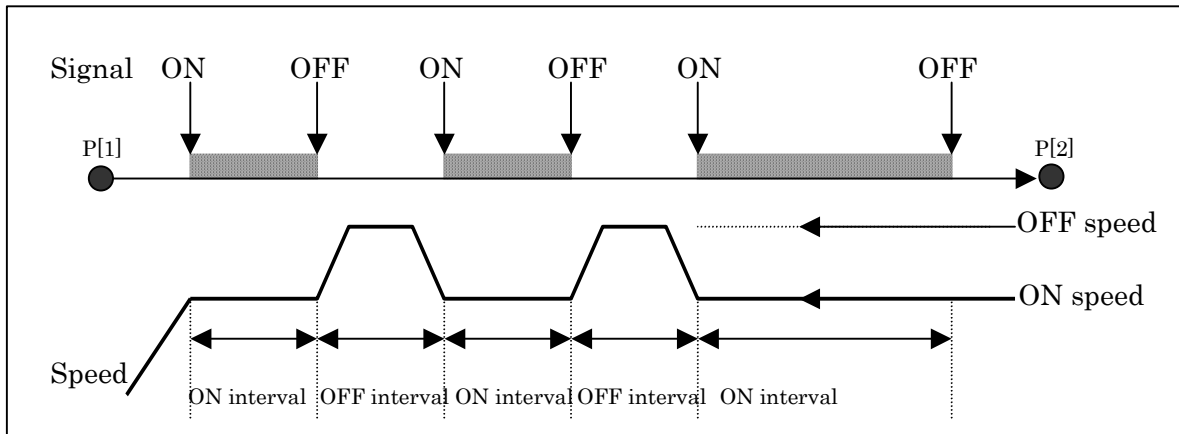
Program B for welding of a workpiece B : WELD B



9.33 STITCH FUNCTION

9.33.1 Overview

For stitch-like sealing and laser welding, stitch function enables easy teaching of such sealing/welding part (as ON interval and OFF interval) and turning signal on/off in each interval.



9.33.2 Specification

9.33.2.1 Instruction

Following two instructions specify an interval in which stitch function is used. This interval is referred as stitch interval hereafter.

- **STITCH[i]**
 - Start stitch process (change of speed, signal control) according to stitch condition.
 - Stitch process doesn't stop until "STITCH END" instruction is executed.
 - Stitch condition is specified in data screen. Please refer to 9.33.2.2.
 - This instruction is taught as usual instruction only. This cannot be taught as motion option.
- **STITCH END**
 - By this instruction, stitch process is stopped. Signal is turned off.

Example

- 1: J P[1] 100% FINE
- 2: **STITCH[1]**
- 3: L P[2] 2000mm/sec CNT100
- 4: L P[3] 2000mm/sec FINE
- 5: **STITCH END**

Stitch interval.

Only linear and circular motions are available. If joint motion type is used, error is posted.

9.33.2.2 Stitch condition

- ◆ Stitch condition is set in data screen. Screens for stitch condition are Stitch Data List screen, Stitch Data Detail screen and Stitch Data Common screen. (Please refer to 9.33.5 for structure of screens.)
- ◆ Items on each screen are as follows.

Stitch Data List screen

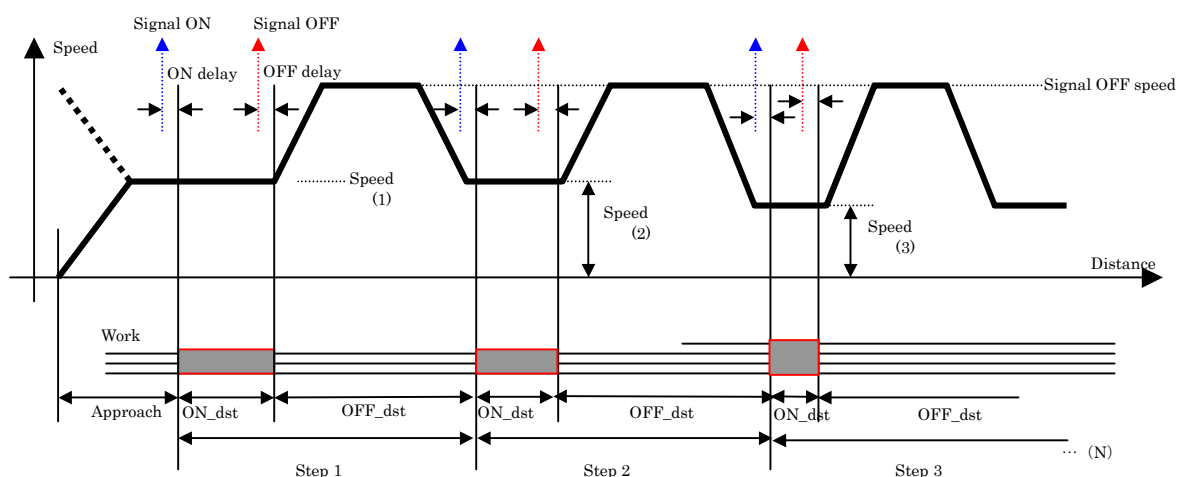
This screen displays list of stitch conditions. Stitch instruction specifies condition by number displayed in this screen. To display detail screen of each condition, set cursor to a condition and press F4.

Comment	Comment for each condition is displayed.	
F3 (COMMON)	To display Stitch Data Common screen.	
F4 (DETAIL)	To display detail screen for selected condition (condition which cursor is set to).	
F5 (COPY)	User can copy selected condition to another condition. Caution) Even if condition that selected condition is copied to is already set, the former one is overwritten.	

Stitch Data Detail screen

◆ In this screen, following items can be set.

Item	Explanation	Range
Comment	Comment for each condition is displayed	
Equipment condition	Type of equipment signal (DO/RO/GO/AO) and index	DO/RO/GO/AO 1-32766
Signal ON delay time	Delay time between signal ON and actual action of equipment can be set here.	0-9999[msec]
Signal OFF delay time	Delay time between signal OFF and actual stop of equipment can be set here.	0-9999[msec]
Approach distance	Approach distance to stabilize speed.	0-9999[mm]
Signal OFF speed.	Speed in OFF intervals. "Speed unit" in Stitch Data Common screen decides unit.	0.1 - \$PARAM_GROUP [*].\$SPEEDLIM Usually, this corresponds to 2000mm/sec
Following item decide condition for each step.		
ON_dst	Distance of signal ON interval	0-9999[mm]
Speed	Speed of signal ON interval. Unit is decided in Stitch Data Common screen.	0.1 - \$PARAM_GROUP[*].\$SP EEDLIM Usually, this corresponds to 2000mm/sec
OFF_dst	Distance of signal OFF interval.	0-9999[mm]
Count	This specifies how many times this step is executed. If you want to repeat a sequence, input time of repetition. If 0 is specified, this step is neglected. Next step with non-zero counter is executed.	0-99[times]
Cond.	Output value of equipment signal is specified here. This value is output in ON interval. 0 is output in OFF interval. If equipment signal is DO or RO, set 1. In case of GO, value specified here is output as binary code.	1-9999
F2 (DATA)	You can display another Stitch Data Detail screen for different condition.	
F3 (LIST)	Go back to Stitch Data List screen.	



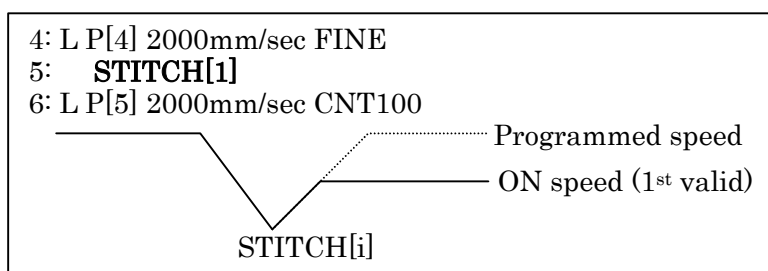
Stitch Data Common screen

◆ In this screen, following items can be set.

Stitch enable	Only when specified signal is ON, stitch process (speed change and signal control) is performed. If this signal is OFF, speed of motion in stitch interval is programmed speed. If index is 0, it corresponds to signal ON.	DI/RI 1-32766
Equipment enable	Only when specified signal is ON, equipment signal is turned on. If this is OFF, only speed control is done. If index is 0, it corresponds to signal ON.	DI/RI 1-32766
Speed unit	Unit of Signal OFF speed and ON speed is specified here.	cm/min mm/sec inch/min

9.33.2.3 Flow of Stitch process

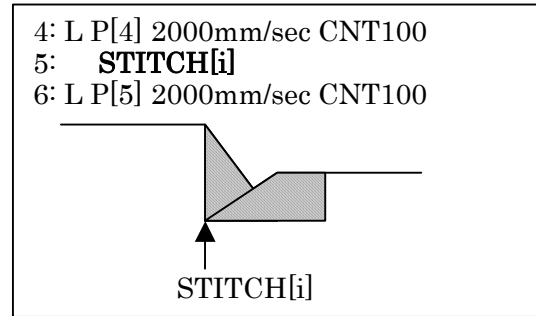
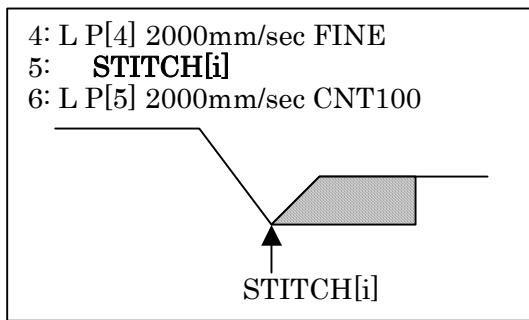
- After execution of STITCH[i], speed of TCP is changed to ON speed of first valid step of stitch condition i. Valid step is a step with non-zero count.
Speed specified by program has no meaning.



- When distance which TCP traveled after STITCH[i] exceeds {approach distance – ON delay (converted to distance) }, equipment signal is output.

CAUTION:

If STITCH[i] instruction is executed before complete of motion, like STITCH[i] just after CNT motion, approach distance is shorter than expected.

**CAUTION:**

ON delay time is interpreted after conversion to distance. On delay time is converted to a distance that TCP travels at On speed and 100% override. Timing of signal output is based on this converted on delay time. Please adjust ON delay time with override 100%.

3. After this, change of speed and ON/OFF of signal is done according to accumulated distance TCP traveled.
4. If motion statement doesn't complete after completion of all steps, TCP moves at OFF speed until STITCH END instruction is executed.

9.33.2.4 Other specifications

- ◆ Several motions in a stitch interval.
Stitch function doesn't care how many motion statements are in a stitch interval. See following example. TCP moves from P[1] to P[3] according to stitch condition
1: J P[1] 100% FINE
2: **STITCH[1]**
3: L P[2] 2000mm/sec CNT100
4: L P[3] 2000mm/sec FINE
5: **STITCH END**
- ◆ Changing stitch condition in a stitch interval
When STITCH[i] instruction is executed in a stitch interval, New stitch condition is used from the instant. Approach distance for new stitch condition is also executed.
- ◆ Halt and resume
When program is halted, equipment signal is turned off.
After resume, the rest of stitch sequence is processed. But timing of stitch may be different from non-halted case due to effect of acceleration and deceleration
- ◆ Single step execution
- ◆ Backward execution
- ◆ Abort of program
- ◆ Change of line
In these state or if one of these is done when stitch is in process, stitch function is automatically disabled. Equipment signal is turned off.
If stitch function is disabled, programmed speed is used.
After resume, stitch process isn't performed. To use stitch function, change execution line to one that is before STITCH[i] instruction and run halted (aborted) program.

- ◆ T1 mode
By doing the followings in T1 mode, INTP-427 WARN (Program, Line) Stitch disable (T1 mode) occurs.
 - Execution of STITCH[i].
 - Pause while executing STITCH, and then Resume after switching to T1 mode.

CAUTION:

If the program is paused while executing stitch motion, and resumed after switching to T1 mode, the resumed motion becomes the same speed as when stitch is enabled.

- ◆ Saving stitch condition
Stitch conditions can be saved by “All of above” in file screen. File name is as follows.
SYSSTCH.SV

9.33.3 Adjustment

Change of speed needs time for acceleration and deceleration. For this reason, it is desired in order to maintain specified speed in ON interval:

1:to accelerate/decelerate with short acceleration/deceleration time

2:to start deceleration at appropriate timing

Speed of robot depends on specified speed. Appropriate acceleration/deceleration time depends on pose of robot even if specified speed is same.

We recommend users to adjust acceleration/deceleration time and timing of deceleration for improved performance.

1. Adjustment of acceleration and deceleration time.

Using ACC instruction, user can adjust acceleration by 1 motion statement.

It is possible to shorten acceleration/deceleration time by adding ACC instruction to motion statements in stitch interval. By ACC instruction, response of robot to change of speed is improved.

```
1: J P[1] 100% FINE
2:   STITCH[1]
3: L P[2] 2000mm/secFINE ACC150
4:   STITCH END
5: L P[3] 2000mm/sec FINE
```

CAUTION:

When Constant path function is valid (\$CPCFG.\$CP_ENABLE=TRUE), you can't adjust acceleration/deceleration between stitch ON and OFF using ACC instruction. If you need to adjust acceleration/deceleration, disable Constant path function (\$CPCFG.\$CP_ENABLE=FALSE) and re-power controller. However, motion path is changed by speed override if you disable Constant path function.

2. Timing of deceleration from OFF interval to ON interval

It is possible to decelerate earlier than usual by hastening deceleration to ON speed.

For this adjustment, following system variables are available.

\$STCH[i]. \$STEP[s]. \$DEC_NUM=1-10 (default value: 2)

OFF interval is divided into pieces of this number in order to change deceleration rate by divisions. Following system variables specify deceleration rates for each division.

$\$STCH[i]$, $\$STEP[s]$, $\$DEC\_RATE[1-10]$ ([1]=100, [2]=0, the others =100)

For each division, speed is set to following speed.

$(OFF\ speed - On\ speed) * \$DEC\_RATE / 100 + On\ speed$

This means that $DEC\_RATE[*] = 100$ corresponds to OFF speed and 0 corresponds to ON speed.

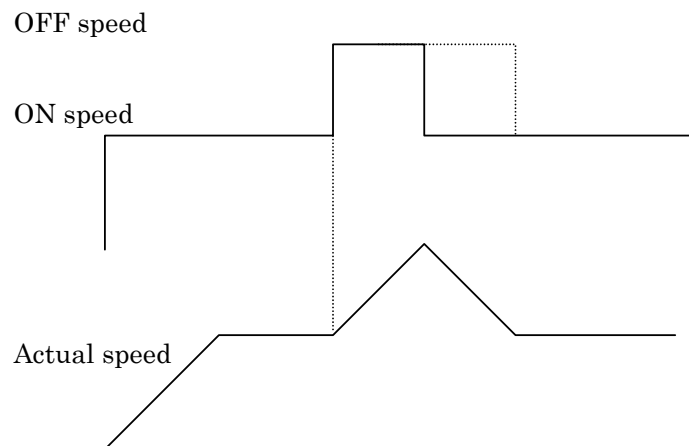
CAUTION:

DEC_RATE value should be in the range from 0 to 100.

Meaning of $\$DEC\_NUM$, $\$DEC\_RATE[]$

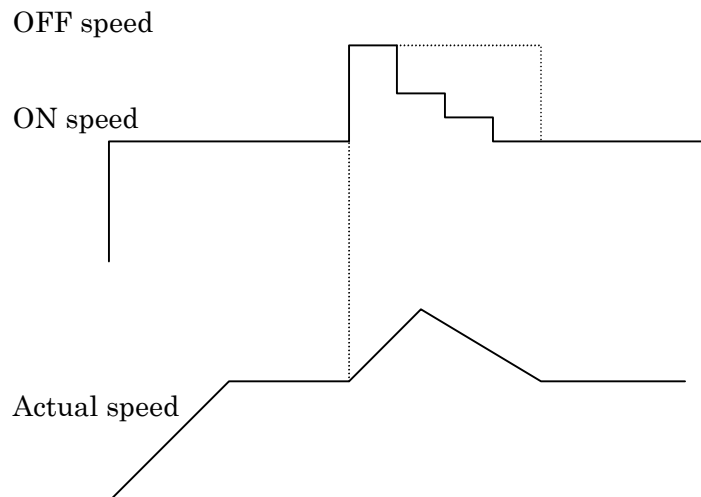
$DEC\_NUM=2$

$DEC\_RATE[1]=100$, $DEC\_RATE[2]=0$



$DEC\_NUM=4$

$DEC\_RATE[1]=100$, $DEC\_RATE[2]=50$, $DEC\_RATE[3]=25$, $DEC\_RATE[4]=0$



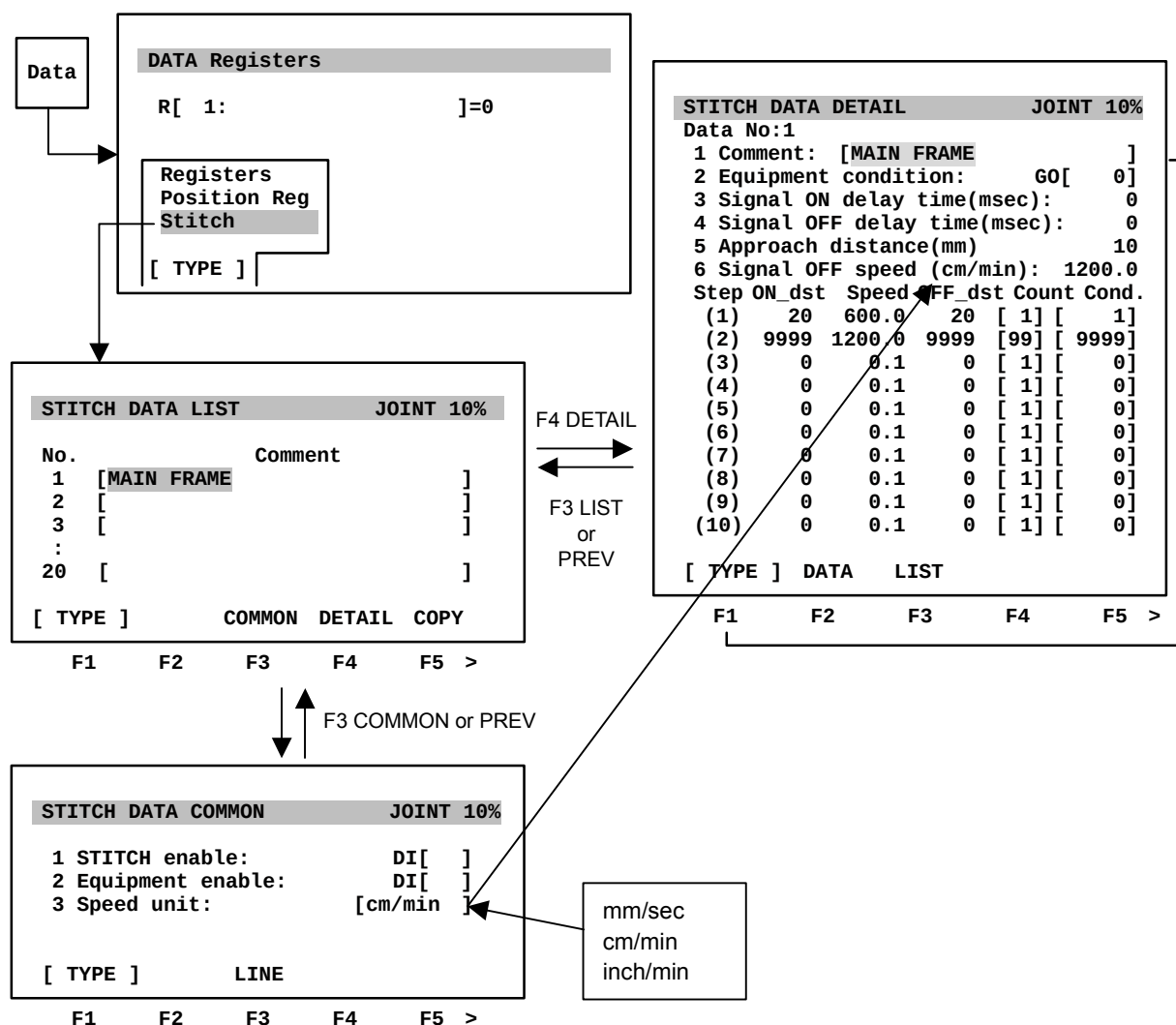
9.33.4 Limitations

- ◆ After E-stop, distance cannot be calculated correctly. After resume distance that TCP moved is calculated. But there may be errors due to overrun after E-stop.
- ◆ JOG doesn't accumulate distance of stitch intervals.
- ◆ In stitch interval, following operations are not available.
JOINT motion
Change of tool frame and user frame.

- ◆ Stitch function cannot be used with following functions.
 - ✓ Continuous turn
 - ✓ Line tracking
 - ✓ Coordinated motion
 - ✓ Remote TCP
 - ✓ Robots which doesn't have Cartesian coordination (Independent axis)
 - ✓ Extended axis
 - ✓ Distance Before (In stitch interval, Distance Before is not available.)
 - ✓ Constant path function

9.33.5 Appendix

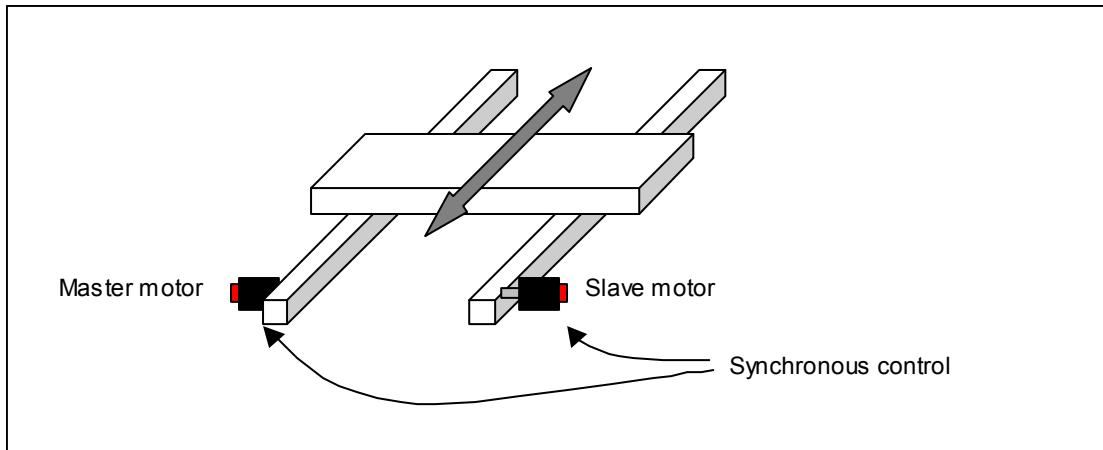
Appendix1 Structure of screens for stitch function



9.34 DUAL MOTOR DRIVE

9.34.1 Introduction

The Dual Motor Drive (Dual Drive) feature allows single axis control of two motors. One motor is called the master motor and the other motor is called the slave motor.



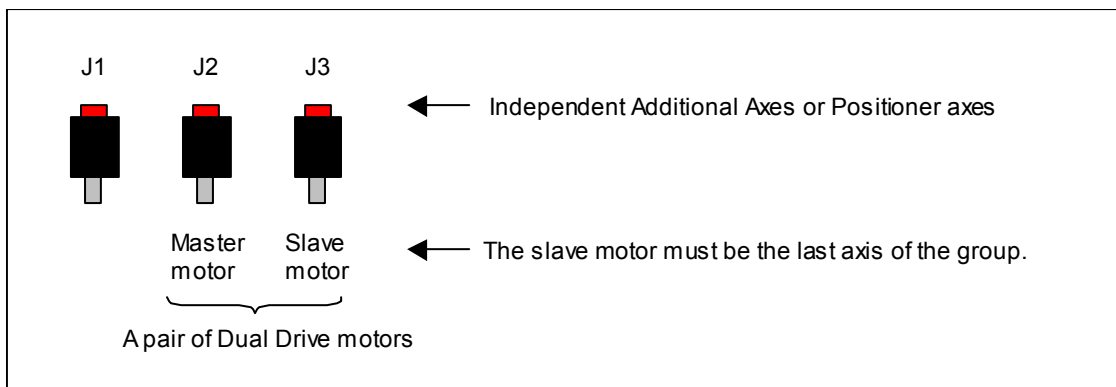
- The main function of the Dual Drive feature is to compensate for synchronous error due to servo delay between master and slave axes.
- Dual Drive feature can realize high-load system which is severe for a single motor.
- Dual Drive feature suppresses the slave motor status on the current position screen, teaching screen, and so on. By jogging or teaching only of the master motor, both motors move in synchronization.

9.34.2 Setup

Preparation

Setup Independent Additional Axes or Positioner axes which you plan to assign as a pair of Dual Drive motors.

- The pair of Dual Drive motors must belong to the same motion group.
- In this step, setup both the master motor and the slave motor individually.
- The slave motor must be the last axis in the group. If there exist multiple Dual Drive motor pairs in one group, the order of the slave axes is unconfined while they must be the last axes in the group.



Dual Drive motor axes setup

Set the following variables, then cycle the power.

System Variable	Description
\$SCR_GRP[g].\$NUM_AXES	This system variable means the total number of axes in the group. Here, count a pair of Dual Drive motors as one axis. In other words, change this variable to the original number minus the number of Dual Drive motor pairs.
\$SCR_GRP[g].\$NUM_ROB_AXS	Change this variable in the same manner as \$NUM_AXES.
\$SCR_GRP[g].\$NUM_DUAL	Enter the number of Dual Drive motor pairs. The maximum value of this system variable is 3.
\$DUAL_DRIVE[g].\$M_AXIS_NUM[n]	Enter the axis number of the master motor. "n" is an index of the Dual Drive motor pair. For the first pair, n = 1.
\$DUAL_DRIVE[g].\$S_AXIS_NUM[n]	Enter the axis number of the slave motor.

[Example 1]

System configuration:

- Group 2 has three motors as Independent Additional Axes.
- J2-J3 of Group 2 are a pair of Dual Drive motors.
- J2 is the master motor and J3 is the slave motor.

Setting:

- \$SCR_GRP[2].\$NUM_AXES = (3 -->) 2
- \$SCR_GRP[2].\$NUM_ROB_AXS = (3 -->) 2
- \$SCR_GRP[2].\$NUM_DUAL = 1
- \$DUAL_DRIVE[2].\$M_AXIS_NUM[1] = 2
- \$DUAL_DRIVE[2].\$S_AXIS_NUM[1] = 3

J2-J3 pair is treated as "J2" when jogging or teaching.

[Example 2]

System configuration:

- Group 2 has four motors as Independent Additional Axes.
- J1-J3 and J2-J4 of Group 2 are pairs of Dual Drive motors.
- J1 and J2 are the master motors and J3 and J4 are the slave motors.

Setting:

- \$SCR_GRP[2].\$NUM_AXES = (4→) 2
- \$SCR_GRP[2].\$NUM_ROB_AXS = (4→) 2
- \$SCR_GRP[2].\$NUM_DUAL = 2
- \$DUAL_DRIVE[2].\$M_AXIS_NUM[1] = 1
- \$DUAL_DRIVE[2].\$M_AXIS_NUM[2] = 2
- \$DUAL_DRIVE[2].\$S_AXIS_NUM[1] = 3
- \$DUAL_DRIVE[2].\$S_AXIS_NUM[2] = 4

J1-J3 pair is treated as "J1", and J2-J4 pair as "J2" when jogging or teaching.

9.35 MULTI ROBOT CONTROL FUNCTION

9.35.1 Introduction

This function is to control two robots with a single robot controller.

The two robots can move synchronously or independently according to robot program.

Multi Robot Control function has the following merit.

- No interlock wiring between two robots
If robots are controlled by some controllers, many input and output signal lines are needed to confirm interlock between the robots.

In Multi Robot Control system, the two robots can confirm interlock easily with internal registers. Interlock wiring is not needed.

- Easy teaching for two robots system
The two robots can be controlled from one teach pendant.
Operator does not need to handle two teach pendants.
- Safety system
In Multi Robot Control system, the two robots share one emergency stop circuit.
So the emergency stop always takes effect for the two robots at the same time.
Also, operator selects one robots to move by manual from single teach pendant.
So when operator is teaching one robot, another robot is not selected and it is always stopped.

9.35.1.1 Specification

- Robot Control Type
The following type of robot control method is available in Multi Robot Control.
 - Single Robot control
Multi Robot Independent Control
Two robots can move independently as if these are controlled by different controllers.
 - Multi Robot Synchronous Control
Two robots are controlled to move synchronously.
- Interlock
The two robots can confirm interlock with internal variable like register.
- Teaching operation
The two robots share a single teach pendant.
The pendant is enabled only for one robot at a time.
Operator can select enabled robot from teach pendant.
- Robot Isolate Function
In robot controller which is equipped with operator panel for Multi Robot Control, there are Connect/Isolate keys for each robots on the operator panel. This key is to isolate the robot from the controller.
When robot is isolated from the controller, then alarm is posted and the group motion of the robot is disabled automatically.
When the robot is connected to the controller, then alarm is posted and the group motion of the robot is enabled automatically.
- End Effector Interface
RI[1]-RI[8] and RO[1]-RO[8] are mapped for GROUP1 robot.
RI[9]-RI[16] and RO[9]-RO[16] are mapped for GROUP2 robot.
- UOP Interface
UOP signals are extended to 22 input signals and 26 output signals for Multi Robot Control.
Normal robot controller uses 18 input and 20 output for UOP signals.

This means that Multi Robot Control function needs to use additional 4 input and 6 output signals for UOP compared to normal robot controller.

So in case of Multi Robot Control, available signal resources for normal digital signals (DI/DO) are reduced from normal robot controller case by 6 input and 6 output signals.
For detail, please refer "9.35.3. User Operator Panel(UOP)".

- Emergency Stop Input
There is one emergency stop circuit in robot controller.
The two robots share this circuit.
- Over Travel/Hand Broken signal
Each robot has its own OVERTRAVEL and HANDBROKEN interface.

9.35.1.2 Software Options

The following software options are needed for Multi Robot Control function.

Multi Motion	A05B-2500-J601
Multi Robot Control	A05B-2500-J605

Robot-Type software options are needed according to the number of actual robot.

Example)

Multi Robot Control function with two R-2000iB/165F needs the following two robot software options.

R-2000iB/165F	A05B-2500-H738
R-2000iB/165F	A05B-2500-H738

9.35.2 Standard Operator Panel

Robot Controller for Multi Robot Control has additional keys on operator panel as "ROBOT#1 CONNECT/ISOLATE" and "ROBOT#2 CONNECT/ISOLATE".

Common to the two robots	Output	Battery Alarm
		TP Enabled
		Alarm
		Remote
	Input	Remote/Local key
		Alarm Reset
		Start
		Hold
		User1
		User2
For ROBOT#1	Input	Connect/Isolate key *
For ROBOT#2	Input	Connect/Isolate key *

- * Multi Robot Control function is available with normal robot controller which does not equip the above additional keys too. To use this function with normal controller, please set system variable \$ROBOT_ISOL to 0.

9.35.2.1 BATTERY ALARM

BATTERY ALARM indicator is ON when a voltage of backup battery for controller's SRAM module is lower than normal level.

If it is ON, then please exchange the backup battery. Please note to keep robot controller power ON at this battery exchange.

9.35.2.2 TP ENABLED

TP ENABLED indicator is ON when TP is enabled.

9.35.2.3 ALARM

ALARM indicator is ON when controller is under alarm status.

9.35.2.4 REMOTE

REMOTE indicator is ON when the following conditions are satisfied.

- Remote condition is satisfied (Remote condition is described later)
- Not alarm status

9.35.2.5 REMOTE/LOCAL Key

Remote mode or local mode of the controller is selected by this key.

In remote mode, if remote condition is satisfied, then robot program can be started by UI signal input.

In local mode, if operator panel condition is satisfied, then robot program can be started by operator panel START button.

Remote condition is satisfied when the all following conditions are satisfied at the same time.

- TP enabled key is OFF
- REMOTE/LOCAL key on operator panel is set to remote.
- UI *SFSPD input signal is ON.
- UI ENBL input signal is ON.
- System variable \$RMT_MASTER is set to 0 (0 means External Device)

Operator panel condition is satisfied when the following conditions are satisfied at the same time.

- TP enabled key is OFF
- REMOTE/LOCAL key on operator panel is set to local.
- UI *SFSPD input is ON.

9.35.2.6 ALARM RESET

When this button is pressed, alarm status is reset for the two robot.

If servo is OFF before reset, then servo is ON by alarm reset.

9.35.2.7 START

When this button is pressed, then selected robot program is started.

About selected program, please refer "9.35.6.Select program".

9.35.2.8 HOLD

When this button is pressed, then all the running robot program is paused.

9.35.2.9 USER1/USER2

These two buttons are used as utility interface. For example, MACRO program can be assigned to USER1 or USER2 buttons.

9.35.2.10 CONNECT/ISOLATE Key

There are two CONNECT/ISOLATE keys for robot1 and robot2.

When a robot is disconnected, then a alarm is posted and the group motion of the robot is disabled automatically.

When a robot is connected from disconnect status, then alarm is posted and the group motion of the robot is enabled automatically.

The following alarms are posted at the above Connect/Disconnect operation.
The severity of these alarms are both PAUSE.G.

System-029 Robot was connected (G:1)
System-030 Robot was isolated (G:1)

WARNING :

For safety, before connecting or disconnecting , please move the robots to safe position and confirm all the programs are ended.

9.35.3 User Operator Panel(UOP)

Tow robots uses the same User Operator Panel(UOP) as single robot system.

INPUT SIGNAL	OUTPUT SIGNAL
*IMSTP	CMDENBL#1
*HOLD#1	SYSRDY#1
*SFSPD	PROGRUN#1
CSTOPI	PAUSED#1
FAULT_RESET	HELD#1
START#1	FAULT#1
HOME	ATPERCH
ENBL	TPENBL
RSR1/PNS1	BATALM
RSR2/PNS2	BUSY
RSR3/PNS3	ACK1/SNO1
RSR4/PNS4	ACK2/SNO2
RSR5/PNS5	ACK1/SNO1
RSR6/PNS6	ACK1/SNO1
RSR7/PNS7	ACK1/SNO1
RSR8/PNS8	ACK1/SNO1
PNSTROBE#1	ACK1/SNO1
PROD_START#1	ACK1/SNO1
	SNACK
	RESERVE
*HOLD#2	CMDENBL#2
START#2	SYSRDY#2
PNSTROBE#2	HELD#2
PROD_START#2	FAULT#2
	PROGRUN#2
	PAUSED#2

Total 22 points (In single robot system, there are 18 UOP's)

Total 26 points (In single robot system, there are 20 UOP's)

I/O Configuration:

In case that DI/DO are 40 points

Specialized input (UI)	22 points
General-purpose input (DI)	18 points
Specialized output (UO)	26 points
General-purpose output (DO)	14 points

In case that DI/DO are 96 points.

Specialized input (UI)	22 points
General-purpose input (DI)	74 points
Specialized output (UO)	26 points
General-purpose output (DO)	70 points

9.35.3.1 Modification and Addition of Signal

In Multi Robot Control Function, the meaning of the following signals are changed.

Input: HOLD#1: UI[2] HOLD is changed to HOLD#1.
 START#1: UI[6] START is changed to START#1.
 PNSTROBE#1: UI[17] PNSTROBE is changed to PNSTROBE#1.
 PROD_START#1: UI[18] PROD_START is changed to PROD_START#1.

Output: CMDENBL#1: UO[1] CMDENBL is changed to CMDENBL#1.
 SYSRDY#1: UO[2] SYSRDY is changed to SYSRDY#1.
 PROGRUN#1: UO[3] PROGRUN is changed to PROGRUN#1.
 PAUSED#1: UO[4] PAUSED is changed to PAUSED#1.
 HELD#1: UO[5] HELD is changed to HELD#1.
 FAULT#1: UO[6] FAULT is changed to FAULT#1.

In Multi Robot Control Function, the following signals are added.

Input: HOLD#2: added as UI[19].
 START#2: added as UI[20].
 ONSTROBE#2: added as UI[21].
 PROD_START#2: added as UI[22].

Output: CMDENBL#2: added as UO[21].
 SYRSY#2: added as UO[22].
 HELD#2: added as UO[23].
 FAULT#2: added as UO[24].
 PROGRUN#2: added as UO[25].
 PAUSED#2: added as UO[26].

9.35.3.2 HOLD#1, #2

Multi Robot System has two HOLD signals (#1,#2). The function of HOLD signal is changed by setting system variable "\$MULTI_ROBO.\$HOLD_TYPE".

- In case of \$MULTI_ROBO.\$HOLD_TYPE = 0 (Default)
 If either of HOLD signals (HOLD#1,#2) is OFF, all the executing programs are paused.

- In case of \$MULTI_ROBO.\$HOLD_TYPE = 1
If HOLD#1 is OFF, "Program#1" is paused.
If HOLD#2 is OFF, "Program#2" is paused.
The executing program, which is neither Program#1 nor Program#2, can NOT be held. Please turn off ENBL to pause the program.

Please refer "9.35.6 Select program" about "Program#1" and "Program#2".

9.35.3.3 START#1, #2

Note)

Please use the Multi Robot System in condition that "START for CONTINUE only:" in system configuration screen is "TRUE". Only paused program can be started by SRART signal under this setting. In case of starting the ended program, use RSR or PROD_START signal, please.

Multi Robot System has two START signals (START#1,#2).

"Program#1" is started when START#1 signal is turned off.

"Program#2" is started when START#2 signal is turned off.

Please refer "9.35.6 Select program" about "Program#1" and "Program#2".

The program is continued to execute from suspended line in case that START signal is inputted with suspended program

The program, which is aborted (neither executing nor paused), can NOT be started and the message "PROG-023 Task is not paused" is displayed in TP screen in case that START signal is inputted with the ended program (neither executing nor suspended).

9.35.3.4 RSR

Multi Robot System shares RSR 1-8 with PROGRAM#1,#2 and has two queues corresponded to RSR1-4 and RSR5-8.

Please refer to "9.35.8.5 Robot Service Request(RSR)".

9.35.3.5 PNSTROBE#1, #2, PROD_START#1, #2

Multi Robot System has two PNSTROBE signals (PNSTROBE#1,#2) and two PROD_START signals (PROD_START#1,#2).

Please refer to "9.35.8.6 Program Number Select(PNS)".

9.35.3.6 CMDENBL#1, #2

Multi Robot System has two CMDENBL signals (CMDENBL#1,#2).

CMDENBL#1 is ON if all the following conditions are satisfied.

- Remote condition is satisfied.
- No alarm. (Both of FAULT#1 and FAULT#2 are OFF.)
- No single step.
- Connection with Robot#1.

CMDENBL#2 is ON if all the following conditions are satisfied.

- Remote condition is satisfied.
- No alarm. (Both of FAULT#1 and FAULT#2 are OFF.)
- No single step.
- Connection with Robot#2.

Remote Condition:

- The Teach Pendant enable switch is set OFF.
- The remote switch on the Operator Panel is set to remote position.
- The *SFSPD input is ON.
- The ENBL input is ON.
- The system variable \$RMT_MASTER is 0 (UOP).

9.35.3.7 SYSRDY#1, #2

Multi Robot System has two SYSRDY signals (SYSRDY#1,#2).

SYSRDY#1 is ON if all the following conditions are satisfied.

- All the axes are ready.
- Connection with Robot#1.

SYSRDY#2 is ON if all the following conditions are satisfied.

- All the axes are ready.
- Connection with Robot#2.

9.35.3.8 ATPERCH

The specified signal ATPERCH is set to ON when the robot of motion group 1 is at the reference position 1. Please assign reference position output except motion group 1

9.35.3.9 PROGRUN#1, #2, PAUSED#1, #2

In Multi Robot system, PROGRUN#1, #2 and PAUSED#1, #2 shows the program execution status as follows.

- When program is running:
PROGRUN#1
 - In case that the program selected as PROGRAM#1 is executing.

PROGRUN#2

- In case that the program selected as PROGRAM#2 is executing.

In case that the program, selected as neither PROGRAM#1 nor PROGRAM#2, is executing, Both of PROGRAM#1 and PROGRAM#2 is outputted.

- When program is paused:
PAUSED#1
 - In case that the program selected as PROGRAM#1 is paused.

PAUSED#2

- In case that the program selected as PROGRAM#2 is paused.

In case that the program, selected as neither PROGRAM#1 nor PROGRAM#2, is executing, Both of PAUSED#1 and PAUSED#2 is outputted.

9.35.3.10 HELD#1, #2

Multi Robot Control System has two HELD signal (HELD#1,#2).

HELD#1 is ON if one of the following conditions is satisfied.

- HOLD#1 is OFF.
- The HOLD switch on the Teach Pendant is pushed.
- The HOLD switch on the Operator Panel is pushed.

HELD#2 is ON if one of the following conditions is satisfied.

- HOLD#2 is OFF.
- The HOLD switch on the Teach Pendant is pushed.
- The HOLD switch on the Operator Panel is pushed.

9.35.3.11 FAULT#1, #2

Multi Robot Control System has two FAULT signal (FAULT#1,#2).

FAULT#1 turns ON when the alarm occurs by “Program#1” execution or other than program execution.

FAULT#2 turns ON when the alarm occurs by “Program#2” execution or other than program execution.

When system is in alarm status, one of FAULT#1 or FAULT#2 must be ON.

9.35.4 ROBOT I/O

System has two sets of ROBOT I/O for each robot.

System has two ROT/HBK inputs for each robot.

System has two PPABN inputs for each robot.

Robot	name	Usage	Robot	name	Usage
# 1	R I 0 1	general-purpose	# 1	R O 0 1	general-purpose
	R I 0 2			R O 0 2	
	R I 0 3			R O 0 3	
	R I 0 4			R O 0 4	
	R I 0 5			R O 0 5	
	R I 0 6			R O 0 6	
	R I 0 7			R O 0 7	
	R I 0 8			R O 0 8	
# 2	R I 0 9	general-purpose	# 2	R O 0 9	general-purpose
	R I 1 0			R O 1 0	
	R I 1 1			R O 1 1	
	R I 1 2			R O 1 2	
	R I 1 3			R O 1 3	
	R I 1 4			R O 1 4	
	R I 1 5			R O 1 5	
	R I 1 6			R O 1 6	

9.35.5 Toggling robot

If Multi Robot Control Function is installed, operator uses several robots by selecting robot that can be operated by teach pendant.

To toggle robot which can be operated by teach pendant, change motion group by following procedure.

Changing motion group from function menu.

- Press FCTN key to display function menu.
- Every time you select “CHANGE GROUP”, motion group is changed to next group. For example, from group1 to group2, group2 to group3, and group3 to group1. The last motion group is changed back to group1.

Changing motion group from JOG menu.

- Press COORD key while SHIFT key is pressed. Jog menu is displayed.
- Set cursor to “Group” in Jog menu and press number key. Motion group is changed to the one you choose.

Currently selected motion group is displayed on teach pendant like “G1”, “G2”.

		Motion group	
Select	G2	JOINT	10%
Select	G1	JOINT	10%
65432 bytes free		1/4	
1	SAMPLE1	#1	[SAMPLE PROGRAM1]
2	SAMPLE2	#2	[SAMPLE PROGRAM2]
3	PROG001		[PROGRAM001]
4	PROG002		[PROGRAM002]
[TYPE]	CREATE	DELETE	MONITOR [ATTR]>
COPY	DETAIL	LOAD	SAVE AS PRINT >

By changing motion group, “Program #1” and “Program #2” can be also toggled automatically. Please refer to “95.3.6 Select program” in detail.

9.35.6 Select program

Multi Robot Control Function enables user to select different program for each of two robots.

Program that is selected for group 1 is referred as “Program #1”

Program which is selected for group 2 is referred as “Program #2”

“Program#1” is run by UOP START#1, PROD_STAR#1.

“Program#2” is run by UOP START#2, PROD_STAR#2.

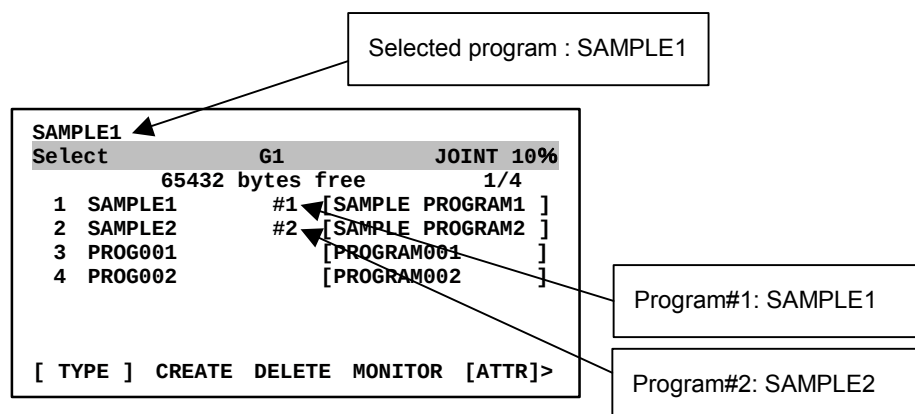
On program selection screen, “#1” is displayed to the right of program name that is used as “Program #1”.

For “Program #2”, “#2” is displayed to the right of program name which is used as “Program #2”.

On the contrary, a program that is displayed on program edit screen is referred as “selected program”.

FWD/BWD keys on teach pendant or start button on Operator Panel run “selected program”.

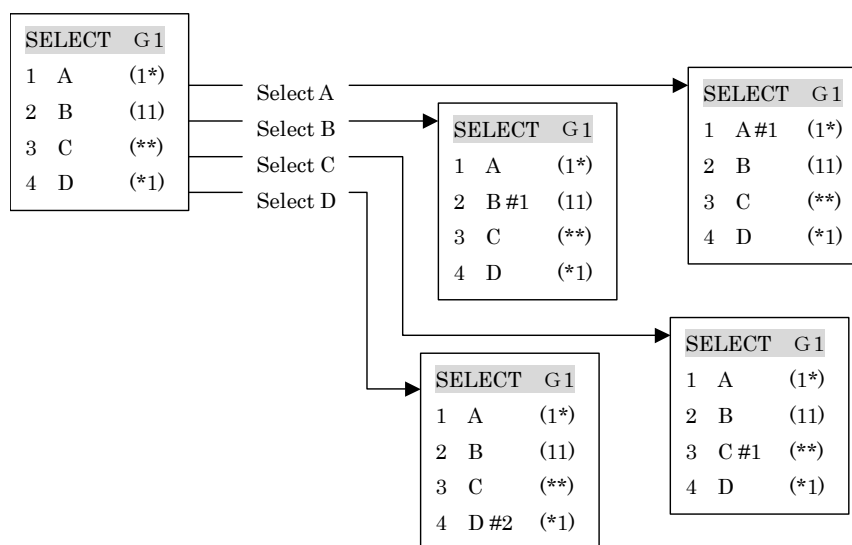
“Selected program” is displayed on left side of second line of TP screen.



If you set cursor to a program and presses ENTER, “selected program” is changed to the program. At the same time, the program is also set as “Program #1” or “Program #2” automatically, according to motion group the program has.

Suppose the program has both group 1 and 2 or the program has no group. In these cases, if group 1 is selected on teach pendant, the program is set to “Program #1”. If group 2 is selected, the program is set to “Program #2”. If neither group1 nor group2 is selected, which the program going is to be is decided automatically, according to motion group that the program has, execution state of program, etc.

Consequently, if you select a program on program list screen, the program becomes “selected program”. At the same time, the program is always chosen as either “Program #1” or “Program #2”.



- (1*) The program has only group 1.
- (*1) The program has only group 2.
- (11) The program has group 1 and 2.

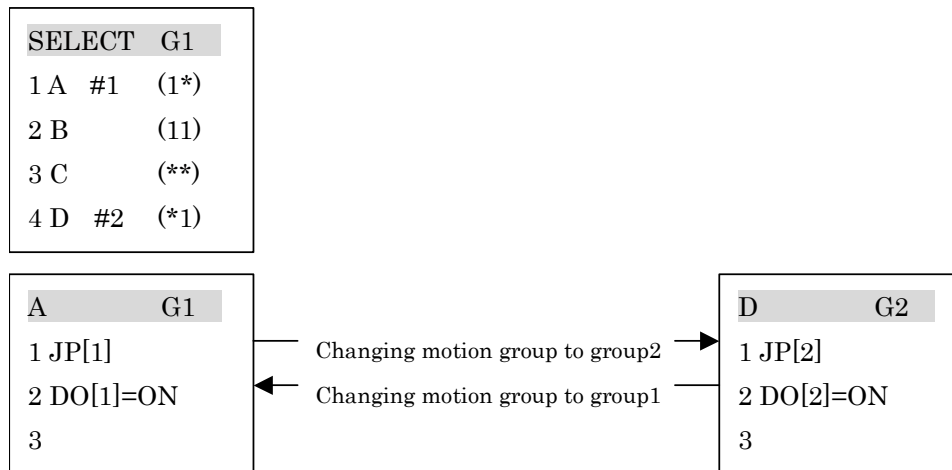
By changing motion group, “selected program” is also changed according to selected motion group. Changed “selected program” is displayed on program edit screen.

For example, suppose “Program #1 “ has only group1 and “Program #2 has” only group2.

By Selecting group1, “Program #1”

By Selecting group2, “Program #2”

automatically becomes “selected program” and the program is displayed on program edit screen.



If “Selected program” has both group1 and group2 or “selected program” has no motion group, change of motion group doesn’t change “selected program”.

To change “selected program” in these cases, display program list screen, set cursor to a program that you want to display and press ENTER.

9.35.7 Teaching and modifying position

Teaching of position is done by selecting standard motion instruction. By this action, position data and motion instruction are taught at the same time.

To modify position data, press and hold in SHIFT key and press F5, TOUCHUP to record current position as new position

In case of Multi Robot Control Function, isolation key for robot#1(group1) and that for robot #2(group2) are on operation panel.

By teaching and modifying position data, current position is not recorded to position data for robot whose isolation key is on isolation side.

When current position is taught, following message is displayed.

When current position of both group1 and 2 is taught,
“Position has been recorded to P[1].”

When only current position of group1 is taught,
“Position has been recorded to P[1](G:1)”

When only current position of group2 is taught,
“Position has been recorded to P[1](G:2)”

When current position of neither group1 nor group2 is taught,
No message is displayed.

9.35.8 Run program

9.35.8.1 Run program by Teach Pendant

The selected program is started when start is required by Teach Pendant.

Please refer “9.35.6 Select program” for ‘the selected program’.

Program does not run at the following situation.

- If the program has any motion group, the program that does not have the selected motion group on teach pendant can not start.

If you try to run this program, the following alarm occurs.

“TPIF-044 Program is unsuitable for robot”

9.35.8.2 Run program by operator panel

When Teach Pendant enable switch is OFF and REMOTE/LOCAL key on Operator Panel is on REMOTE, you can start program by Operator Panel.

The selected program is started when start is required by Operator Panel.

Please refer “9.35.6 Program select” for ‘the selected program’.

The selected program always start regardless of the selected motion group.

9.35.8.3 Run two programs at the same time

Both “Program#1” and “Program#2” can run at the same time by multi robot control function.

Example

SELECT	G1
1 A	#1 (1*)
2 B	(11)
3 C	(**)
4 D	(*1)



Select program ‘A’



A	G1
1	JP[1]
2	DO[1]=ON
3	



Run program by FWD/BWD key of Teach Pendant or START button of Operator Panel.



Program ‘A’ is started.



Change motion group by FCTN menu or JOG menu.



Selected motion group becomes group2, and program ‘D’ is displayed in Edit menu automatically.



D	G2
1	P[2]
2	DO[2]=ON
3	

↓
 Run program by FWD/BWD key of Teach Pendant or START button of Operator Panel.
 ↓
 Program 'D' is started, and program 'A' is still running.

9.35.8.4 Select program at running or paused

When "Program#1" is paused, if you select another program as "Program#1":

- When Teach Pendant Enable switch is ON, the original "Program#1" is aborted, and the selected program becomes "Program#1".
- When Teach Pendant Enable switch is OFF, you can not select the program, and the message "TPIF-013 Other program is running" is displayed.

When "Program#1" is running, if you select another program as "Program#1", you can not select the program regardless of Teach Pendant Enable switch, and the message "TPIF-013 Other program is running" is displayed.

"Program#2" is same as "Program#1".

Note: The above specification is available only when system variable "\$SCR.\$MAXNUMTASK" is 2 (Default). If "\$SCR.\$MAXNUMTASK" is more than 3, you can select another program even though "Program#1" and "Program#2" are running or paused. But in this case, it is difficult to know what program is running now. Please not change "\$SCR.\$MAXNUMTASK" if you don't need to run more than 3 programs at the same time.

9.35.8.5 Robot Service Request (RSR)

You can select RSR or PNS(Optional).
 If PNS is selected, RSR is not available.

In single robot system, when RSR1-8 are turned on, the program corresponded to the RSR number is started if the program is aborted. If another program is running or paused, the request is joined to the queue. And the program is started when the running program is completed.

In multi robot system, RSR1-8 is divided to RSR1-4 for Robot#1 and RSR5-8 for Robot#2. There are two queues for RSR1-4 and RSR5-8. By this function, two programs can run at the same time.

The program started by RSR1-4 becomes "Program#1".

The program started by RSR5-8 becomes "Program#2".

In the following situation, program is not started.

- The program started by RSR1-4 does not have motion group 1 and it has motion group 2.
- The program started by RSR5-8 does not have motion group 2 and it has motion group 1.

In this case, message "TPIF-44 Program is unsuitable for robot" is displayed.

9.35.8.6 Program Number Select (PNS)

You can select RSR or PNS(Optional).
 If RSR is selected, PNS is not available.

In multi robot system, there are PNSTROBE#1, PNSTROBE#2, PROD_START#1 and PROD_START#2.

The program selected by PNSTROBE#1 becomes “Program#1”. This program can be started by PROD_START#1 and it is resumed from pause by START#1.

The program selected by PNSTROBE#2 becomes “Program#2”. This program can be started by PROD_START#2 and it is resumed from pause by START#2.

In the following situation, program is not selected.

- The program selected by PNSTROBE#1 does not have motion group 1 and it has motion group 2.
- The program selected by PNSTROBE#2 does not have motion group 2 and it has motion group 1.

In this case, message “TPIF-44 Program is unsuitable for robot” is displayed.

Examples:

The followings are recommended motion group of the selected program.

Example1:

“Program#1” had motion group 1 only.

“Program#2” had motion group 2 only.

Example2:

“Program#1” had motion group 1 and 2.

“Program#2” does not have motion group.

Operation:

The following is the operation example to start two programs by PNS.

- Select PNS0001 by PNSTROB#1, and run it by PROD_START#1. PNS0001 is selected as “Program#1”.
- Select PNS0002 by PNSTROB#2, and run it by PROD_START#2. PNS0002 is selected as “Program#2”.
- Use HOLD#1 to pause PNS0001. And use START#1 to resume PNS0001.
Use HOLD#2 to pause PNS0002. And use START#2 to resume PNS0002.
(Please set “\$MULTI_ROBO.\$HOLD_TYPE” 1 to pause PNS0001 and PNS0002 separately.)

9.35.9 Stop program

9.35.9.1 Pause program by Teach Pendant

When hold button on Teach Pendant, all running programs are paused.

9.35.9.2 Pause program by Operator Panel

When hold button on Operator Panel, all running program are paused.

9.35.9.3 Pause program by UOP

When HOLD#1 or HOLD#2 turns on, running program is paused.

Please read “9.35.3.2 HOLD #1, #2” for detail.

9.35.9.4 Abort program by UOP

If 'CSTOPI for ABORT' in System Configuration menu is TRUE, programs are aborted when CSTOPI turns on.

The default setting of 'CSTOPI for ABORT' is FALSE.

And 'Abort all programs by CSTOPI' setting defines the function of CSTOPI as follows.

When 'Abort all programs by CSTOPI' is FALSE (Default)

Only the selected program is aborted by CSTOPI.

When 'Abort all programs by CSTOPI' is TRUE

All programs are aborted by CSTOPI.

TRUE for 'CSTOPI for ABORT' TRUE' is recommended for multi robot control function.

9.35.10 System variables

Name	\$MULTI_ROBO.\$MLT_ENABLE
Default	TRUE
Type	BOOLEAN
Protection	Protected
Range	TRUE/FALSE
Meaning	Enable/Disable multi robot control function. TRUE: Multi robot control function is enabled. FALSE: Multi robot control function is disabled.

Name	\$MULTI_ROBO.\$HOLD_TYPE
Default	0
Type	INTEGER
Protection	Not protected
Range	0/1
Meaning	Select the function of HOLD#1 and HOLD#2. 0: All programs are paused when HOLD#1 or HOLD#2 turns OFF. 1: "Program#1" is paused when HOLD#1 turns OFF. "Program#2" is paused when HOLD#2 turns OFF.

The following system variable must be set when multi robot control function is enabled.

\$SCR.\$ITP_TIME	12

Note) The different value may be recommended according to the robot configuration.

9.36 Robot Isolation function

9.36.1 Introduction

Robot Isolation function can make certain robot be isolated state by the flick of a switch when multiple robot is controlled by single controller. In teaching in the system with multiple robot, when just partial robots are taught, it is possible to keep the other robots unmoving. On this account, user can have safety and convenience in teaching

And this is optional function(J544).

Specification

- Isolated state
Robot turns into machine lock state by isolating. Servo power is turned off and robot comes into unmoving state. And user can't teach new point to isolated robot. By unlocking isolated state, robot turns into normal state.
- Method of operating
Isolation switch is prepared up to 5 on the operation panel. And by Isolation switch, user can change certain robot state between isolated and connected. When robot is isolated, robot turns into machine lock state after alarm is displayed. And When robot is connected, robot turns into normal state after alarm is displayed.
- Setting
Each Isolation switch corresponds certain motion group. User can see correspondence between Isolation switch and motion group by system variable \$ROBOT_ISOLC.

9.36.2 Method of operating

9.36.2.1 Operation panel

As is often the case, Isolation switch is mounted on Operation panel.

By turning switch ENABLE side, robot become operable.

By turning switch DISABLE side, robot become isolated.

9.36.2.2 SOP

Each Isolation switch is assigned to input of SOP as the following:

Isolation switch 1 – SI[10]

Isolation switch 2 – SI[11]

Isolation switch 3 – SI[14]

Isolation switch 4 – SI[15]

Isolation switch 5 – SI[12]

(ON: connected, OFF: isolated)

9.36.3 System variable: \$ROBOT_ISOLC

Depending on ON/OFF of each Isolation switch, corresponding motion group state become normal / machine lock. Correspondence between Isolation switch and motion group is decided by system variable \$ROBOT_ISOLC.

Which robot's power current to servo is turned ON/OFF by Isolation switch is decided by hardware configuration. Therefore, it is decided uniquely which motion group should be machine lock state by software when certain isolation switch is turned.

So \$ROBOT_ISOLC is needed to be set just one time when robot system is established.

\$ROBOT_ISOLC have 5 alignment, and each one corresponds one Isolation switch. If the number of Isolation switch is smaller than 5, \$ROBOT_ISOLC alignments are available as many as the number of Isolation switch, and the other alignments is ineffective.

SW1:	\$ROBOT_ISOLC[1]
SW2:	\$ROBOT_ISOLC[2]
SW3:	\$ROBOT_ISOLC[3]
SW4:	\$ROBOT_ISOLC[4]
SW5:	\$ROBOT_ISOLC[5]

The each bit of \$ROBOT_ISOLC value is equivalent to each motion group.

Bit31	...	Bit1	Bit0
Motion Group32	...	Motion Group2	Motion Group1

For example, the default values of \$ROBOT_ISOLC are set as the following, it is assumed that one Isolation switch is equivalent to one motion group.

Example1 (Default)

```

SW1: $ROBOT_ISOLC[1] = 1
SW2: $ROBOT_ISOLC[2] = 2
SW3: $ROBOT_ISOLC[3] = 4
SW4: $ROBOT_ISOLC[4] = 8
SW5: $ROBOT_ISOLC[5] = 16

```

In case that SW1 is equivalent to motion group 1,4,
 SW2 is equivalent to motion group 2,5,
 and SW3 is equivalent to motion group 3,
 \$ROBOT_ISOLC value is the following.

Example 2

```

SW1: $ROBOT_ISOLC[1] = 9
SW2: $ROBOT_ISOLC[2] = 18
SW3: $ROBOT_ISOLC[3] = 4
SW4: $ROBOT_ISOLC[4] = 0
SW5: $ROBOT_ISOLC[5] = 0

```

9.36.4 Teaching to motion instruction

When user teaches motion instruction, current position isn't recorded for isolated robot.
 And when teaches target position again by SHIFT key + F5 "TOUCHUP", position isn't recorded similarly.

Then, the message that informs isolated group can't be taught is displayed.

```

SYST-168 Robot(G:1) is isolated
-----

  1: J P[1] 100% FINE
  2: J P[1] 100% FINE
[End]

Not recorded to isolated Groups.
POINT                                TOUCHUP

```

9.37 Multi UOP Interface Function

9.37.1 Introduction

Multi UOP Interface function enable us to use up to 5 UOP sets that have a connection with start of program. Therefore, It is possible to control (select, execute, pause and so on) multiple program independently with a single R-30iA controller or R-30iA Mate controller.

9.37.1.1 Specification

The number of usable UOP set
Up to 5 UOP sets are usable.

UOP Interface

The number of UOP signals depend on how many UOP sets is used.

Number of set	1	2	3	4	5
Input	18	23	32	41	50
Output	20	26	36	46	56

9.37.1.2 Software Options

The following software option is needed for Multi UOP Interface function.

Mult UOP Interface	A05B-2500-J964

If 2 or more motion groups (multiple robot, independent axis and so on) is used, the following software option is needed.

Multi-Group Motion	A05B-2500-J601

But, if the following software option is ordered, the above 2 softwares option is not needed to be ordered because this option includes the above.

Multi-Arm Pkg	A05B-2500-R671

9.37.1.3 The difference from Multi Robot Control function about UOP

If Multi Robot Control (J605) option is ordered, 2 UOP sets are usable. The difference between Multi Robot Control function and Multi UOP Interface function is the following:

- Multi Robot Control function enable us to use only 2 UOP sets, but Multi UOP Interface function enable us to use up to 5 UOP sets.
- Multi Robot Control function has one CSTOPI in all, but Multi UOP Interface has one CSTOPI signal per UOP set. (Therefore, even if 2 UOP sets are used, Multi UOP Interface function has more UI signals than Multi Robot Control function by CSTOPI#2.)
- Robot Isolation (J544) option is included by Multi Robot Control (J605) option, but not by Multi UOP Interface (J964) option.
- And in case of Multi Robot Control (J605) function, the motion groups intended by each UOP depend on coupling structure of Isolation switch of Robot Isolation (J544) function. But in case of Multi UOP Interface (J964) function, user can configure the motion groups intended by each UOP freely.

And when both Multi Robot Control (J605) option and Multi UOP Interface (J964) option are ordered, Multi UOP Interface function is used.

(It is possible to be used Multi Robot Control function. Please refer “9.37.6 System variables”)

9.37.2 Peripheral I/O

The following signals are provided in response to the number of UOP set.

INPUT SIGNAL	OUTPUT SIGNAL
<Number of set : 1> *IMSTP *HOLD#1 *SFSPD CSTOP#1 FAULT_RESET START#1 HOME ENBL RSR1/PNS1 RSR2/PNS2 RSR3/PNS3 RSR4/PNS4 RSR5/PNS5 RSR6/PNS6 RSR7/PNS7 RSR8/PNS8 PNSTROBE#1 PROD_START#1 Total 18 points	CMDENBL#1 SYSRDY#1 PROGRUN#1 PAUSED#1 HELD#1 FAULT#1 ATPERCH TPENBL BATALM BUSY ACK1/SNO1 ACK2/SNO2 ACK1/SNO1 ACK1/SNO1 ACK1/SNO1 ACK1/SNO1 ACK1/SNO1 ACK1/SNO1 SNACK RESERVE Total 20 points
<Number of set : 2> *HOLD#2 START#2 PNSTROBE#2 PROD_START#2 CSTOP#2 Total 23 points	CMDENBL#2 SYSRDY#2 HELD#2 FAULT#2 PROGRUN#2 PAUSED#2 Total 26 points
<Number of set : 3> RSR9 RSR10 RSR11 RSR12 *HOLD#3 START#3 PNSTROBE#3 PROD_START#3 CSTOP#3 Total 32 points	ACK9 ACK10 ACK11 ACK12 CMDENBL#3 SYSRDY#3 HELD#3 FAULT#3 PROGRUN#3 PAUSED#3 Total 36 points

<Number of set : 4> RSR13 RSR14 RSR15 RSR16 *HOLD#4 START#4 PNSTROBE#4 PROD_START#4 CSTOP#4 Total 41 points	ACK13 ACK14 ACK15 ACK16 CMDENBL#4 SYSRDY#4 HELD#4 FAULT#4 PROGRUN#4 PAUSED#4 Total 46 points
<Number of set : 5> RSR17 RSR18 RSR19 RSR20 *HOLD#5 START#5 PNSTROBE#5 PROD_START#5 CSTOP#5 Total 50 points	ACK17 ACK18 ACK19 ACK20 CMDENBL#5 SYSRDY#5 HELD#5 FAULT#5 PROGRUN#5 PAUSED#5 Total 56 points

I/O Configuration:

In case that DI/DO are 40 points

Number of set	1	2	3	4	5
Specialized input (UI)	18	23	32	-	-
General-purpose input (DI)	22	17	8	-	-
Specialized output (UO)	20	26	36	-	-
General-purpose output (DO)	20	14	4	-	-

In case that DI/DO are 96 points.

Number of set	1	2	3	4	5
Specialized input (UI)	18	23	32	41	50
General-purpose input (DI)	78	73	64	55	46
Specialized output (UO)	20	26	36	46	56
General-purpose output (DO)	76	70	60	50	40

9.37.2.1 Setting of the number of UOP set

The number of UOP set is 1 by default. Please change \$MULTI_ROBO.\$NUM_PROG from 1 to the number you want to use when you want to change the number of UOP set. You can change the value up to 5. And the number of UOP set is changed after you cycle power.

Then the number of UI/UO signal is changed, but signal assignment isn't changed. If you clear assignment, new assignment in response to new number of UI/UO signal is given automatically (You can clear assignment by selecting "[F5] CLR_ASG" in I/O Link Device screen. Power cycling is needed after that. And all assignments are cleared. If this is inconvenient for you, please change assignment manually).

Here from, 'N' means the number of UOP set, in other words, the value of \$MULTI_ROBO.\$NUM_PROG ('N' is used for explanation after "9.37.2.4 HOLD#1~#N").

9.37.2.2 Modification and Addition of Signal

In Multi UOP Interface Function, the meaning of the following signals are changed.

Input:	HOLD#1:	UI[2] HOLD is changed to HOLD#1.
	CSTOPI#1:	UI[4] CSTOPI is changed to CSTOPI#1.
	START#1:	UI[6] START is changed to START#1.
	PNSTROBE#1:	UI[17] PNSTROBE is changed to PNSTROBE#1.
	PROD_START#1:	UI[18] PROD_START is changed to PROD_START#1.
Output:	CMDENBL#1:	UO[1] CMDENBL is changed to CMDENBL#1.
	SYSRDY#1:	UO[2] SYSRDY is changed to SYSRDY#1.
	PROGRUN#1:	UO[3] PROGRUN is changed to PROGRUN#1.
	PAUSED#1:	UO[4] PAUSED is changed to PAUSED#1.
	HELD#1:	UO[5] HELD is changed to HELD#1.
	FAULT#1:	UO[6] FAULT is changed to FAULT#1.

In Multi UOP Interface Function, the following signals are added.

<Number of set : 2>

	Specialized I/O signal	UI/UO[*]
Input	HOLD#2	UI[19]
	START#2	UI[20]
	PNSTROBE#2	UI[21]
	PROD_START#2	UI[22]
	CSTOPI#2	UI[23]
Output	CMDENBL#2	UO[21]
	SYSRDY#2	UO[22]
	HELD#2	UO[23]
	FAULT#2	UO[24]
	PROGRUN#2	UO[25]
	PAUSED#2	UO[26]

<Number of set : 3>

	Specialized I/O signal	UI/UO[*]
Input	RSR9	UI[24]
	RSR10	UI[25]
	RSR11	UI[26]
	RSR12	UI[27]
	HOLD#3	UI[28]
	START#3	UI[29]
	PNSTROBE#3	UI[30]
	PROD_START#3	UI[31]
Output	CSTOPI#3	UI[32]
	ACK9	UO[27]
	ACK10	UO[28]
	ACK11	UO[29]
	ACK12	UO[30]
	CMDENBL#3	UO[31]
	SYSRDY#3	UO[32]
	HELD#3	UO[33]
	FAULT#3	UO[34]
	PROGRUN#3	UO[35]
	PAUSED#3	UO[36]

<Number of set : 4>

	Specialized I/O signal	UI/UO[*]
Input	RSR13	UI[33]
	RSR14	UI[34]
	RSR15	UI[35]
	RSR16	UI[36]
	HOLD#4	UI[37]
	START#4	UI[38]
	PNSTROBE#4	UI[39]
	PROD_START#4	UI[40]
	CSTOPI#4	UI[41]
Output	ACK13	UO[37]
	ACK14	UO[38]
	ACK15	UO[39]
	ACK16	UO[40]
	CMDENBL#4	UO[41]
	SYSRDY#4	UO[42]
	HELD#4	UO[43]
	FAULT#4	UO[44]
	PROGRUN#4	UO[45]
	PAUSED#4	UO[46]

<Number of set : 5>

	Specialized I/O signal	UI/UO[*]
Input	RSR17	UI[42]
	RSR18	UI[43]
	RSR19	UI[44]
	RSR20	UI[45]
	HOLD#5	UI[46]
	START#5	UI[47]
	PNSTROBE#5	UI[48]
	PROD_START#5	UI[49]
	CSTOPI#5	UI[50]
Output	ACK17	UO[47]
	ACK18	UO[48]
	ACK19	UO[49]
	ACK20	UO[50]
	CMDENBL#5	UO[51]
	SYSRDY#5	UO[52]
	HELD#5	UO[53]
	FAULT#5	UO[54]
	PROGRUN#5	UO[55]
	PAUSED#5	UO[56]

9.37.2.3 PMC

If PMC(FAPT LADER) option (J760) is ordered, added signals are assigned as the following.

PMC Address X	UOP Input Signal
X1002.2-7	UI[19-24]
X1003.0-7	UI[25-32]
X1004.0-7	UI[33-40]
X1030.0-7	UI[41-48]
X1031.0-1	UI[49-50]

PMC Address Y	UOP Output Signal
Y1002.4-7	UO[21-24]
Y1003.0-7	UO[25-32]
Y1004.0-7	UO[33-40]
Y1030.0-7	UO[41-48]
Y1030.1-7	UO[49-56]

PMC Address G	Signal Name
G1002.2	HOLD#2
G1002.3	START#2
G1002.4	PNSTROBE#2
G1002.5	PROD_START#2
G1002.6	CSTOPI#2
G1002.7	RSR9
G1003.0	RSR10
G1003.1	RSR11
G1003.2	RSR12
G1003.3	HOLD#3
G1003.4	START#3
G1003.5	PNSTROBE#3
G1003.6	PROD_START#3
G1003.7	CSTOPI#3
G1004.0	RSR13
G1004.1	RSR14

PMC Address F	Signal Name
F1002.4	CMDENBL#2
F1002.5	SYSRDY#2
F1002.6	HELD#2
F1002.7	FAULT#2
F1003.0	PROGRUN#2
F1003.1	PAUSED#2
F1003.2	ACK9
F1003.3	ACK10
F1003.4	ACK11
F1003.5	ACK12
F1003.6	CMDENBL#3
F1003.7	SYSRDY#3
F1004.0	HELD#3
F1004.1	FAULT#3
F1004.2	PROGRUN#3
F1004.3	PAUSED#3

PMC Address G	Signal Name
G1004.2	RSR15
G1004.3	RSR16
G1004.4	HOLD#4
G1004.5	START#4
G1004.6	PNSTROBE#4
G1004.7	PROD_START#4
G1005.0	CSTOPI#4
G1005.1	RSR17
G1005.2	RSR18
G1005.3	RSR19
G1005.4	RSR20
G1005.5	HOLD#5
G1005.6	START#5
G1005.7	PNSTROBE#5
G1006.0	PROD_START#5
G1006.1	CSTOPI#5

PMC Address F	Signal Name
F1004.4	ACK13
F1004.5	ACK14
F1004.6	ACK15
F1004.7	ACK16
F1005.0	CMDENBL#4
F1005.1	SYSRDY#4
F1005.2	HELD#4
F1005.3	FAULT#4
F1005.4	PROGRUN#4
F1005.5	PAUSED#4
F1005.6	ACK17
F1005.7	ACK18
F1006.0	ACK19
F1006.1	ACK20
F1006.2	CMDENBL#5
F1006.3	SYSRDY#5
F1006.4	HELD#5
F1006.5	FAULT#5
F1006.6	PROGRUN#5
F1006.7	PAUSED#5

9.37.2.4 HOLD#1~#N

Here from, the word “Program#1” appears. “Program#1” means the program selected by UOP set 1. “Program#2” ~ “Program#N” are same as “Program#1”. Please refer “97.3 Select program”.

Multi UOP Interface has one HOLD signal per UOP set (#1 ~ #N). The function of HOLD signal is changed by setting system variable “\$MULTI_ROBO.\$HOLD_TYPE.

- In case of \$MULTI_ROBO.\$HOLD_TYPE = 0 (Default)
If whichever of HOLD signals (HOLD#1~ #N) is turned OFF, all the executing programs are paused.
- In case of \$MULTI_ROBO.\$HOLD_TYPE = 1
If HOLD#X is turned OFF, “Program#X” is paused (X = 1 ~ N).
The running program which is not selected as “Program#X”, can NOT be held. Please turn off ENBL to pause the program.
(However, if “Program#X” is not selected, when HOLD#X is OFF, all the programs in execution are paused.)

But, when a program is tried to execute, regardless of \$MULTI_ROBO.\$HOLD_TYPE value, if any HOLD signal is OFF, it is impossible to execute program. Then if program execution is needed, please turn ON all HOLD signals.

9.37.2.5 CSTOPI#1~#N

Multi UOP Interface has one CSTOPI signal per UOP set (#1 ~ #N). The function of CSTOPI signal is changed by setting “CSTOPI for ABORT” or “Abort all programs by CSTOPI” in System Config screen.

- When “Abort all programs by CSTOPI” is FALSE (Default)
If CSOTPI#X is turned ON, “Program#X” is aborted (X = 1 ~ N).
It also release (clear) programs in UOP set X queue from the wait state by RSR.

Please refer “9.37.2.7 RSR”.

- When “Abort all programs by CSTOPI” is TRUE
If whichever of CSTOPI signals (HOLD#1~ #N) is turned ON, all the executing programs are aborted.
It also release (clear) programs in all UOP set queue from the wait state by RSR.

TRUE for “Abort all programs by CSTOPI” is recommended for Multi UOP Interface function.

The way of terminating program is changed by setting “CSTOPI for ABORT” in System Config screen.

- When “CSTOPI for ABORT” is FALSE (Default)
If CSOTPI signal is turned ON, this signal aborts the program in execution as soon as execution of the program completes.
- When “CSTOPI for ABORT” is TRUE
If CSOTPI signal is turned ON, this signal immediately aborts the program in execution.

9.37.2.6 START#1~#N

Note)

In the system with Multi UOP Interface, multiple programs are controlled at the same time, and it is more complex to control program than control single program. So it is recommended that the Multi UOP Interface is in condition that “START for CONTINUE only:” in system config screen is “TRUE”. Only paused program can be started by START signal under this setting. In case of starting the ended program, use RSR or PROD_START signal, please.

Multi UOP Interface has one START signal per UOP set (#1 ~ #N).

“Program#X” is started when START#X is turned off (X = 1 ~ N).

The program is continued to execute from paused line in case that START signal is input with paused program

When “START for CONTINUE only:” in system config screen is “TRUE”, the program, which is aborted (neither running nor paused), can NOT be started and the message “PROG-023 Task is not paused” is displayed in TP screen in case that START signal is inputted with the ended program (neither running nor paused).

9.37.2.7 RSR

Multi UOP Interface has four RSR signal per UOP set (#1 ~ #N).
(However, if the number of UOP set is 1, eight RSR is usable.)

Number of set	RSR1~4	RSR5~8	RSR9~12	RSR13~16	RSR17~20
1	#1		-	-	-
2	#1	#2	-	-	-
3	#1	#2	#3	-	-
4	#1	#2	#3	#4	-
5	#1	#2	#3	#4	#5

And each UOP set has RSR queue, so the number of RSR queue is N.
Please refer “9.37.4.1 Robot Service Request (RSR)”.

9.37.2.8 PNSTROBE#1~#N,PROD_START#1~#N

Multi UOP Interface has one PNSTROBE and one PROD_START signal per UOP set (#1 ~ #N).
Please refer “9.37.4.2 Program Number Select (PNS)”.

9.37.2.9 CMDENBL#1~#N

Multi UOP Interface has one CMDENBL signal per UOP set (#1 ~ #N).

CMDENBL#1 is ON if all the following conditions are satisfied.

- Remote condition is satisfied.
- No alarm. (Every FAULT#X are OFF. (X = 1 ~ N))
- No single step.
- If Isolation switch is used, motion groups that is intended by UOP set 1 are connected.
 (“Remote condition” and “motion groups that is intended by UOP set 1” is mentioned after)

In case of CMDENBLE#2 ~ #N, only last condition is different.

- If Isolation switch is used, motion groups that is intended by UOP set 2 (~ N) are connected.

1. Remote Condition:

- The Teach Pendant enable switch is set OFF.
- The remote switch on the Operator Panel is set to remote position.
- The *SFSPD input is ON.
- The ENBL input is ON.
- The system variable \$RMT_MASTER is 0 (UOP).

2. Motion groups that is intended by UOP set

Motion groups that is intended by UOP set is changed by setting system variable
“\$MULTI_ROBO.\$MULTI_PROGC”.

Please refer “9.37.3.1 Setting of motion groups intended by each UOP set”.

9.37.2.10 SYSRDY#1~#N

Multi UOP Interface has one SYSRDY signal per UOP set (#1 ~ #N).

SYSRDY#1 is ON if all the following conditions are satisfied.

- All the axes are ready. (If system has Isolation switch, all the axes except for isolated are ready.)
- If Isolation switch is used, motion groups that is intended by UOP set 1 are connected.

In case of SYSRDY#2 ~ #N, only last condition is different.

- If Isolation switch is used, motion groups that is intended by UOP set 2 (~ N) are connected.

9.37.2.11 PROGRUN#1~#N,PAUSED#1~#N

Multi UOP Interface has one PROGRUN and one PAUSED signal per UOP set (#1 ~ #N).
PROGRUN and PAUSED signals show the program execution status as the following.

- When program is running:
PROGRUN#X (X = 1 ~ N)
 - In case that “Program#X” is in execution.

In case that the program, which is not “Program#X”, is in execution, every PROGRUN#X is output.

- When program is paused:
PAUSED#X (X = 1 ~ N)
 - In case that “Program#X” is paused.

In case that the program, which is not “Program#X”, is paused, every PAUSED#X is output.

9.37.2.12 HELD#1~#N

Multi UOP Interface has one HELD signal per UOP set (#1 ~ #N).

HELD#X is ON if one of the following conditions is satisfied. (X = 1 ~ N)

- HOLD#X is OFF.
- The HOLD switch on the Teach Pendant is pushed.
- The HOLD switch on the Operator Panel is pushed.

9.37.2.13 FAULT#1~#N

Multi UOP Interface has one FAULT signal per UOP set (#1 ~ #N).

FAULT#X turns ON when the alarm occurs by “Program#X” execution or other than program execution. (X = 1 ~ N)

When system is in alarm status, at least one of FAULT#X must be ON.

9.37.2.14 ATPERCH

The specified signal ATPERCH is set to ON when the robot of motion group 1 is at the reference position 1. Please assign reference position output except motion group 1.

9.37.3 Select program

In Multit UOP Interface function, it is possible for each UOP set to select program individually. One UOP set is given one program # for selected program.

Here from, “Program#1” means the program selected by UOP set 1.

“Program#1” is run by UOP START#1, PROD_STAR#1.

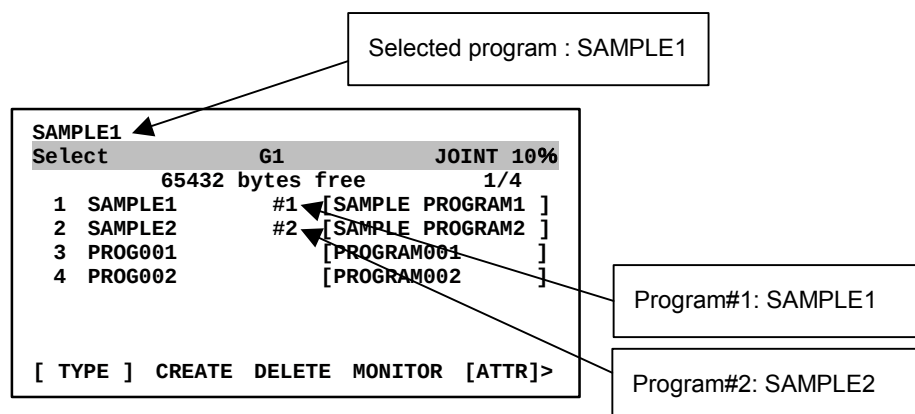
On program list screen, “#1” is displayed to the right of program name that is used as “Program #1”.

“Program#2” ~ “Program#N” are same as “Program#1”.

On the contrary, a program that is displayed on program edit screen is referred as “selected program”.

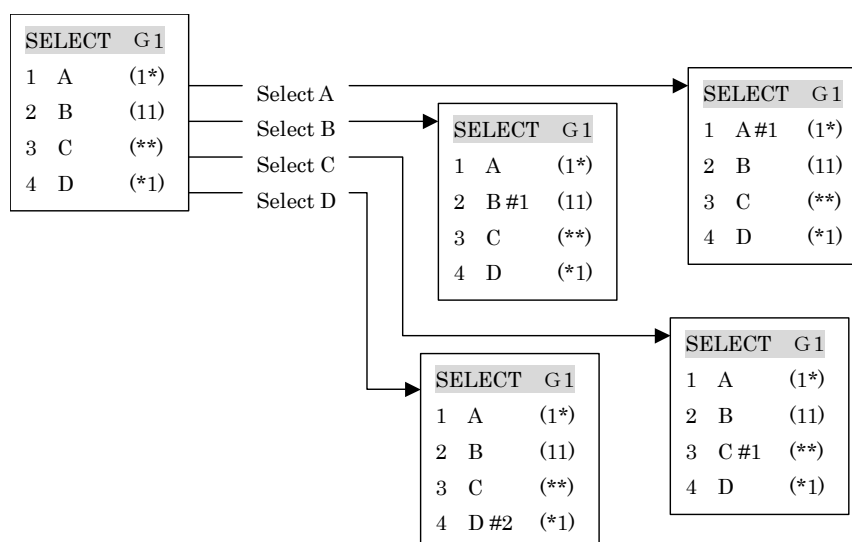
FWD/BWD keys on teach pendant or start button on Operator Panel run “selected program”.

“Selected program” is displayed on left side of second line of TP screen.



If you set cursor to a program and presses ENTER, “selected program” is changed to the program. Then, the program changes to “Program#X” automatically (What # is set depend on motion group of the program and what motion group is intended by each UOP sets).

For example, when the number of UOP sets is 2, and UOP set 1 intends Motion group 1 and UOP set 2 intends Motion group 2, the behavior is as the following figure.



- (1*) The program has only group 1.
- (*1) The program has only group 2.
- (11) The program has group 1 and 2.

9.37.3.1 Setting of motion groups intended by each UOP set

What UOP set intends what motion group, in other words, what # is set to a program is changed by setting system variable “\$MULTI_ROBO.\$MULTI_PROGC.

\$MULTI_PROGC have 5 alignment, and each one corresponds one UOP set. If the number of UOP set is smaller than 5, \$MULTI_PROGC alignments are available as many as the number of UOP set, and the other alignments is ineffective.

```

UOP set 1: $MULTI_PROGC[1]
UOP set 2: $MULTI_PROGC[2]
UOP set 3: $MULTI_PROGC[3]
UOP set 4: $MULTI_PROGC[4]
UOP set 5: $MULTI_PROGC[5]

```

The each bit of \$MULTI_PROGC value is equivalent to each motion group.

Bit31	• • •	Bit1	Bit0
Motion Group32	• • •	Motion Group2	Motion Group1

For example, the default values of \$MULTI_PROGC are set as the following, it is assumed that one UOP set is equivalent to one motion group.

Example1 (Default)

```

UOP set 1: $MULTI_PROGC[1] = 1
UOP set 2: $MULTI_PROGC[2] = 2
UOP set 3: $MULTI_PROGC[3] = 4
UOP set 4: $MULTI_PROGC[4] = 8
UOP set 5: $MULTI_PROGC[5] = 16

```

In case that PROGRAM1 is equivalent to motion group 1,4,
PROGRAM2 is equivalent to motion group 2,5,
and PROGRAM3 is equivalent to motion group 3,
\$MULTI_PROGC value is the following.

Example 2

```

UOP set 1: $MULTI_PROGC[1] = 9
UOP set 2: $MULTI_PROGC[2] = 18
UOP set 3: $MULTI_PROGC[3] = 4
UOP set 4: $MULTI_PROGC[4] = 0
UOP set 5: $MULTI_PROGC[5] = 0

```

Every UOP set can select a program having no motion group. However, when a program having no motion group is selected by TP, if \$MULTI_PROGC[#x] is 0, the program become “Program#X” by priority.

Detailed explanation is mentioned in “9.37.6 System variables”. Please refer.

9.37.3.2 Select program at running or paused

When “Program#1” is paused, if you try to select another program as “Program#1”:

- When program is selected by UOP, you can not select the program, and the message “TPIF-013 Other program is running” is displayed.

- When program is selected by Teach Pendant and Teach Pendant Enable switch is OFF, you can not select the program, and the message “TPIF-013 Other program is running” is displayed.
- When program is selected by Teach Pendant and Teach Pendant Enable switch is ON, the original “Program#1” is aborted, and the selected program becomes “Program#1”.
(This is the case that “Multi Program Selection” is disabled. If “Multi Program Selection” is enabled, “Program#1” doesn’t change.)

When “Program#1” is running, if you select another program as “Program#1”, you can not select the program regardless of selection way, and the message “TPIF-013 Other program is running” is displayed.

“Program#2” ~ “Program#N” are same as “Program#1”.

9.37.3.3 Program edit screen display when operating motion group is changed

When user teach motions to each motion group, user needs to operate to select motion group that can be operated by teach pendant.

It is possible to select motion group to select “CHANGE GROUP” in function menu.

Currently selected motion group is displayed on teach pendant like “G1”, “G2”.

By changing motion group, “selected program” is also changed according to selected motion group. Changed “selected program” is displayed on program edit screen.

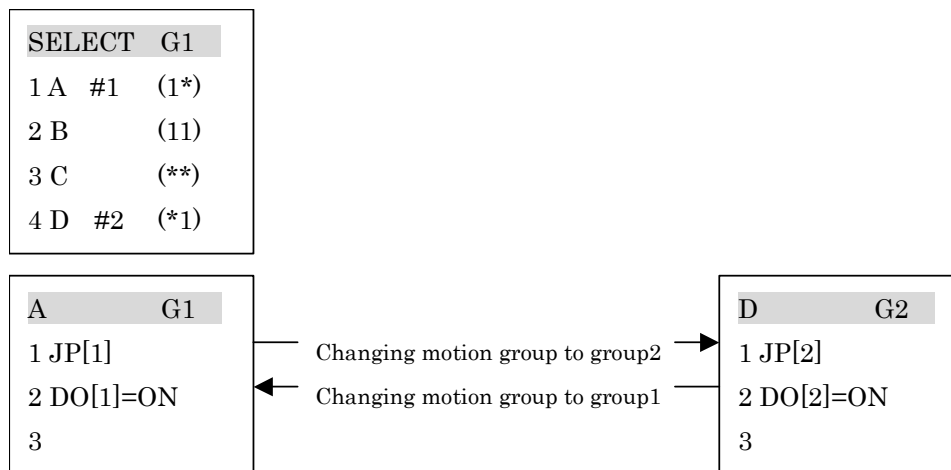
For example, UOP set 1 intends Motion group 1, and UOP set 2 intends Motion group 2.

Suppose “Program #1” has only group1 and “Program #2” has only group2.

By Selecting group1, “Program #1”

By Selecting group2, “Program #2”

automatically becomes “selected program” and the program is displayed on program edit screen.



If “Selected program” has both group1 and group2 or “selected program” has no motion group, change of motion group doesn’t change “selected program”.

To change “selected program” in these cases, display program list screen, set cursor to a program that you want to display and press ENTER.

9.37.4 Run program

9.37.4.1 Robot Service Request (RSR)

If the number of UOP set is 1, when RSR1-8 are turned on, the program corresponded to the RSR number is started if the program is aborted.

If another program is running or paused, the request is joined to the queue. And the program is started when the running program is completed.

Multi UOP Interface has four RSR signal per UOP set (#1 ~ #N) if the number of UOP set is 2 and over. RSR1~4 are for UOP set 1. RSR5~8, RSR9~12, ... are same as RSR1~4 (Please refer “9.37.2.7 RSR”). And each UOP set has RSR queue, so the number of RSR queue is N.

By this function, N programs can run at the same time.

The program started by RSR1-4 becomes “Program#1”.

In the following situation, program is not started.

- The program started by RSR1-4 does not have motion group intended by UOP set 1 and it has motion group intended by other UOP set.

In this case, message “TPIF-44 Program is unsuitable for robot” is displayed.

RSR5~8, RSR9~12, ... are same as RSR1~4.

9.37.4.2 Program Number Select (PNS)

Multi UOP Interface has one PNSTROBE and one PROD_START signal per UOP set (#1 ~ #N).

The program selected by PNSTROBE#X becomes “Program#X”. This program can be started by PROD_START#X and it is resumed from pause by START#X (X = 1 ~ N).

In the following situation, program is not selected.

- The program selected by PNSTROBE#X does not have motion group intended by UOP set X and it has motion group intended by other UOP set.

In this case, message “TPIF-44 Program is unsuitable for robot” is displayed.

9.37.5 Stop program

9.37.5.1 Pause program by UOP

When whichever of HOLD signals (HOLD#1 ~ #N) is turned OFF, running program is paused. Please read “9.37.2.4 HOLD#1~#N” for detail.

9.37.5.2 Abort program by UOP

If “CSTOPI for ABORT” in System Config is TRUE, programs are aborted when CSTOPI turns ON. The default setting of ‘CSTOPI for ABORT’ is FALSE. Please read “9.37.2.5 CSTOPI#1~#N” for detail.

9.37.5.3 Other

When hold button on Teach Pendant or Operator Panel, all running programs are paused.

9.37.6 System variables

Name	\$MULTI_ROBO.\$MLT_ENABLE
Default	TRUE
Type	BOOLEAN
Protection	Not protected
Range	TRUE/FALSE
Meaning	Enable/Disable multi robot control function. TRUE: Multi robot control function is enabled. FALSE: Multi robot control function is disabled.

Name	\$MULTI_ROBO.\$HOLD_TYPE
Default	0
Type	INTEGER
Protection	Not protected
Range	0/1
Meaning	Select the function of HOLD#1 and HOLD#2. 0: All programs are paused when HOLD#1 or HOLD#2 turns OFF. 1: "Program#1" is paused when HOLD#1 turns OFF. "Program#2" is paused when HOLD#2 turns OFF.

Name	\$MULTI_ROBO.\$NUM_PROG
Default	1
Type	INTEGER
Protection	Not protected
Range	1 ~ 5
Meaning	The number of UOP set to use.

Name	\$MULTI_ROBO.\$MULTI_PROGC[5]
Default	[1]1, [2]2, [3]4, [4]8, [5]16
Type	INTEGER
Protection	Not protected
Range	0 ~ 0xFFFF
Meaning	Bit mask for intended motion group of each UOP set. In the default, UOP set 1 intends Motion group 1, UOP set 2 intends Motion group 2, and so on.

[Supplement] Basis of selection for program #

We can select program by UOP or TP, and program number is set on the basis of \$MULTI_PROGC. Program number setting rule is the following:

< Common rule for UOP and TP >

- Rule 1) If motion group of \$MULTI_PROGC[#n] and motion group of program match at least one group, the program can be set #n.
- Rule 2) No motion group program can be set ALL program number.
- Rule 3) If motion group of every \$MULTI_PROGC[#n] and motion group of program match none, the program is treated equally as no motion group program. (So conform to Rule 2.)

Example \$MULTI_PROGC[1-5] = 1, 2, 3, 4, 0

	Motion group of program A~G G1, 2, 3, 4, 5, 6, 7, 8	Program number				
		#1	#2	#3	#4	#5
PROGRAM A	(1, *, *, *, *, *, *, *)	#1		#3		
PROGRAM B	(*, 1, *, *, *, *, *, *)		#2	#3		
PROGRAM C	(1, 1, *, *, *, *, *, *)	#1	#2	#3		
PROGRAM D	(1, *, 1, *, *, *, *, *)	#1		#3	#4	
PROGRAM E	(*, *, *, *, *, *, *, *)	#1	#2	#3	#4	#5
PROGRAM F	(*, 1, *, 1, *, *, *, *)		#2	#3		
PROGRAM G	(*, *, *, 1, *, *, *, *)	#1	#2	#3	#4	#5

The above figure illustrates PROGRAM C can be set #1, #2, or #3, and conversely, #2 can select PROGRAM B, C, E, F or G, for instance.

\$MULTI_PROGC[1] = 1 >> match at least one group to PROGRAM A, C, D
 \$MULTI_PROGC[2] = 2 >> match at least one group to PROGRAM B, C, F
 \$MULTI_PROGC[3] = 3 >> match at least one group to PROGRAM A, B, C, D, F
 \$MULTI_PROGC[4] = 4 >> match at least one group to PROGRAM D
 \$MULTI_PROGC[5] = 0 >> match at least one group to none

\$MULTI_PROGC[1-5] >> can select no motion group PROGRAM E, G(along Rule 3)

< Rule for TP only >

Rule 4) If selected program is already set program number, program number don't change.

Rule 5) If selected program is not set program number yet, the program is set program number along the following priority. And if some program number have same priority, small number comes first.

(In the following Priorities from 1 to 4, #X is the number that "Program #X" is aborted or is not selected.)

[Priority 1] Motion group of \$MULTI_PROGC[#x] is equal to motion group of program.

(If PROGRAM C is selected by TP, set #3 not #1 or #2.

If PROGRAM E is selected by TP, set #5 not #1, #2, #3 or #4.)

[Priority 2] Motion group of \$MULTI_PROGC[#x] inhere motion group of program.

(If PROGRAM B is selected by TP, set #2 not #3.)

[Priority 3] Motion group of \$MULTI_PROGC[#x] and motion group of program match at least one group.

[Priority 4] If program isn't along the above priority from 1 to 3, the program is set to smallest number X that "Program #X" is aborted or is not selected.

(A program along Rule 2 or 3 is treated here unless along priority 1.)

[Priority 5] #X is the number that "Program#X" is paused. On that basis, the smallest number that complies with priority 3.

(Priority 5 is applied when TP is enabled and "Multi Program Selection" is disabled. Then paused program is aborted.)

When none of the above Priorities is corresponded, it means that all "Program#X" that #X complies with Priorities from 1 to 4 are in execution. Then warning message is displayed and # isn't set to selected program.

Name	\$MULTI_ROBO.\$STRAD_MODE
Default	FALSE
Type	BOOLEAN
Protection	Not protected
Range	TRUE/FALSE
Meaning	Compatible flag for Multi Robot Control Function. TRUE: Compatible mode - There is single CSTOPI signal. - \$MULTI_ROBO.\$NUM_PROG is 2. - \$ROBOT_ISOLC plays role of \$MULTI_ROBO.\$MULTI_PROGC. FALSE: Normal Multi UOP Interface mode

9.38 Error Code Output function

9.38.1 Introduction



CAUTION

This is optional function (J527), and the expanded I/O board is necessary.

This function outputs the contents of the alarm that occurred in the robot as a digital output signal code with strobe.

9.38.2 Specification

9.38.2.1 Types of Alarms

All alarms except ones whose severity WARN are the targets of external output.

9.38.2.2 Input and Output signals

Expanded I/O board is necessary to use this function.

33 points of DO are used as output signals of external error code, and 1 point of DI is used as input signal.

\$ER_OUT_PUT.\$OUT_NUM : **Number of output signal**

\$ER_OUT_PUT.\$IN_NUM : **Number of input signal**

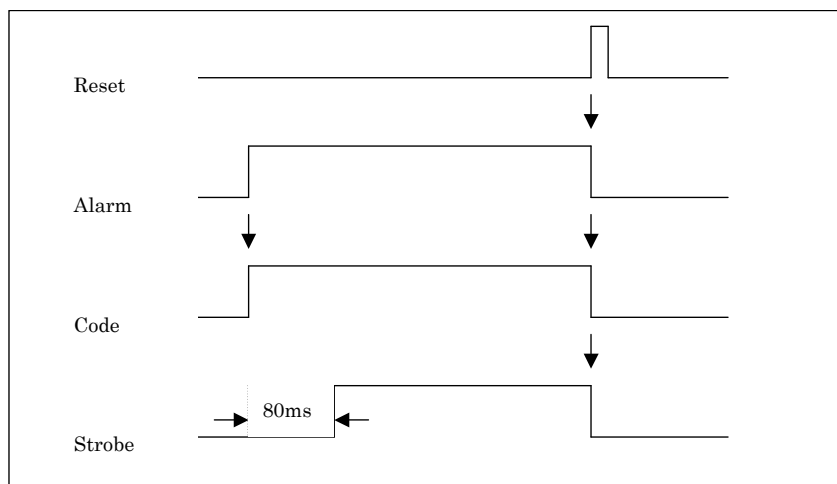
- Output signals
\$ER_OUT_PUT.\$OUT_NUM determines which DO is used.
33 points of DO from setup number are used.
When setup number is 0, this function is disabled.
- Types of output signal
Output of both code signals and strobe are kept on until reset signal is inputted.

Meanings of DO (in case of \$ER_OUT_PUT.\$OUT_NUM=1)

DO	Types of alarms
1~32	Output alarms as binary (32 bit).
33	Used as strobe.

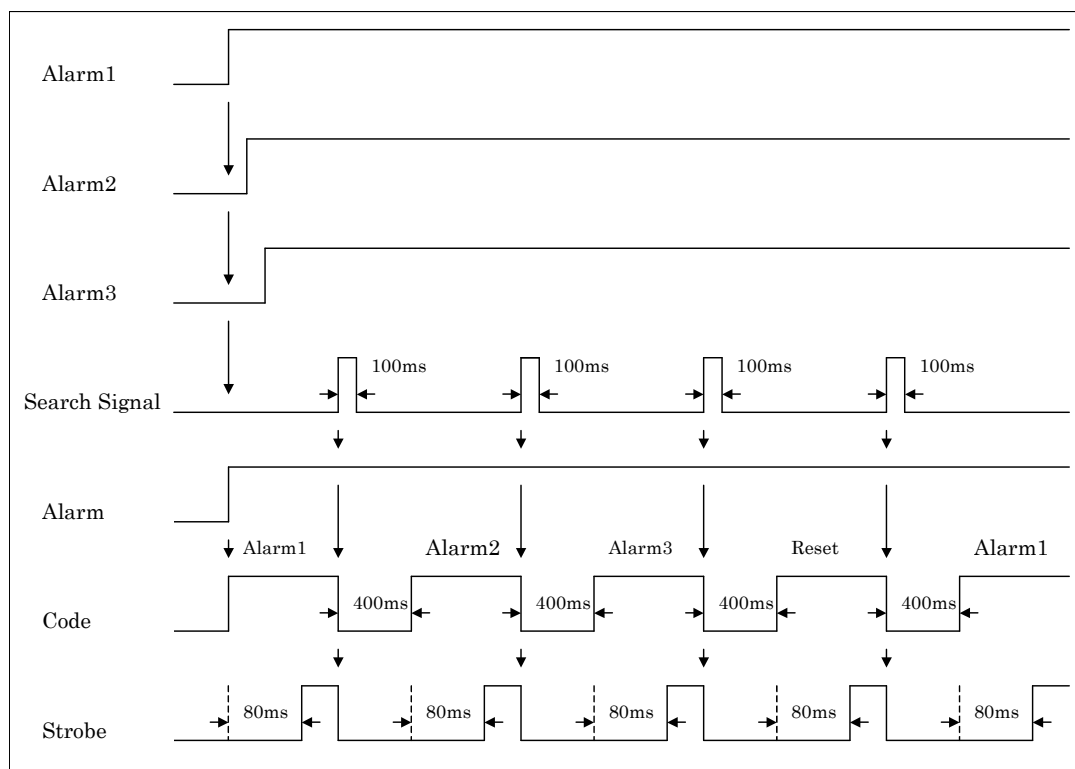
- Input signal (1bit)
When there are multiple errors, input signal is used as search signal.
\$ER_OUT_PUT.\$IN_NUM determines which DI is used.

- Time chart of signals
When an alarm occurs, the time chart is as follows.



Time chart of an alarm

When multiple alarms occur at a time, the initial alarm is outputted at first. Then, the next alarm is output every time search signal is input. These are cyclically output and after that reset signal with strobe is output. Up to 100 alarms can be stored.



Time chart of multiple alarms

9.38.3 Meaning of Alarm Code

In case of \$ER_OUT_PUT.\$OUT_NUM = 1

DO	Types of alarms
1~16	Output an alarm number as binary (16bit).

17~24	Output an alarm ID as binary (8bit).
25~32	Output an alarm Severity as binary (8bit).
33	Used as strobe.

9.38.3.1 Severity of Alarm

DO Severity	25	26	27	28	29	30	31	32
NONE	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON
WARN	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF

Errors that have the severity above will not be outputted.

DO Severity	25	26	27	28	29	30	31	32
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

DO[25] and DO[26] means the severity of running a program.

DO [25]	DO [26]	
OFF	OFF	The program continues running.
OFF	ON	The program will be paused.
ON	ON	The program will be aborted.

DO[27] and DO[28] means the severity of motion.

DO [27]	DO [28]	
OFF	OFF	The motion continues running.
ON	OFF	The motion will be stopped.
OFF	ON	The motion will be canceled.

DO[29] means the severity of a servomotor.

OFF	The servomotor continues running.
ON	The servomotor will be powered down.

DO[30] means the severity of the target tasks.

OFF	Effective only for the task.
ON	Effective for all the tasks.

DO[31] means the request for recovery.

OFF	No need to power down.
ON	Cold start is necessary.

DO[32] means the request for the display on TP.

OFF	No need to display.

ON	Need to display.

9.38.3.2 Alarm ID

Alarm ID will be outputted as the number in the following.

Typical alarm ID:

Number	Alarm ID
0	OS
3	PROG
7	MEMO
9	TPIF
11	SRVO
12	INTP
15	MOTN
19	JOG
20	APPL
23	SPOT
24	SYST
26	PALT
53	ARC
57	MACR
58	SENS
59	COMP

Other alarm ID:

Number	Alarm ID	Number	Alarm ID
1	SRIO	36	TKSP
2	FILE	37	COPT
4	COND	38	APSH
5	ELOG	42	CMND
6	MCTL	43	RPM
8	GUID	44	LNTK
10	FLPY	45	WEAV
13	PRIO	46	TCP
14	TPAX	47	TAST
16	VAR	48	MUPS
17	ROUT	49	MIGE
18	WWDW	50	LSR
21	LANG	51	SEAL
25	SCIO	52	PANE
27	UAPL	54	TRAK
33	DICT	55	CALB
34	KCLI	56	SP
35	TRAN	60	THSR

9.38.3.3 Alarm Number

Alarm number will be directly output as binary data.

Example

In case of “SRVO 002 (severity SERVO)”
Alarm number is 2.

Alarm ID “SRVO” is 11.

The severity is SERVO.

DO [1]	OFF	Alarm number 2	DO [17]	ON	Alarm ID 11
DO [2]	ON		DO [18]	ON	
DO [3]	OFF		DO [19]	OFF	
DO [4]	OFF		DO [20]	ON	
DO [5]	OFF		DO [21]	OFF	
DO [6]	OFF		DO [22]	OFF	
DO [7]	OFF		DO [23]	OFF	
DO [8]	OFF		DO [24]	OFF	
DO [9]	OFF		DO [25]	OFF	Alarm Severity SERVO
DO [10]	OFF		DO [26]	OFF	
DO [11]	OFF		DO [27]	ON	
DO [12]	OFF		DO [28]	OFF	
DO [13]	OFF		DO [29]	ON	
DO [14]	OFF		DO [30]	ON	
DO [15]	OFF		DO [31]	OFF	
DO [16]	OFF		DO [32]	OFF	

9.39 DATA MONITOR

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 option (J675) is needed to use this function.

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 teach pendant screens and add two teach pendant program instructions. Specifically,

- Enable or disable any specific features in the Data Monitor Utility screen
- Select items to monitor (such as arc current feedback) with one of the Data Monitor schedules
- Add Sample Start[schedule number] and Sample End instructions to your TP program to control when monitoring occurs

You can monitor up to five 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 Fig.9.39 (a). The data is tab delimited for importing into a spreadsheet application.

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	1.192	TEST	3	20.000	200.000
7	348	1.392	TEST	4	0.000	0.000

Fig. 9.39 (a) Report Example

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] 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)
- Axis position.

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

Data monitoring can check each sampled item against upper and lower limits. If the average of the item samples is out of limits for a specified time period (T_{min} in Figure 9.39 (b)) a warning or a pause alarm will occur, if enabled. You must specify a nominal value, a warning limit, a pause limit, and a time duration for each monitored data item.

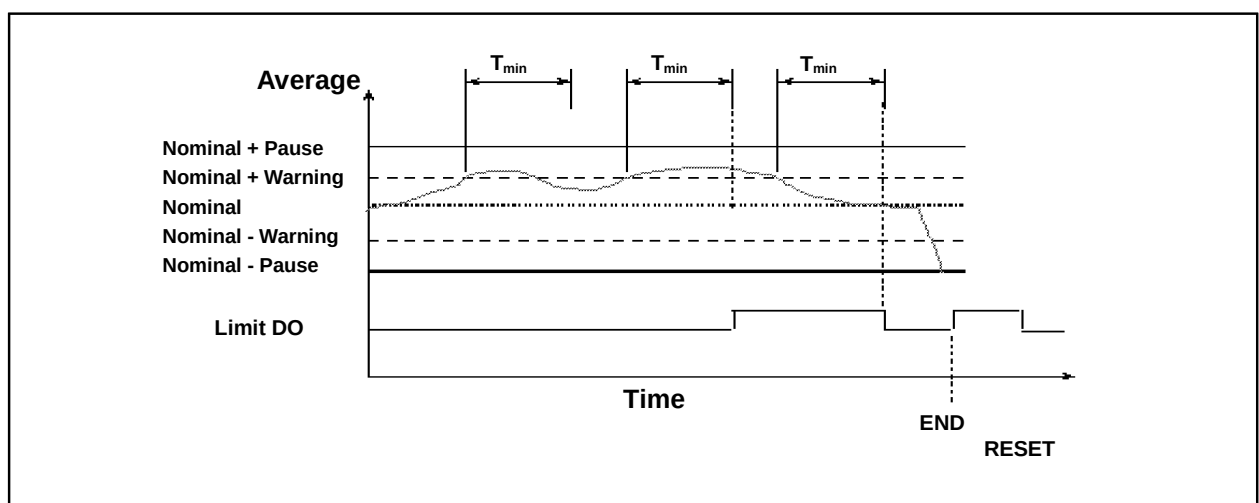


Fig. 9.39 (b) Process Limits

If a warning limit is crossed for the specified time, a WARN severity error is posted and the limit digital output is turned ON. If the average of the item samples returns to within the WARN limit for the specified time the digital output is turned OFF.

If a pause limit is crossed for the specified time a PAUSE severity error is posted, the limit digital output is set to ON, and the program is paused.

When the program ends, if a WARN or PAUSE limit error occurred during execution, the limit digital output is turned ON. It is turned OFF by a system RESET.

9.39.1 Data Monitor Setup

You must set up Data Monitor before you can use it. Table 9.39.1 lists and describes the items found on the Data Monitor Setup screen.

Table 9.39.1 Data Monitor Setup Screen Menu Items

ITEM	DESCRIPTION
Data Monitor Operation Default: Enabled	This item enables and disables operation of the Data Monitor function. NOTE Data monitoring is also disabled during step mode.
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 begin monitored stops.
Warning Limits Default: Disabled	This item turns Warning Limits ON and OFF.
Pause Limits Default: Disabled	This item turns Pause Limits ON and OFF.
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 Default: 10 Min: 1 Max: 99	This item specifies the size of the sample buffer.
Record Buffer Size Default: 10 Min: 1 Max: 99	This item specifies the size of the record buffer.
Setup Default: Disabled	This item enables or disables printing of data monitor setup information in the report header. See Fig. 9.39.1
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.
Triggers Default: Disabled	This item enables or disables printing of Trigger information in the report header.
Program Name Default: Enabled	This item enables or disables printing the program name column in the Data Monitor report. See Fig. 9.39.1.
Line Number Default: Enabled	This item enables or disables printing the line number column in the Data Monitor report. See Fig. 9.39.1.
Date Default: Disabled	This item enables or disables printing the date and time of day column in the Data Monitor report. See Fig. 9.39.1.
Tick + time Default: Disabled	This item enables or disables printing the Tick and Time column in the Data Monitor report. See Fig. 9.39.1.
Event Default: Disabled	This item enables or disables printing the event column in the Data Monitor report. See Fig. 9.39.1. NOTE At this time, there is only one event defined. An event value of 1

ITEM	DESCRIPTION
	indicates the data was recorded as a result of the recording frequency.
Distance Default: Enabled	This item enables or disables printing the distance column in the Data Monitor report. See Fig. 9.39.1.

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

Fig. 9.39.1 Sample Report

Procedure 9-28 Setting up data monitor

Condition

Data Monitor is installed on your controller.

Step

- 1 Press **MENUS**.
- 2 Select **UTILITIES**.
- 3 Press **F1**, **[TYPE]**.
- 4 Select **Data Monitor**. You will see a screen similar to the following.

UTILITIES DMON SET G1		JOINT 10 %
		1/20
1 Data Monitor Operation:	ENABLED	
2 Recording:	ENABLED	
3 Filing:	ENABLED	
4 Pause on File Errors:	DISABLED	
5 Warning limits:	DISABLED	
6 Pause limits:	DISABLED	
7 Limit error output:	DO[0]	
8 Sample buffer size:	10 samples	
9 Record buffer size:	10 samples	
ITEM DESCRIPTION	ITEM NUM	
10 Item 1	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	
[TYPE]	DETAIL ENABLED DISABLED	

- 5 Select each item on the menu and set it as desired.

Edit Data Monitor Items

- 6 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.

ITEM DESCRIPTION	ITEM NUM
10 Item 1	1
[TYPE] DETAIL [CHOICE] HELP	

- 7 Press F4, [CHOICE]. You will see a screen similar to the following

```

1 Item 1          5 Item 5
2 Item 2          6 Item 6
3 Item 3          7 Item 7
4 Item 4          8 -- NEXT --
UTILITIES DMON SET

9 Record buffer size:    10 samples

ITEM DESCRIPTION          ITEM NUM
10 Item 1                  1

REPORT HEADER CONTENTS

[ TYPE ]          DETAIL [CHOICE]  HELP

```

- 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 Item 2] has been selected. You will see a screen similar to the following.

```

UTILITIES DMON SET

Data monitor item number:    2/ 20
1 Item type:                  INTEGER
Item sub type:               **
Port or register number:     ***
Axis number:                  G:* A:*
2 Program name: [              ]
3 Var: [                      ]
4 Des: [                      ] Item 2]
5 Units: [                      ]
6 Slope:                    0.00
7 Intercept:                 0.00

[ TYPE ] ITEM  EXIT  [CHOICE]  HELP

```

You can specify comment about the item in [4 Des] and [5 Unit].
[6 slope] [7 Intercept] is for linear transformation. When Non-zero value is specified as slope,

Data is transformed as follows.

$$\text{Data} = [\text{Slope}] * [\text{raw data}] + [\text{intercept}]$$

When 0 is specified as slope, raw data is used without transformation.

Notice: Transformed data is used for comparison between PAUSE/WARN limit or condition of Start/End trigger

- 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 [CHOICE]. You will see a screen similar to the following.

1 INTEGER	5 AXIS
2 REAL	6
3 I/O	7
4 REGISTER	8
UTILITIES DMON ITM	
Data monitor item number: 2/ 20	
1 Item type:	INTEGER
Item sub type:	**
Port or register number:	***
Axis number:	G:* A:*
2 Program name:	[]
[TYPE] ITEM EXIT [CHOICE] HELP	

- 11 To change the Item type, move the cursor to Item type, and press ENTER.
 Case that I/O is selected:
 Specify [2 Item sub type] and [3 Port or register number].

Example for setting DO[3]

UTILITIES DMON ITM	
1 Item type:	I/O
2 Item sub type:	DO
3 Port or register number:	3
Axis number:	G:* A:*
Program name:	[*****]
Var:	[*****]
4 Des:	[Item 2]
5 Units:	[]
6 Slope:	0.00
7 Intercept:	0.00
[TYPE] ITEM EXIT HELP	

- Case that REGISTER is selected:
 Specify [3 Port or register number].

Example for setting Register[5]

UTILITIES DMON ITM	
1 Item type:	REGISTER
Item sub type:	**
2 Port or register number:	5
Axis number:	G:* A:*
Program name:	[*****]
Var:	[*****]
3 Des:	[Item 2]
4 Units:	[]
5 Slope:	0.00
6 Intercept:	0.00
[TYPE] ITEM EXIT HELP	

- Case that Axis is selected:
 Specify [2 Axis number].
 Specify group number of robot in [G:] and axis number in [A:] .

Caution: About J3 axis, J2/J3 interaction is not considered.

Example for setting J1 axis of group 1 robot

UTILITIES DMON ITM			
1	Item type:	AXIS	
	Item sub type:	**	
	Port or register number:	***	
2	Axis number:	G:1 A:1	
	Program name:	[*****]	
	Var:	[*****]	
3	Des:	[	Item 2]
4	Units:	[	]
5	Slope:	0.00	
6	Intercept:	0.00	
[TYPE] ITEM EXIT HELP			

Case that INTEGER or REAL is selected:

Select Integer or Real depending on data type of system valuable or KAREL valuable.

Specify [2 Program name].

Specify [*system*] for system valuable

or [name of KAREL program] for KAREL valuable.

Specify [3 Var].

Specify valuable name in [3 Var].

Example for setting system valuable \$MCR.\$GENOVERRIDE

UTILITIES DMON ITM			
1	Item type:	INTEGER	
	Item sub type:	**	
	Port or register number:	***	
	Axis number:	G:* A:*	
2	Program name:	[	*SYSTEM*]
3	Var:	[	\$MCR.\$GENOVERRIDE]
4	Des:	[	Item 3]
5	Units:	[	]
6	Slope:	0.00	
7	Intercept:	0.00	
[TYPE] ITEM EXIT HELP			

- 12 When you are done 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.

9.39.2 Data Monitor Schedule

You choose the items you want to monitor in a Data Monitor schedule.

You can also specify:

- Report file naming details
- Sampling frequencies.
- WARN and STOP limits
- 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 9.39.2 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 writing a report. You can choose from FLPY:, PRN:, FR:, MC:, CONS:, or RD:.
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".
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 will be incremented. For example if the file name is SAMPL then successive reports will be named SAMPL001.DT, SAMPL002.DT, and so forth.
File size 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.
Sampling	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 may be greater than or less than the sampling frequency you specify.
Monitoring	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 may be greater than or less than the monitoring frequency you specify.
Recording	This item specifies the recording frequency. -Request - This is the recording frequency you specify. -Actual - This is the actual recording frequency that Data Monitor will use. Since there are only certain frequencies available, it may be greater than or less than the recording frequency you specify.
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 Default: 1 Min: 1 Max: 6	This item specifies how many items are monitored by this particular schedule.
Start item	This item specifies the condition for starting data recording and enable/disable of this condition after Sample Start instruction. If this condition is set to disable, data recording is started just after Sample Start instruction. The condition is set as the comparison between the value of the monitoring item and setting item. For example, if condition is [recording is started after current command (item No.=2) becomes more than 150.], please set to [2 > 150]. NOTE: The user can check the item No. for monitoring by data monitor schedule screen.
Stop item	This item specifies the condition for finishing data recording and enable/disable of this condition after Sample Start instruction. If this condition is set to disable, data recording is not finished till performing Sample End instruction. The method of setting condition is the same as Start item.

Procedure 9-29 Setting Up and Editing a Data Monitor Schedule

Condition

You have installed the Data Monitor schedule on your controller.

Step

- 1 Press **MENUS**.
- 2 Select **UTILITIES**.
- 3 Press **F1**, [**TYPE**].
- 4 Select Data Monitor Schedules. You will see a screen similar to the following.

UTILITIES DMON SCH		JOINT 10 %
Sched: 1/5 [Weld cmd + fbk]		12/17
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
[TYPE] SCHEDULE LIMITS [CHOICE] HELP		

- 5 There are 5 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 after the prompt.
- 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 don't 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 may 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 5 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.

Specify WARN and STOP Limits for Process Signals

- 15 To specify WARN and STOP limits, move the cursor to each item you have specified on lines 11 to 15 and press the LIMITS function key.
- 16 Move the cursor to one of monitored items.
- 17 Press F3, LIMITS. You will see a that contains details on how specifically to monitor the item you selected.

UTILITIES DMON LIM		JOINT 10%
Schedule: 1		[Sample example] 1/4
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:		0.0 seconds
[TYPE]	EXIT	HELP

- 18 Select each item and set it as desired.
- 19 When you finished setting items, press F3, EXIT, to display the previous screen.

Start and Stop Trigger Items

- 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.

9.39.3 Programming

You can use the following teach pendant instructions to start and end data monitoring:

- Sample Start[]
- Sample End

The Start instruction has a schedule number as an input parameter.

NOTE

You cannot start multiple data monitoring sessions at one time. You must end a monitoring session with a Sample End before executing another Sample Start.

Sample Start[1]

- To start data monitoring, include the Sample Start[] instruction in a teach pendant program.

Sample End

- To stop data monitoring, include the Sample End instruction in a teach pendant program.

See Fig. 9.39.3 for an example of how to use these instructions in a teach pendant program.

```

1: Sample Start[1]
2: J P[1] 40% FINE
  : Arc Start[1]
3: L P[2] 20.0cm/min FINE
  : Arc End[1]
4: Sample End
[End]

```

Fig. 9.39.3 Example of Using Sample Start[] and Sample End in a Teach Pendant Program

The Sample Start and Sample End instructions are located in the Data Monitor category of the Teach Pendant Editor INST menu.

9.39.4 Data Monitor Chart

Data monitor chart screen enables you to display the graphs of the monitor reports on *i*Pendant. The graphs of the monitor reports are displayed after reading from other memory (MC etc.). So it is needed to insert other memory to a robot controller before displaying data monitor chart screen.

Procedure 9-30 Data monitor chart

Condition

Data monitor function is ordered.

Robot controller have already detect other memory which have monitor reports.

Step

- 1 Press MENUS.
- 2 Select [1 Utility].
- 3 Press F1[TYPE].
- 4 Select [DMON Chart]. The following screen is displayed.

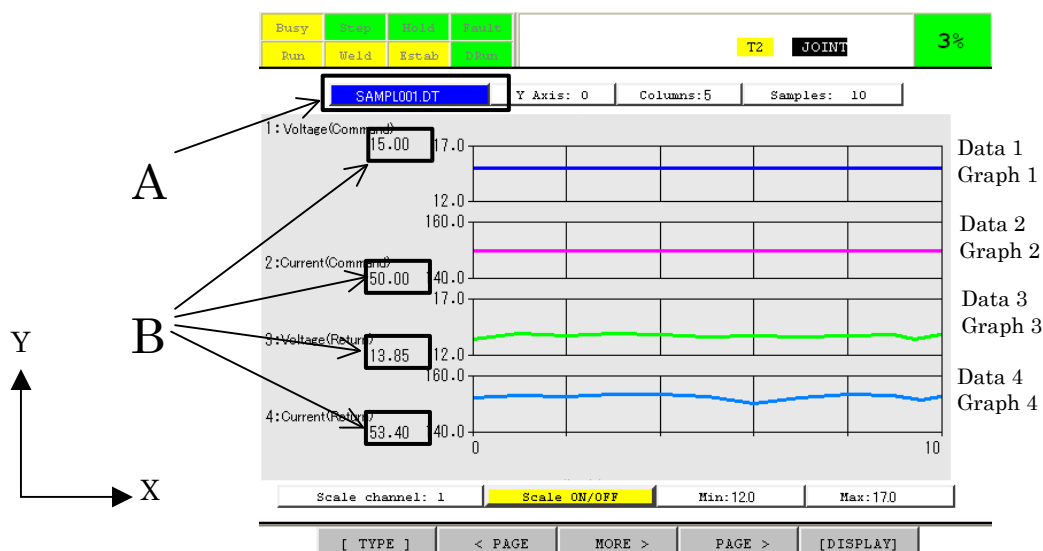


Fig. 9.39.4 Data Monitor Chart screen

- 5 Data monitor chart screen can have up to 8 graphs. Y direction of a graph is reported data and X direction is the reported count.
- 6 The file name of the monitor report is displayed on Fig 9.39.4 A. When the cursor is moved to A and ENTER key is pressed, file select screen is displayed and the user can select the [.DT] file in other memory.
- 7 In Fig 9.39.4 B, newer value of the Y direction of each graph is displayed.
- 8 [Y Axis:] can compare displayed graphs. If 1 is set to [Y Axis:], Data 2-4 are displayed on Graph 1. If 2 is set to [Y Axis:], Data 1,3,4 are displayed on Graph 2.
- 9 [Columns:] can specify the number of the scale of X direction. [Columns:] can be set from 1 to 20.
- 10 [Samples:] can specify the number of displayed data. [Samples:] can be set from 1 to 100. If the user want to the part of the data, please set small number to this item.
- 11 [Scale channel:], [Scale ON/OFF], [Min:] and [Max:] can change the scale of Y direction in each graph. First, enter the graph number which you want to change scale to [Scale channel:]. If 1 is set to [Scale channel:], the user can change the scale of graph 1. Next, set the minimum and maximum value of Y axis to [Min:] and [Max:]. After that, move the cursor to [Scale ON/OFF] and press ENTER key. Y axis scale of the graph which is specified by [Scale channel:] is changed.
- 12 F2[< PAGE] and F4[PAGE >] key can change the scale of X direction of each graph.
- 13 F4[PAGE >] key can move the column of X axis by +1.
- 14 F5[DISPLAY] key can switch display enable/disable of each graph.

9.40 BRAKE CHECK FUNCTION

9.40.1 Overview

Brake check function (A05B-2500-J951) is a function to check that motor does not move when motor torque is applied under the brake is engaged. This function makes user to cause early detection of brake failure. This function also informs the check results to user by warning message or system variables.

9.40.2 Initial setting before use

Enable/Disable switch for brake check function

The following system variable is the switch that enable/disable brake check function.

`$BCCFG.$BCK_ENABLE` TRUE: Enable (default) / FALSE: Disable

Please confirm this variable is TRUE (Enable) before use. The default value of this variable is TRUE (Enable). If this variable is FALSE (Disable), brake check never starts.

Enable/Disable switch for each axis

On this function, user can customize enable/disable brake check for each axis. Please set the following system variable to TRUE (Enable) for the axis where you want to execute brake check.

`$BCK_GRP[group].$CHK_AXIS[axis]` TRUE: Enable / FALSE: Disable

On robot axis, the default value is TRUE (Enable). On except robot axis (aux axis, servo gun, or etc.), the default value is FALSE (Disable).

In addition, if you execute brake check on dual drive axis, please set this variable to TRUE (Enable) for both master and slave axis.

Setting brake torque

This function needs brake torque for each axis in order to determine torque command during brake check.

Please set the following system variable with the brake torque [kgf*cm] for each axis.

`$BCK_GRP[group].$BRK_TORQUE[axis]` brake torque [kgf*cm]

On robot axis, this variable is set automatically, however, on except robot axis (aux axis, servo gun, or etc.), this variable is not set automatically, so please manually set this variable with brake torque if you execute brake check on except robot axis. For specification value of brake torque, please refer *AC SERVO MOTOR DESCRIPTIONS* “9.1 BRAKE SPECIFICATIONS”. Please note that the unit of this variable is [kgf*cm].

In addition, if you execute brake check on dual drive axis (except robot axis), please set this variable as follows. (For robot axis, it is no need to set even if it's dual drive axis.)

Case-1: Either master or slave motor has one brake.

Please set this variable with the brake torque [kgf*cm] for 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 brake respectively.

Please set this variable with the brake torque [kgf*cm] for both master and slave axes. And then the brake torque value of master and slave axis must be same. If these values are different in the Case-2, it is not supported.

9.40.3 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 start brake check by parameter instruction of TP program or manually setting on system variable screen. This variable automatically returns to FALSE when brake check is finished. Please not to move robot till this variable automatically returns to FALSE. In addition, if brake check function is disabled (`$BCCFG.$BCK_ENABLE=FALSE`), this variable immediately returns to FALSE even if this variable is set to TRUE.

Follows is a sample of TP program that execute 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 as specification, so this program waits for robot to stop completely by WAIT instruction at L3. And this program starts brake check at L4 and waits to finish brake check at L5. As just described on this sample program, brake check must be executed under the robot stops completely.



NOTE

In case of brake failure, the motor can move during brake check. So please be sure to execute brake check in sufficiently spaced layout. Total motion command per single check is vary by robot or axis, however, it is up to 200000[pulse]. This is about 0.4[rev] in motor revolution and about 1.4[deg] in degree of the angle assuming the gear ratio as 1/100.

Additionally, in case where the power is shut OFF during brake check for some reason, the following alarm will be posted at next power ON. This alarm indicates 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 was normally finished

In this case, the results of brake check will not be generated, so please execute brake check again at next power ON. If next brake check would be normally finished, this variable will return to FALSE automatically.

9.40.4 Results of brake check

`$BCCFG.$BCK_START` automatically returns to FALSE when brake check finishes. And then the results are 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 on the following system variable at the same time. The value of this variable is kept till next brake check starts.

`$BCK_GRP[group].$CHK_RESULT[axis]` TRUE: Brake failure / FALSE: Normal brake (default)

9.40.5 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 under updating.
- Dual motor is under the adjusting 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.
- Any axis is near by stroke limit when brake check starts.

9.40.6 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.

9.41 PANEL WIZARD

9.41.1 Overview

The Panel Wizard option allows you to use the controller to create up to four operation panels to be displayed on the *iPendant*. You don't need PC to create panel.

9.41.2 Software Requirements and Explanation

- 1 This function is Panel wizard (A05B-2500-R594).
- 2 This function is option.
- 3 KAREL use support (A05B-2500-J971) includes this function.
- 4 You can order R594 only, not J971.
- 5 System software version must be 7DA5 series or later.

9.41.3 Hardware Requirements

This function needs *iPendant*.

9.41.4 Setting up

9.41.4.1 Overview

Use procedure 9-31 to setup the Panel Wizard

Procedure 9-31 Setting up the Panel Wizard

Step

- 1 Press MENU.
- 2 Select BROWSER.
- 3 Press F1, [TYPE].
- 4 Select Panel Wizard. You will see screen similar to the following.



Fig. 9.41.4.1 (a): Panel Wizard start screen

- 5 Select Start Wizard. You will see a screen similar to the following.

NOTE

Double Pane mode will be displayed automatically.

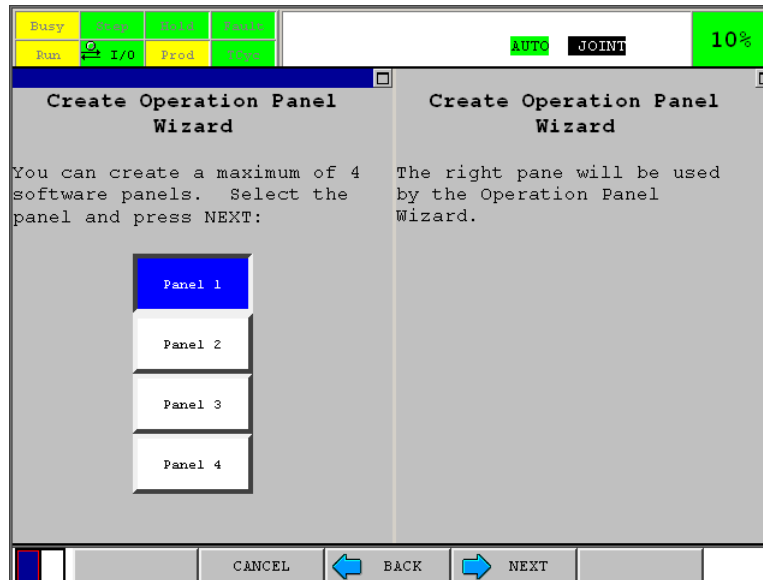


Fig. 9.41.4.1 (b) Panel selection screen

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 Fig. 9.41.4.1 (c) for a flowchart overview of the steps required to create an operator panel.

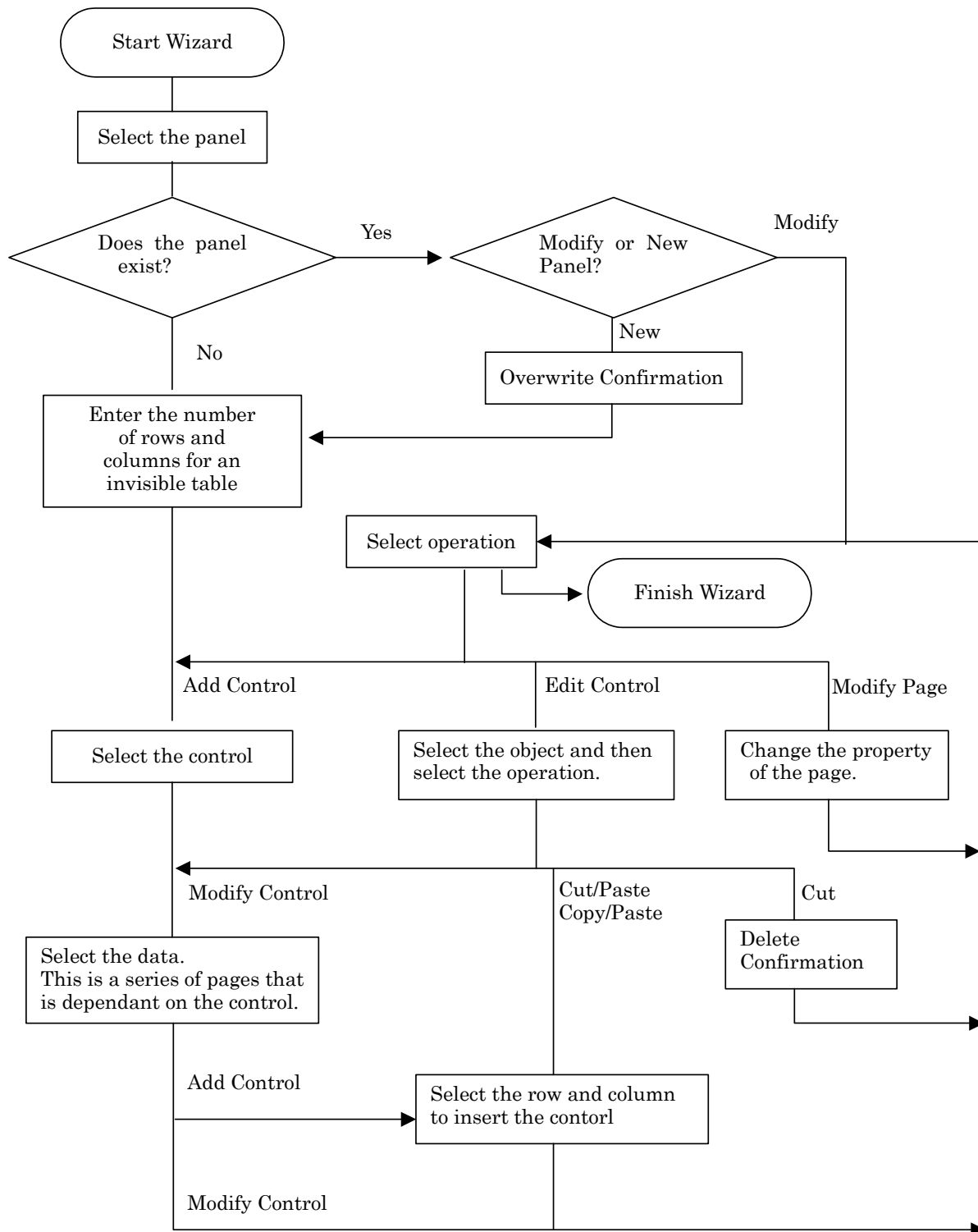


Fig. 9.41.4.1 (c) Create operation panel flowchart

NOTE

Control in the chart is *i*Pendant control. It corresponds to parts like buttons displayed by Panel Wizard

9.41.4.2 Available iPendant Controls

Panel wizard create operation panel by placing iPendant control on invisible table. iPendant Control to add is selected in a screen similar to following. Please refer to fig. 9.41.4.1.(c) for how to display the screen.

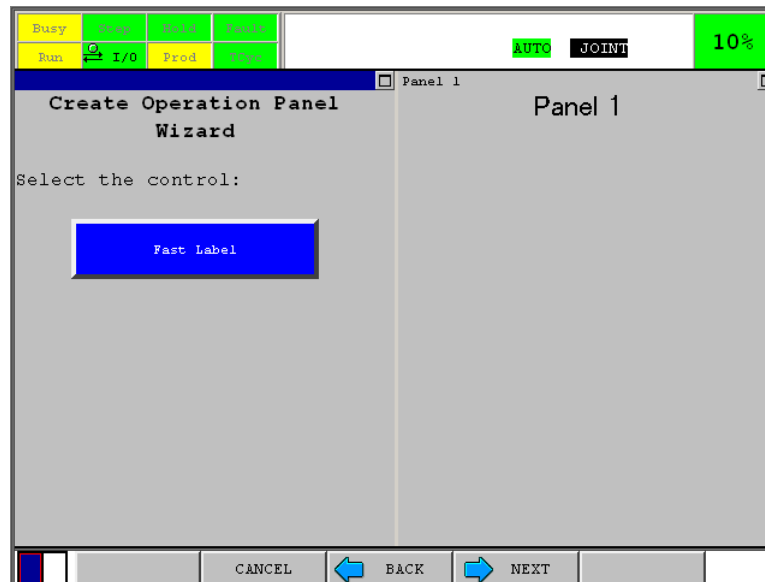


Fig. 9.41.4.2 iPendant Control selection screen

Pressing button on the center of the screen displays lists of available iPendant Controls. Following table lists the choices.

Table 9.41.4.2 Available iPendant controls

Choice	Description
Fast Label	This is simply shorted version of another choice "Label". It is described later. Refer to the description.
Fast lamp	This is simply shorted version of another choice "Toggle Lamp". It is described later. Refer to the description.
Fast Switch	This is simply shorted version of another choice "Toggle Button". It is described later. Refer to the description.
Button Change	This control is used to display the specified web page. It can also be used to run KAREL program.
Command Button	This control is used to write the specified value to a Register, System or KAREL Variable (except XYZWPR type) or I/O whenever the button is pushed.
EditBox	This control is used to change the value of a Register, System or KAREL Variable (except XYZWPR type) or I/O.
Label	The value of a Register, System or KAREL Variable or I/O is displayed.
Toggle Button	This control is used to change the value of a Register, System or KAREL Variable (except XYZWPR type) or I/O to the specified value following the ON (Pushed)/OFF (Popped) status of the button.
Toggle Lamp	This control is used to change the color of the control if the value of a Register, System or KAREL Variable (except XYZWPR type) or I/O fulfills the specified condition with the specified value or not.

Setup of each control is described following sections.

NOTE

Wizard is always display in double pane mode. Setup is done in the left pane. Right pane is used to display rough image of result of setup. Because display on the right pane depends on content of setup, figures in this manual usually do not match what you see during setup.

9.41.4.3 Setting up Fast Label

Fast lamp is shorted version of Label control. Label control displays value of a Register, System or KAREL Variable or I/O. Use procedure 9-32 to add Fast Label.

Procedure 9-32 Addition of Label control by Fast Label

Step

- 1 Select Fast Label in *i*Pendant Control selection screen (Fig. 9.41.4.2).
- 2 Press F4. You will see a screen similar to the following.

Fig. 9.41.4.3 (a) Size setup screen

Table 9.41.4.3 (a) Items in size setup screen

Item	Description
Caption (format)	If data type is static (this is always true for Fast Label), input string is text displayed. If the other data type is used, caption is format string.
Font Size	Size of font. Choices are 14,16,18 and 24.
Bold	If this is checked, value is displayed in bold.
Width and %	This is width of control. If % is not checked, width is specified in pixels. If % is checked, width is specified in percentage of screen size.
Height and %	This is height of control. Unit is specified in the same way as width.

NOTE

In setup of Fast Label, data type is default to static. Input string to be displayed to Caption.

- 3 Setup items as required
- 4 Press F4, NEXT. You will see a screen similar to the following.

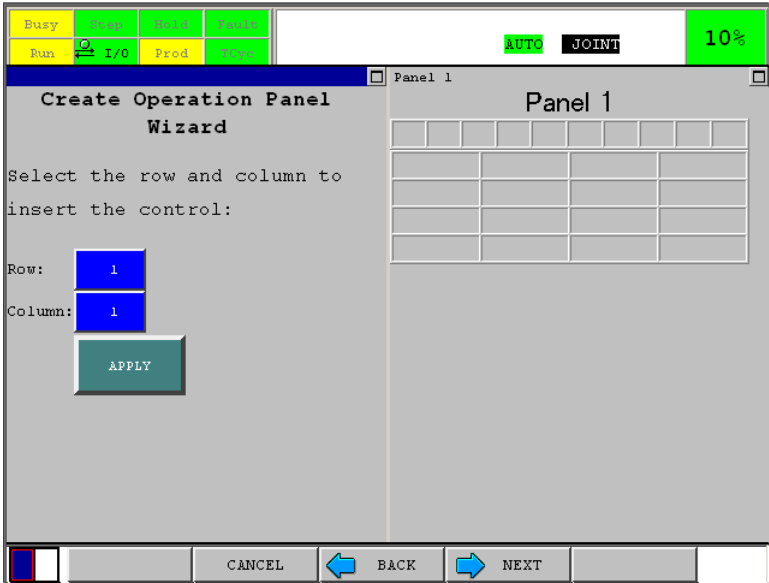


Fig. 9.41.4.3 (b) Control insertion screen

Table 9.41.4.3 (b) Items in control insertion screen

Item	Description
Row and Column	Row and column to insert iPendant Control
APPLY	This is not setup item. Pressing the button displays rough image of screen after insertion is displayed on right pane. Please note that screen image in right pane is just a rough guide.

5 Press F4, NEXT. This is end of addition of Label control by Fast Label.

9.41.4.4 Setting up fast lamp

Fast lamp is shorted version of Toggle lamp. Toggle lamp is used to change the color of the control if the value of Register, System or KAREL Variable (except XYZWPR type) or I/O fulfills the specified condition with the specified value or not. Use procedure 9-33 to add Fast Lamp.

Procedure 9-33 Addition of toggle lamp by Fast Lamp

Step

- 1 Select Fast Lamp in iPendant Control selection screen (Fig. 9.41.4.2).
- 2 Press F4. You will see a screen similar to the following.

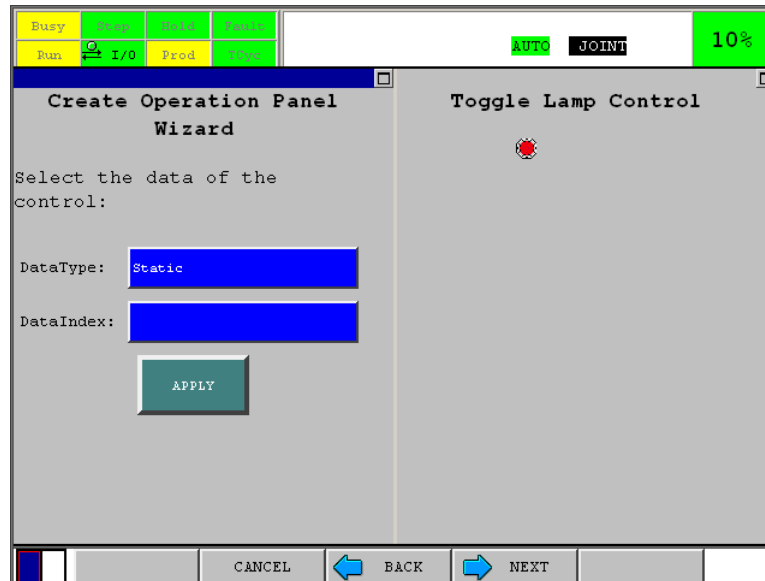


Fig. 9.41.4.4 (a) Date type setup screen

DataType specifies the type of the data to be monitored or modified depending on the type of control. DataIndex specifies the number or the variable name associated with the DataType.

Table 9.41.4.4 (a) Data type and Data index

Data type	Description
Static	String in caption is displayed.
Numeric Register	The value of numeric register specified in Date index is used.
System variable	The value of system variable specified in Data index is used. For System Variables, the type must be Integer, Real, Boolean, Short, Byte, or String. For example, \$MNUFRAMENUM[1]
KAREL variable	The value of KAREL variable specified in Data index is used. For KAREL Variables, enclose the program name inside [...]. The type must be Integer, Real, Boolean, Short, Byte, or String. Following is example of variable STR_VAR of KAREL program USEREXT. [USEREXT]STR_VAR
Dictionary Element	Dictionary element specified in Data index is used. For Dictionary Elements, specify the dictionary name and enclose the dictionary element inside [...]. For example, TPAR[5] Only Label control uses dictionary element.
DI	The value of DI specified in Data index is used.
DO	The value of DO specified in Data index is used.
..... (the other I/O type)	The value of I/O type specified in Data type and Data index is used. Data index is port number.

Toggle lamp uses specified data to decide ON/OFF of lamp.

3 Setup items as required

4 Press F4, NEXT. You will see a screen similar to the following.

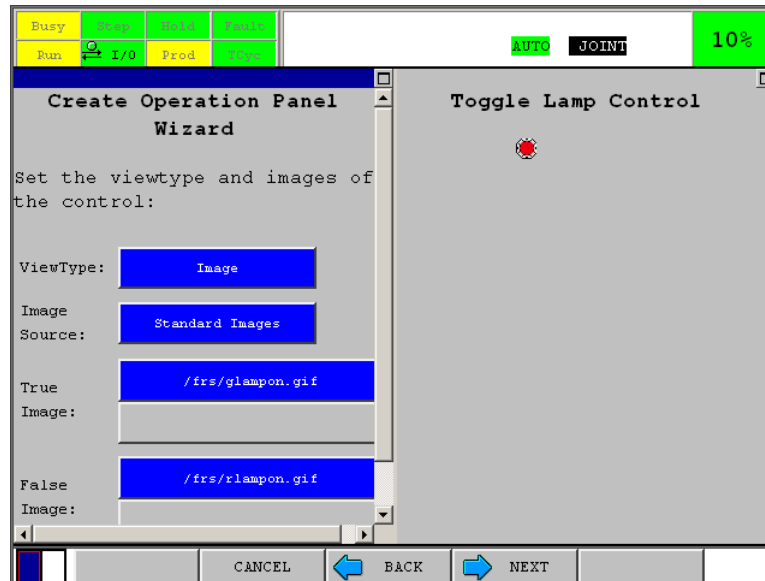


Fig. 9.41.4.4 (b) Image setup screen

Table 9.41.4.4 (b) Items in image setup screen

Item	Description
View type	Type of button to display Panel: specify panel type Circle: specify circle type Fixed circle: specify circle with fixed size Image: specify image type
Image Source	This is active when view type is image. Standard Images: image files installed can be selected in sub menu of True image and false image. fr:*.gif : GIF file in FR: can be used. FR is device on F-ROM of controller. fr:*.jpg : JPG file in FR: can be used. Keyboard Entry: Input path of image file.
True Image	This specifies the image to be displayed when the read value fulfill the condition expression. Used only in case that ViewType is Image type. When image source is Keyboard Entry, button of lower side is active. Input path of image file by the button. When image source is not Keyboard Entry, button of upper side is active. Submenu is displayed by the button. Select image file from the sub menu.
False Image	This specifies the image to be displayed when the read value does not fulfill the condition expression. This is used only in case that View Type is Image type. Refer to description of True Image for how to specify image file.

- 5 Setup items as required.
- 6 Press of F4, NEXT leads you to control insertion screen. Please refer to Section 9.41.4.3.
- 7 Decide position to insert and press F4, NEXT. This is end of addition of Toggle Lamp by Fast Lamp.

9.41.4.5 Setting up fast switch

Fast switch is shorted version of Toggle Button. Toggle button is used to change the value of a Register, System or KAREL Variable (except XYZWPR type) or I/O to the specified value following the ON (Pushed)/OFF (Popped) status of the button. Use procedure 9-34 to add Toggle button by Fast Switch.

Procedure 9-34 Addition of Toggle button by Fast Switch

Step

- 1 Select Fast Switch in *iPendant Control* selection screen (Fig.9.41.4.2)
- 2 Press of F4, NEXT leads you to data type setup screen. Please refer to section 9.41.4.4.
- 3 Press of F4, NEXT leads you to image setup screen (Fig. 9.41.4.4). Screen is same as section 9.41.4.4. But choices of view type are different.



Fig. 9.41.4.5 Toggle button image setup screen

Table 9.41.4.5 Items in toggle button image setup screen

Item	Description
View type	Type of button to display. Panel: specify panel type Check box: specify check box type Image: specify image type F2-F5, F7-F10: specify function key type. Circle: specify circle (Size of circle is fixed. This is just like fixed circle)
Image Source	This is same as toggle lamp. Please refer to table 9.41.4.4 (b).
True Image	This specifies the image to be displayed when the button status is ON. This is used only in case that ViewType is image type. How to setup is just same as toggle lamp. Please refer to table 9.41.4.4 (b).
False Image	This specifies the image to be displayed when the button status is OFF. This is used only in case that ViewType is image type. How to setup is just same as toggle lamp. Please refer to table 9.41.4.4 (b).

- 4 Press of F4, NEXT leads you to control insertion screen. Please refer to Section 9.41.4.3.
- 5 Decide position to insert and press F4, NEXT. This is end of addition of Toggle button by FAST Switch.

9.41.4.6 Addition of button change control

Button Change control is used to display the specified web page. It can also be used to run KAREL program. Method to run KAREL program is described later. Use Procedure 9-35 to add Button change control.

Procedure 9-35 Addition of Button Change Control

Step

- 1 Select Button Change in *iPendant* Control selection screen (Fig. 9.41.4.2).
- 2 Press F4. You will see a screen similar to the following.

Fig. 9.41.4.6 (a) URL setup screen

Table 9.41.4.6 (b) URL setup screen

Item	Description
PageName	Press of upper side button displays STM/HTM files in FR:. Select file to display. If you want to display the other file, press lower side button and input URL.

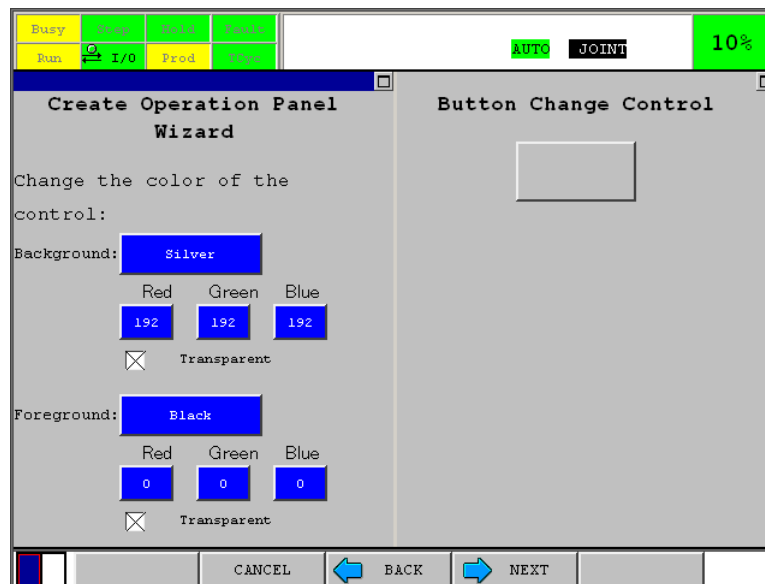
- 3 Press of F4, NEXT, leads you to Button Change image setup screen. Screen is same as image setup screen of section 9.41.4.4. But choices of view type are different.

Fig. 9.41.4.6 (b) Button Change image setup screen

Table 9.41.4.6 (b) Items in Button change image setup screen

Item	Description
View type	Type of button to display. Normal: Specify the normal button. Image: specify image type F2-F5, F7-F10: specify function key type
Image Source	This is same as toggle lamp. Please refer to table 9.41.4.4 (b).
True Image	This specifies the image to be displayed when the button is pushed. Used only in case that ViewType is Image type. How to setup is just same as toggle lamp. Please refer to table 9.41.4.4 (b).
False Image	Specify the image to be displayed when the button is not pushed. Used only in case that ViewType is Image type. How to setup is just same as toggle lamp. Please refer to table 9.41.4.4 (b).

6 Press F4, NEXT. You will see a screen similar to the following.

**Fig. 9.41.4.6 (c) Color setup screen****Table 9.41.4.6 (c) Items in Color setup screen**

Item	Description
BackGround	This specifies background color. Pressing button displays list of standard colors in submenu.
(BackGround) Red, Green, Blue	Specify background color by RGB. This enables more detailed setup than selection from menu of BackGround button.
(BackGround) Transparent	With check of this item, background color of control becomes background color of web page.
Foreground	This specifies color of character. How to setup is same as background color.
(Foreground) Red, Green, Blue	Please refer to the same item for background color.
(Foreground) Transparent	With check of this item, foreground color of the control becomes foreground color of web page.

- 4 Press of F4, NEXT leads you to size setup screen (Fig. 9.41.4.3 (a)). Please refer to Section 9.41.4.3. Caption is used as fixed string. It is not used as format.
- 5 Press F4, NEXT. You will see a screen similar to the following.

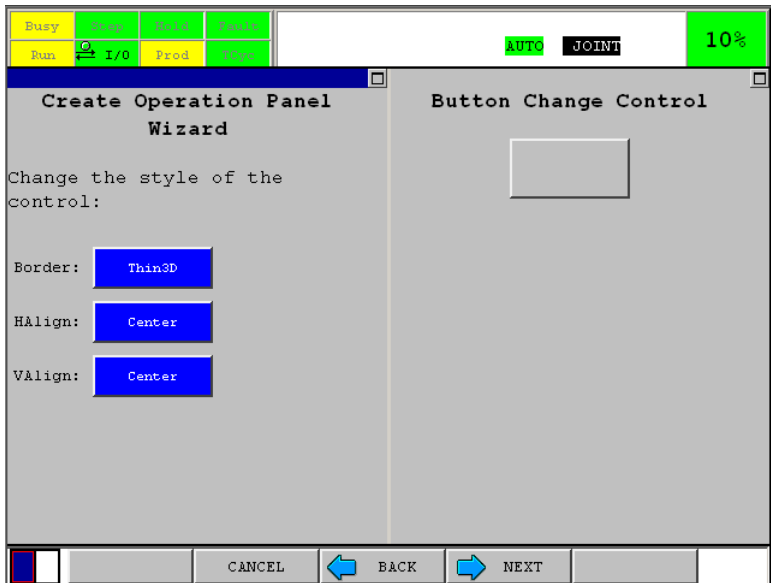


Fig. 9.41.4.6 (d) Style setup screen

Table 9.41.4.6 (d) Items in style setup screen

Item	Description
Border	This specifies type of borderline of button. Choices are Thin 3D, None, Black, ForeColor and Bold 3D.
HAlign	This specifies alignment of string in horizontal direction. Choices are Left, Center and Right.
VAlign	This specifies alignment of string in vertical direction. Top, Center and Bottom.

Following figure shows difference between types of borderline. The right pane shows buttons of each 5 border types. This figure is just a sample that has nothing to do with flow to add button change control.

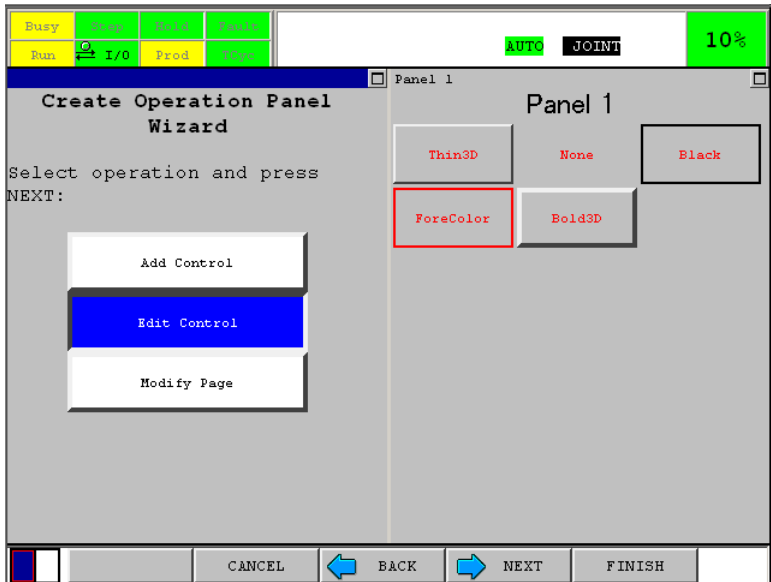


Fig. 9.41.4.6 (e) Type of borderlines

- 6 Press of F4, NEXT leads you to control insertion screen. Please refer to Section 9.41.4.3.
- 7 Decide position to insert and press F4, NEXT. This is end of addition of Button Change control.

9.41.4.7 Addition of command button control

Command Button Control is used to write the specified value to a Register, System or KAREL Variable (except XYZWPR type) or I/O whenever the button is pushed. Use Procedure 9-36 to add Command Button Control.

Procedure 9-36 Addition of Command Button Control

Step

- 1 Select Command Button in *iPendant* Control selection screen (Fig. 9.41.4.2).
- 2 Press of F4, NEXT leads you to data type setup screen. Please refer to section 9.41.4.4.
- 3 Press F4, NEXT. You will see a screen similar to the following.

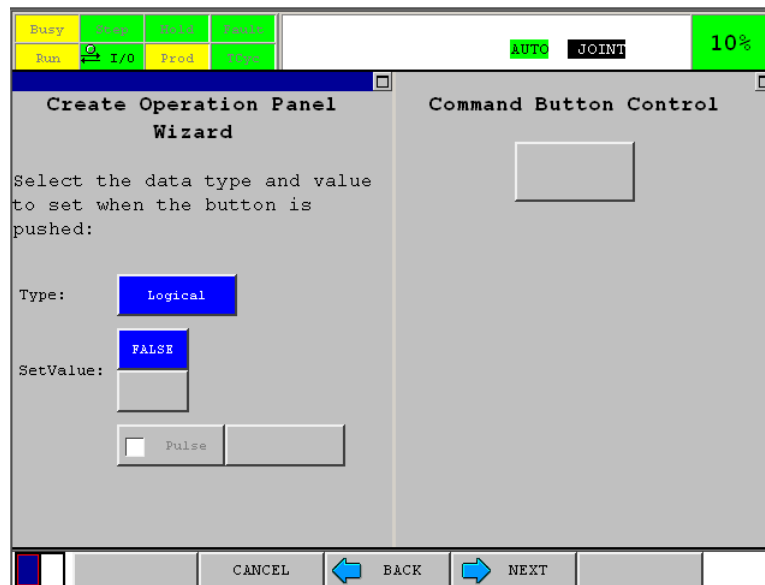


Fig. 9.41.4.7 Command Button setup screen

Table 9.41.4.7 Items in Command Button setup screen

Item	Description
Type	Logical: Data is transacted as bool type. (SetValue is TRUE/FALSE) Numerical: Data is transacted as numerical type. The value of SetValue is applied.
SetValue	This specifies the value written when the button is pushed.
Pulse	This is active only when data type is DO. If this is checked, signal output becomes pulsed output.
Pulse width (right item of pulse check box)	This specifies width of pulse in msec.

- 4 Press of F4, NEXT leads you to image setup screen. This is same as Button change image setup screen (Fig. 9.41.4.6 (b)). Please refer to section 9.41.4.6.
- 5 Press of F4, NEXT leads you to color setup screen. Please refer to section 9.41.4.6.
- 6 Press of F4, NEXT leads you to size setup screen. Please refer to section 9.41.4.3. Caption is used as fixed string. It is not used as format.
- 8 Press of F4, NEXT leads you to style setup screen. Please refer to section 9.41.4.6.
- 9 Press of F4, NEXT lead you to control insertion screen. Please refer to section 9.41.4.3.
- 10 Decide position to insert and press F4, NEXT. This is end of addition of Command Button Control.

9.41.4.8 Addition of edit box control

Edit Box Control is used to change the value of a Register, System or KAREL Variable (except XYZWPR type) or I/O. Use Procedure 9-37 to add Edit Box Control

Procedure 9-37 Addition of Edit Box Control

Step

- 1 Select Edit Box in iPendant Control selection screen (Fig. 9.41.4.2).
- 2 Press of F4, NEXT leads you to data type setup screen. Please refer to section 9.41.4.4.
- 3 Press F4, NEXT. You will see a screen similar to the following.

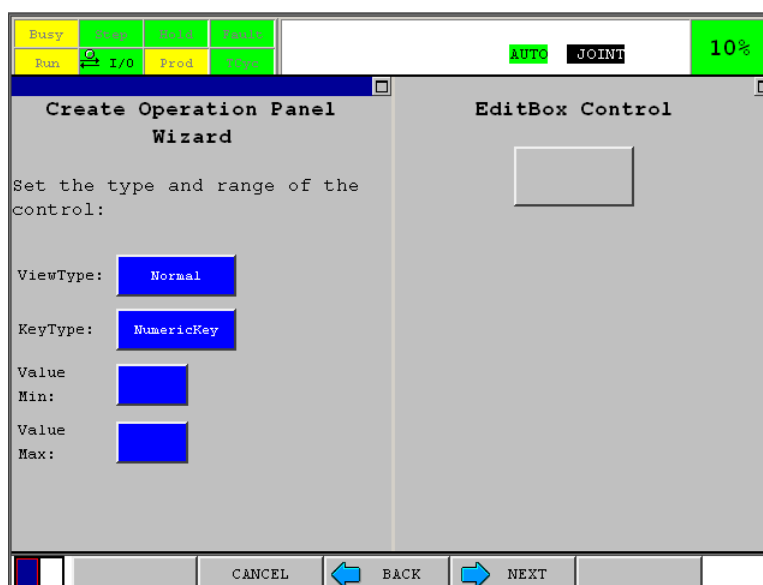


Fig. 9.41.4.8 Edit Box Setup screen

Table 9.41.4.8 Items in Edit Box Setup screen

Item	Description
View type	Normal: specify simple button F2-F5, F7-F10: specify function key type.
KeyType	Select the type of virtual keyboard. NumericKey: virtual keyboard for number input FullKey: virtual keyboard for string input
Value Min, Value Max	Minimum and maximum value of numerical value.

- 4 Press of F4, NEXT leads you to color setup screen. Please refer to section 9.41.4.6.
- 5 Press of F4, NEXT leads you to size setup screen. Please refer to section 9.41.4.3.
- 6 Press of F4, NEXT leads you to style setup screen. Please refer to section 9.41.4.6
- 7 Press of F4, NEXT lead you to control insertion screen. Please refer to section 9.41.4.3.
- 8 Decide position to insert and press F4, NEXT. This is end of addition of Edit Box Control.

9.41.4.9 Addition of label control

Label control displays value of a Register, System or KAREL Variable or I/O. Use procedure 9-38 to add Label Control.

Procedure 9-38 Addition of Label Control

Step

- 1 Select Label in *i*Pendant Control selection screen (Fig. 9.41.4.2).
- 2 Press of F4, NEXT leads you to data type setup screen. Please refer to section 9.41.4.4.
- 3 Press of F4, NEXT leads you to size setup screen. Please refer to section 9.41.4.3.
- 4 Press of F4, NEXT leads you to style setup screen. Please refer to section 9.41.4.6
- 5 Press of F4, NEXT leads you to color setup screen. Please refer to section 9.41.4.6.
- 6 Press of F4, NEXT lead you to control insertion screen. Please refer to section 9.41.4.3.
- 7 Decide position to insert and press F4, NEXT. This is end of addition of Label Control.

Setup of Fast Label included step 3 and 6 only. Default value is used for items in the other steps. You can add Label Control by Fast Label and then modify it.

9.41.4.10 Addition of toggle button control

Toggle Button Control is used to change the value of a Register, System or KAREL Variable (except XYZWPR type) or I/O to the specified value following the ON (Pushed)/OFF (Popped) status of the button. Use procedure 9-39 to add Toggle Button Control.

Procedure 9-39 Addition of Toggle Button Control

Step

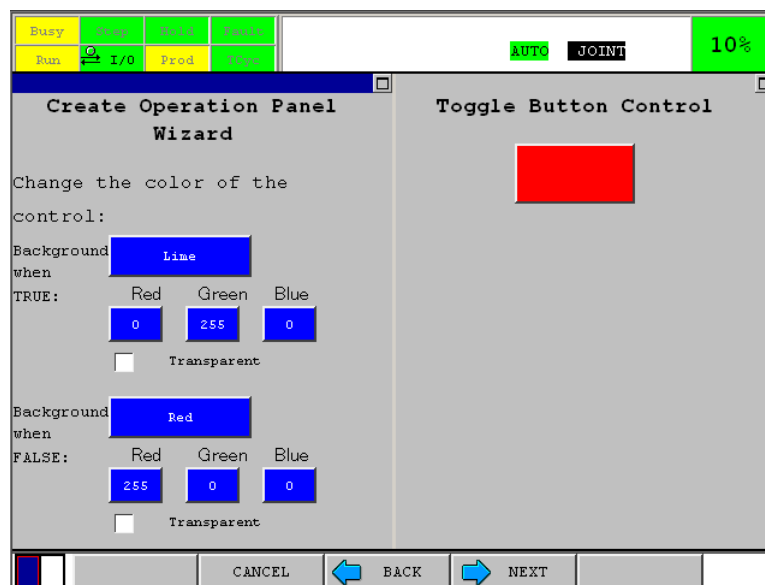
- 1 Select Toggle Button in *i*Pendant Control selection screen (Fig. 9.41.4.2).
- 2 Press of F4, NEXT leads you to data type setup screen. Please refer to section 9.41.4.4.
- 3 Press F4, NEXT. You will see a screen similar to the following.

Fig. 9.41.4.10 (a) Toggle Button setup screen

Table 9.41.4.10 (a) Items in Toggle Button setup screen

Items	Description
Type	This specifies data type to process. Logical: Data is transacted as boolean type. False value and True Value are not used. Numerical: Data is transacted as numerical type. False Value and True Value are used.
False Value	This is active when Type is Numerical. This specifies the value written when the button status is changed to ON (TRUE).
True Value	This is active when Type is Numerical. This specifies the value written when the button status is changed to OFF (FALSE).
Other Phase	This is active when Type is Numerical. This specifies the status (TRUE/FALSE) in case that the value is equal to neither TrueValue nor FalseValue.
Caption	This specifies the fixed String.
True Caption (optional)	This specifies the fixed String when the value is TRUE. If not specified, Caption is used.

- 4 Press of F4, NEXT leads you to Toggle Button Image setup screen. Please refer to section 9.41.4.5.
- 5 Press F4, NEXT. You will see a screen similar to the following.

**Fig. 9.41.4.10 (b) True/False color setup screen****Table 9.41.4.10 (b) Items in True/False color setup screen**

Item	Description
Background when TRUE	This specifies the color displayed when the read value is equal to TrueValue or not equal to FalseValue. How to setup is same as color setup screen. Please refer to section 9.41.4.6.
Red, Green, Blue	Role is same as color setup screen. Please refer to section 9.41.4.6.
Transparent	Role is same as color setup screen. Please refer to section 9.41.4.6.
Background when FALSE	This specifies the color displayed when the read value is equal to FalseValue or not equal to TrueValue. How to setup is same as color setup screen. Please refer to section 9.41.4.6.

- 6 Press of F4, NEXT leads you to color setup screen. Please refer to section 9.41.4.6.
- 7 Press of F4, NEXT leads you to size setup screen. Please refer to section 9.41.4.3. Caption is used as fixed string. It is not used as format.
- 8 Press of F4, NEXT leads you to style setup screen. Please refer to section 9.41.4.6.
- 9 Press of F4, NEXT lead you to control insertion screen. Please refer to section 9.41.4.3.
- 10 Decide position to insert and press F4, NEXT. This is end of addition of Toggle Button Control.

Setup of Fast Switch includes setup of data type (step2), image (step 4) and control insertion (step 9) only. Default value is used for items in the other steps. You can add Toggle Button Control by Fast Switch and then modify it.

9.41.4.11 Addition of toggle lamp control

Toggle Lamp Control is used to change the color of the control if the value of a Register, System or KAREL Variable (except XYZWPR type) or I/O fulfills the specified condition with the specified value or not. Use procedure 9-40 to add Toggle Lamp Control

Procedure 9-40 Addition of Toggle Lamp Control

Step

- 1 Select Toggle Lamp in *i*Pendant Control selection screen (Fig. 9.41.4.2).
- 2 Press of F4, NEXT leads you to data type setup screen. Please refer to section 9.41.4.4.
- 3 Press F4, NEXT. You will see a screen similar to the following.

Fig. 9.41.4.11 Toggle Lamp setup screen

Table 9.41.4.11 (a) Items in Toggle Lamp setup screen

Item	Description
Type	This specifies data type to process Logical: Data is transacted as bool type Numerical: Data is transacted as numerical type. The setting of CmpOperator and the value of CmpValue is applied.
CmpOperator	This is active when Type is Numerical. This selects the condition expression evaluated as TRUE. EQ: Specify the equal case (=). NE: Specify the not equal case (<>). LT: Specify the less than case (<). LE: Specify the less than or equal case (<=). GT: Specify the greater than case (>). GE: Specify the greater than or equal case (>=).
CmpValue	This specifies the standard value for comparison. Result of comparison between read value and this value decides status of lamp, TRUE or FALSE. It is reflected to display of lamp.

- 4 Press of F4, NEXT leads you to image setup screen. Please refer to section 9.41.4.4.

- 5 Press of F4, NEXT leads you to Fig. 9.41.4.10 (b): True/False color setup screen (Fig. 9.41.4.10 (b)). Please refer to section 9.41.4.10.
- 6 Press of F4, NEXT leads you to color setup screen. Please refer to section 9.41.4.6.
- 7 Press of F4, NEXT leads you to size setup screen. Please refer to section 9.41.4.3. Caption is used as fixed string. It is not used as format.
- 8 Press of F4, NEXT leads you to style setup screen. Please refer to section 9.41.4.6
- 9 Press of F4, NEXT lead you to control insertion screen. Please refer to section 9.41.4.3.
- 10 Decide position to insert and press F4, NEXT. This is end of addition of Toggle Lamp Control.

Setup of Fast Lamp includes setup of data type (step2), image (step4) and control insertion (step9) only. Default value is used for items in the other steps. You can add Toggle Lamp Control by Fast Switch and then modify it.

Background color used by Toggle Lamp depends on its View type.

Table 9.41.4.10 (b) BackGround color of Toggle Button

View type	Status	Background color
Panel	True	Background when TRUE
	False	Background when FALSE
Circle, Fixed Circle	True	Circle part: Background when TRUE (If transparent is checked, black) The other part: Background
	False	Circle part: Background when TRUE (If transparent is checked, black) The other part: Background
Image	Both true and false	Image uses background color if it uses transparent color.
		1 Background when TRUE (False) if transparent is not checked
		2 Background if transparent is not checked
		3 Otherwise background color of web page

9.41.4.12 Modification of panel

Use procedure 9-41 to modify created panel.

Procedure 9-41 Modification of panel

- 1 Start panel wizard by procedure 9-31 and select panel.
- 2 If selected panel already exists, you will see a screen similar to the following.

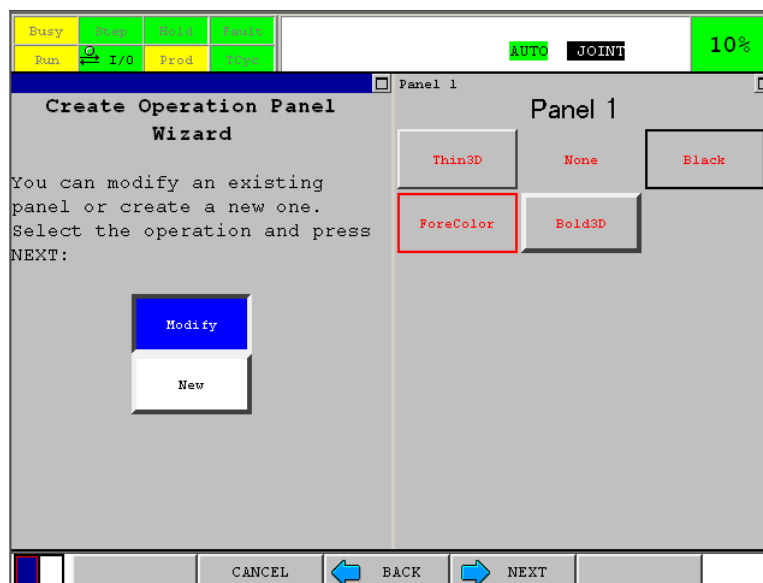


Fig. 9.41.4.12 (a) Selection of modification or creation

- 3 Select modify and press F4, NEXT. You will see a screen similar to the following.

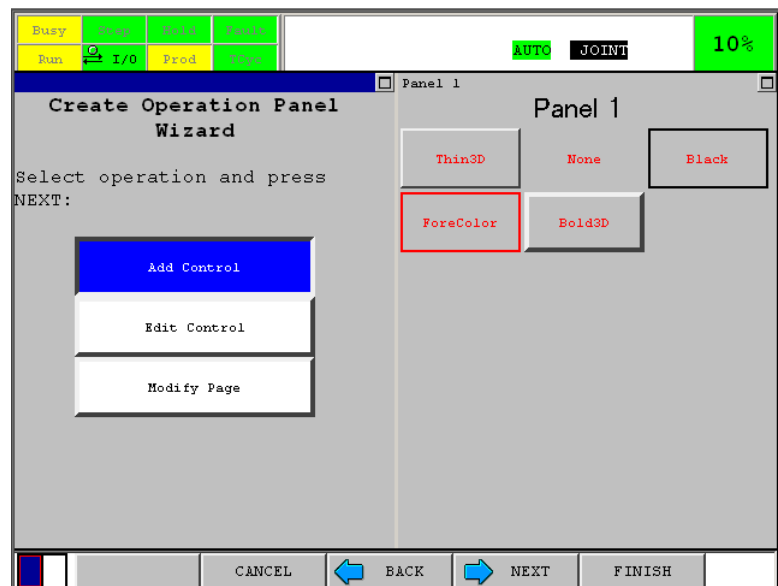


Fig. 9.41.4.12 (b) Selection of operation screen

Table 9.41.4.12 Items in selection of operation screen

Item	Description
Add Control	You can add control to panel. Please refer to section for control to add.
Edit Control	You can modify control already added to panel. Please refer to section 9.41.4.13.
Modify Page	You can modify other property of panel. For example, title of panel, background color. Please refer to section 9.41.4.16.

- 4 Select operation and press F4, NEXT.

For detail of each operation, please refer to section written in table above.

9.41.4.13 Modification of control

Use procedure 9-42 to modify control that is already added to panel.

Procedure 9-42 Modification of control

- 1 Display selection of operation screen by procedure 9-41.
- 2 Select Edit Control.
- 3 Press F4, NEXT. You will see a screen similar to the following.

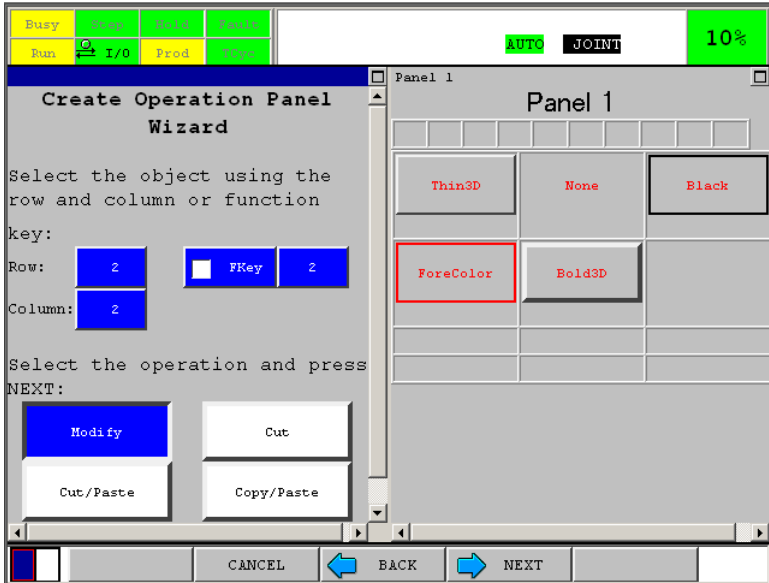


Fig. 9.41.4.13 Edit control screen

Table 9.41.4.13 Items in Edit control screen

Item	Description
Row, Column	These specifies control to modify, cut, and copy.
Fkey	If this is checked, control assigned to function key is selected.
The number next to Fkey	This number specifies function key when Fkey is checked.
Modify	Select this to modify control already added to panel.
Cut	Select this to delete control.
Cut/Paste	Select this to cut selected control and paste it elsewhere.
Copy/Paste	Select this to copy selected control and paste it elsewhere.

- 4 Select Modify and press F4, NEXT.
- 5 Setup screen appropriate for the selected control is displayed. If you added the control by Fast label, Fast switch or Fast lamp, setup screen for corresponding control is displayed. For example, setup screen for Label control is displayed for control added by Fast label.
- 6 Setup control just same as addition of control. Control insertion screen at the last step is not displayed.

9.41.4.14 Delete of control

Use procedure 9-43 to delete control.

Procedure 9-43 Delete of control

- 1 Display Edit control screen by procedure 9-42.
- 2 Select control to delete.
- 3 Select Cut and press F4, NEXT.
- 4 You will see a screen similar to the following.

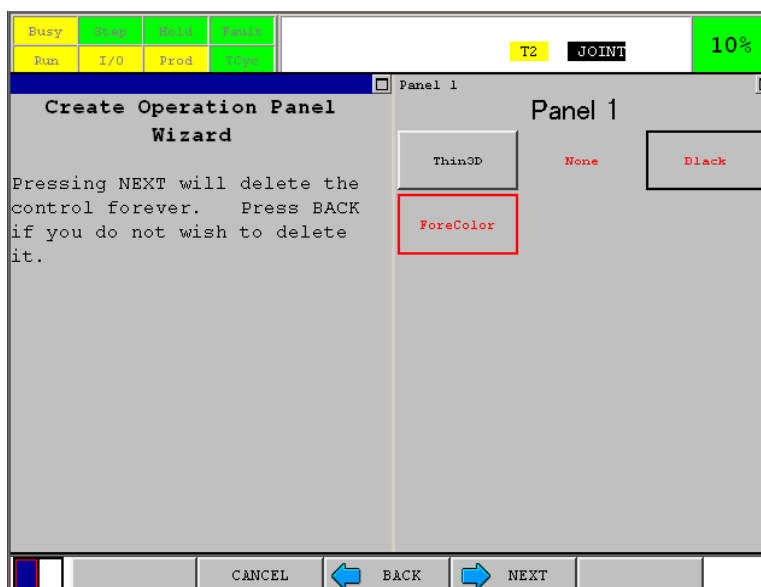


Fig. 9.41.4.14 Confirmation of delete of control

Press F4 to delete. Otherwise, press F3.

NOTE

F2 CANCEL cancel panel wizard. As shown in the right pane in the confirmation screen, specified control is already deleted (Press of F3 recovers it. Press of F4 is just to proceed). Press of F2 quit panel wizard at this stage and leave specified control deleted.

9.41.4.15 Cut/Copy paste of control

You can move added control by cut and paste. You can copy added control by copy and paste. Use procedure 9-44.

Procedure 9-44 Cut/Copy paste of control

- 1 Display Edit control screen by procedure 9-42.
- 2 Select control.
- 3 If you want to move the control, select Cut/Paste. If you want to copy it, select Copy/Paste. Press F4, NEXT.
- 4 Next screen depends on control you selected.
 - 4-1 If selected control is NOT assigned to function key, you will see a screen similar to the following.

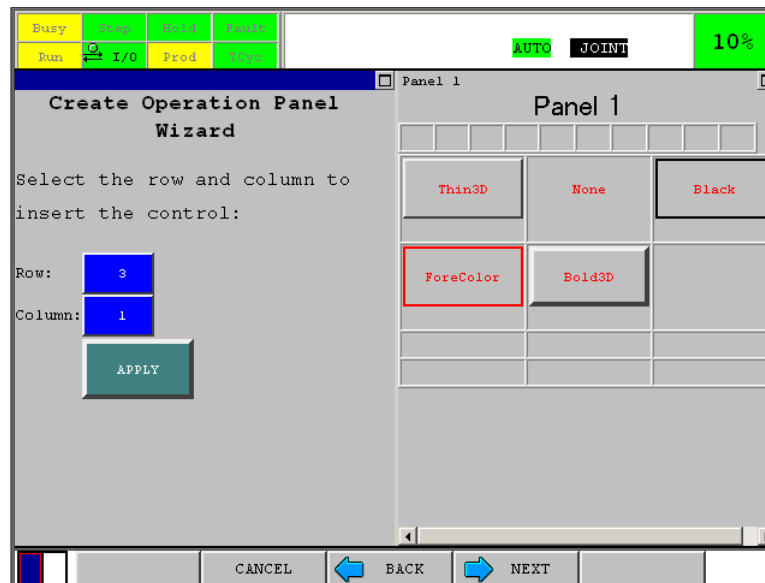


Fig. 9.41.4.15 (a) Control insertion screen

- 4-2 If selected control is assigned to function key, you will see a screen similar to the following. This screen is to select function key to paste control.

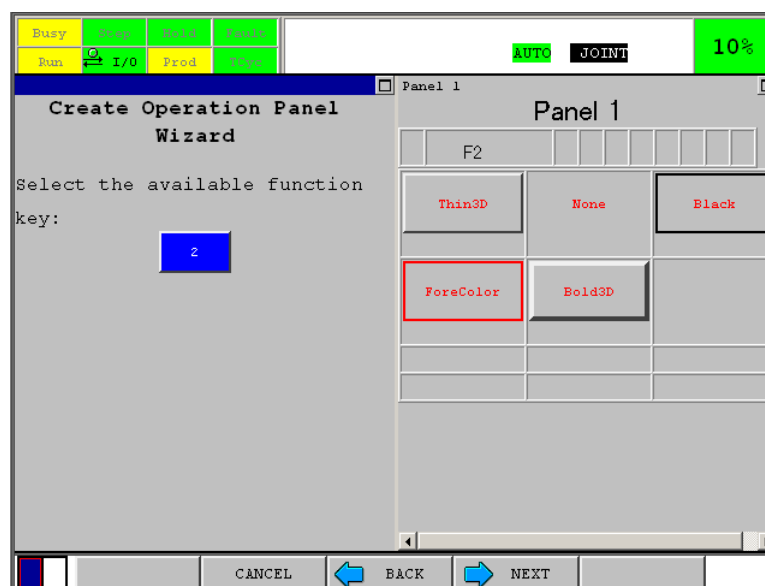


Fig. 9.41.4.15 (b) control insertion screen (function key)

- 5 Decide position to insert control and Press F4, NEXT.

9.41.4.16 Modification of page

Use procedure 9-45 to modify property of page (panel). Background color and title for example.

Procedure 9-45 Modification of page

- 1 Display selection of operation screen by procedure 9-41.
- 2 Select Modify page.
- 3 Press F4, NEXT. You will see a screen similar to the following.

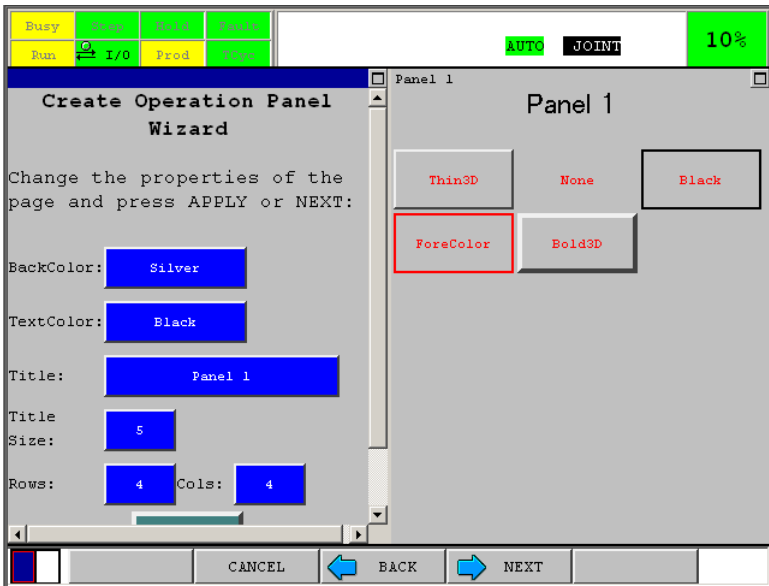


Fig. 9.41.4.16 Page edit screen

Table 9.41.4.16 Item in page edit screen

Item	Description
BackColor	This is the background color of the page.
Text Color	This is the color of title text of the page.
Title	This is title displayed at top of panel. If panel1, it defaulted to Panel 1.
Title Size	Size of title. Range is from 1 to 5.
Rows	Control is placed in invisible table. This item is the number of rows of the table. You cannot decrease it.
Cols	The number of columns of invisible table. You cannot decrease it.
APPLY	This is not item to setup. Press of the button applies property edited and you can see rough image in right pane.

7 Setup as required and Press F4, NEXT. This is the end of modification of page.

9.41.4.17 Re-creation of panel

Use procedure 9-46 to re-create panel with disposing existing panel.

Procedure 9-46 Re-creation of panel

- 1 Display panel selection screen by procedure 9-31.
- 2 Select panel to re-create and press F4, NEXT.
- 3 Screen to select modification or creation is displayed.
- 4 Select New.

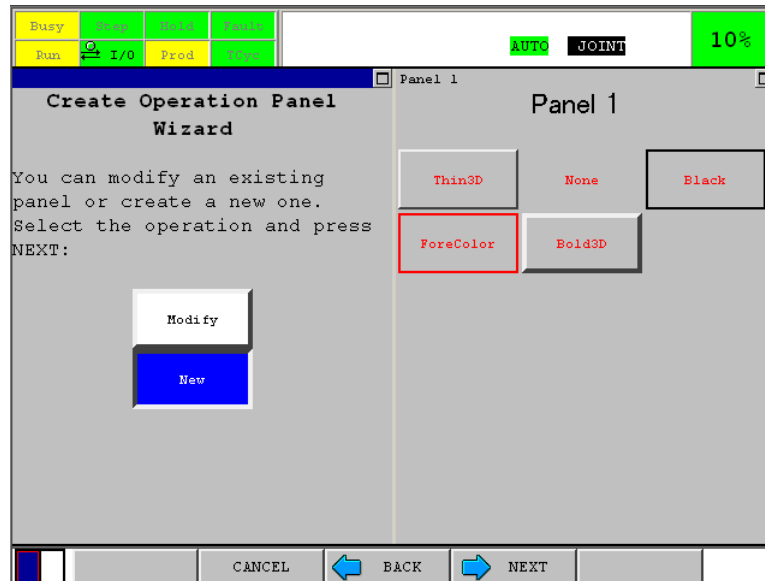


Fig. 9.41.4.17 (a) Selection of modification or creation

- 5 Press F4, NEXT. You will see a screen similar to the following.

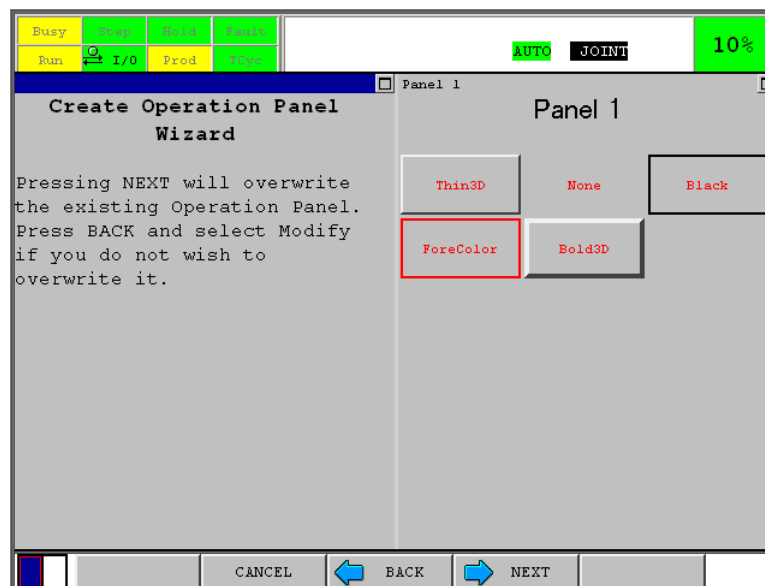


Fig. 9.41.4.17 (b) Confirmation of re-creation of panel

- 6 Press of F4 delete existing panel. Re-creation start from setup of the number of rows and columns.

9.41.5 Run KAREL Program by Panel

You can create button to run KAREL program. Required steps are following.

- (A) Creation of KAREL program that can be run by panel
- (B) Addition and setup of Button Change Control

NOTE

You cannot run arbitrary KAREL program. There is caution of creation of KAREL program for the program to be run by button change control

9.41.5.1 Caution for creation of KAREL program

- (A) Create static (program level) INTEGER variable, return_code. Set it to 204.
- (B) Program must complete in a short time
- (C) If it takes long time to complete the program, run another program which is started by RUN_TASK built-in in the program started by button change control.
- (D) Specify %NOLOCKGROUP.

9.41.5.2 Creation of run button

Use procedure 9-47 to create button to run program.

Procedure 9-47 Creation of run button

- 1 Start addition of Button Change Control and display URL setup screen.
Please refer to section 9.41.4.6 for procedure to add Button Change control.

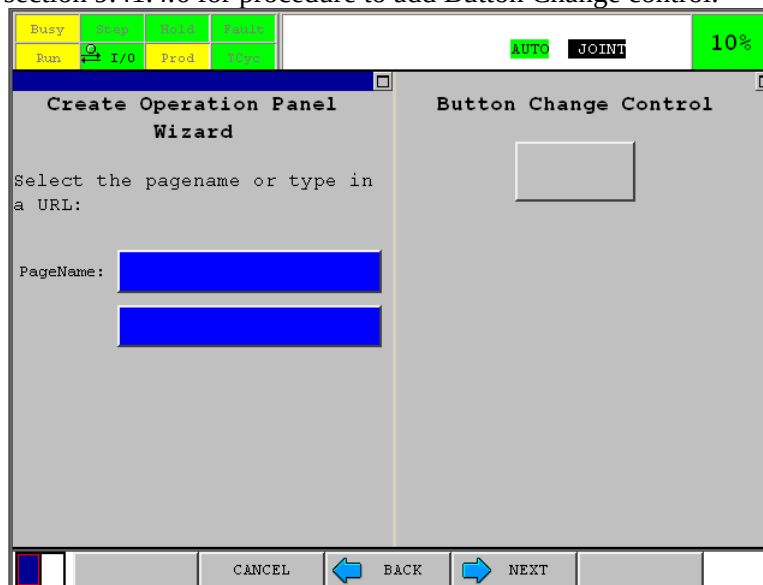


Fig. 9.41.5.2 (a) RUL setup screen

- 2 Press lower side button and input URL in format of “/KAREL/program name”
For example, /KAREL/TESTPROG to run KAREL program TESTPROG.

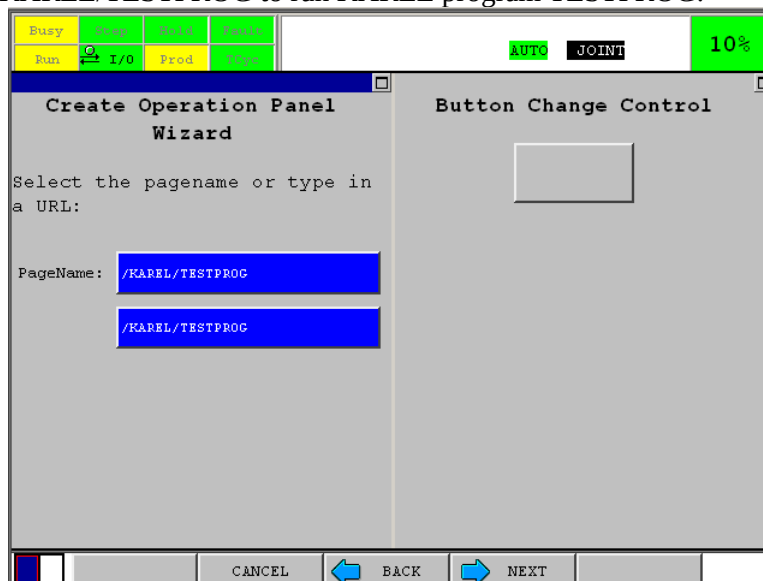


Fig. 9.41.5.2 (b) Setup to run TESTPROG

9.41.6 Usage

9.41.6.1 Display of panel

MENUS->BROWSER->[TYPE] displays panels created.

NOTE

Created panels are used in all languages. If you display the panel in more than one language, create the panel in English. Japanese panel is not displayed correctly in English.

If there are too many entries in [TYPE] menu of BROWSER to add panel to the menu, you will see a screen similar to following at finishing addition of panel.

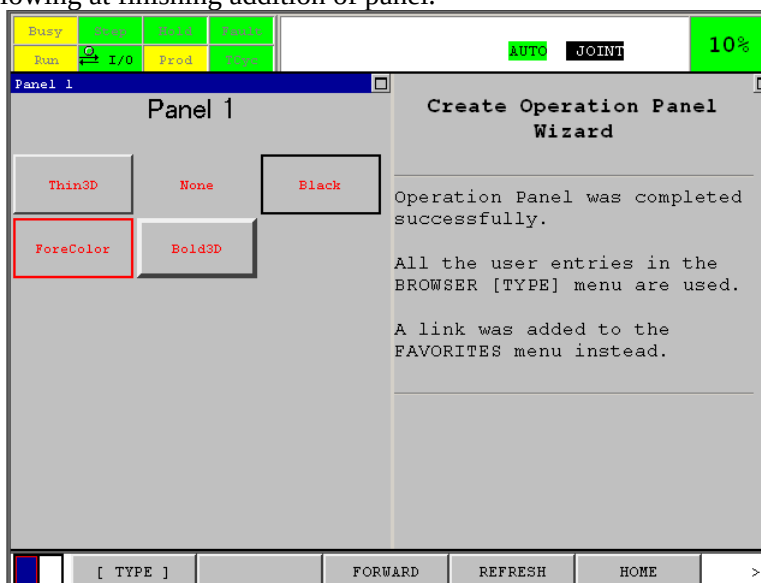


Fig. 9.41.6.1 (a) Notification of addition to favorite

In this case, panel is added to link in FAVORITE. Display panel by selecting link.

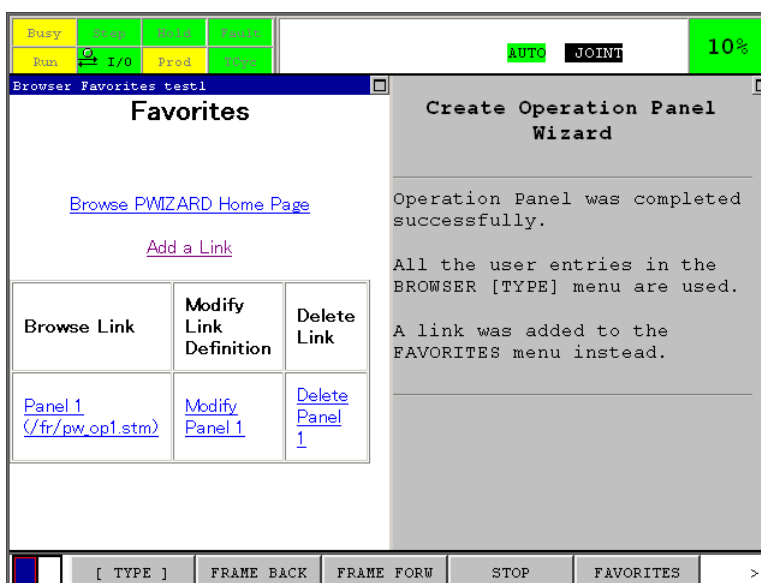


Fig. 9.41.6.1 (b) panel1 added to favorite

9.41.6.2 Backup/restore

Created panels can be saved and restored in file screen. It is categorized as “Application”.
Panel1...Panel 4 corresponds to PW_OP1.STM...PW_OP4.STM, respectively.

NOTE

PANEL1.STM,PANEL1.DT,.....,PANEL4.STM, PANEL4.DT have nothing to do with this function.

- If you use image file you prepared, use it in GIF or JPG format. Copy them to FR: and refer to it. FR: *.GIF and FR: *.JPG are saved and restored in file screen. They are also in category of “Application”.
- Registration data to menu of MENUS->BROWSER->[TYPE] is saved in SYSVARS.SV. If you want to backup/restore the registration, SYSVARS.SV should be backed up and restored.
- If you do not want to restore SYSVARS.SV, you have to register panel to [TYPE] menu manually. Use procedure below.
 - Restore backup of panel (PW_*.STM and required image files)
 - Display Selection of operation screen (Fig. 9.41.4.12 (b)) by procedure 9-41.
 - Press F5, FINISH. This means there is actually no change.
 - These steps add panel to [TYPE] menu of BROWSER.

9.42 ENHANCED MIRROR IMAGE

The enhanced mirror image option (A05B-2500-R698) 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 9.42 for descriptions of the Mirror Image Shift screen items.

Table 9.42 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 F4, [CHOICE], select the name of program, and press ENTER again.

ITEM	DESCRIPTION
Range	This item indicates whether the whole program (F5, [WHOLE]) or a portion (F4, [PART]) will be shifted using mirror image.
Start line (used only for "PART" range)	This item is the first line in the program that will be shifted using mirror image.
End line (used only for "PART" range)	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 insert into 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 Method	• F5, [TEACH] is a method that requires the user to enter position data to define the mirror plane. • F4, [FRAME] is a method that allows the user to choose existing frames.
Orientation Shift	• F5, [MIRROR] is a mirror image shifting method that mirrors the orientation as well as the positions. • F4, [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 (used only for "FRAME" method)	World, User Frame, or Jog Frame can be selected as the mirror frame.
Frame No. (used only for "FRAME" method)	When User frame or Jog frame is selected, this indicates which frame number will be used.
Mirror Plane (used only for "FRAME" method)	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.
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.

9.42.1 Parallel Mirror Image

A parallel mirror image mirrors the program about a mirror plane without an offset or a rotation. See Fig. 9.42.1 (a) and Fig. 9.42.1 (b).

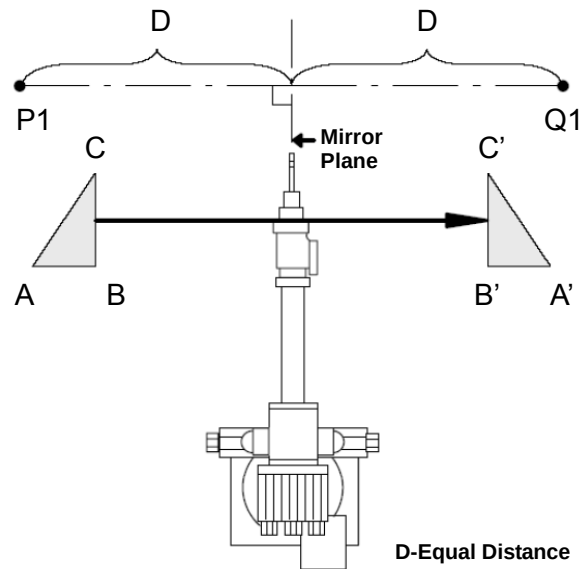


Fig. 9.42.1 (a) Parallel Mirror Image with Mirror Plane in Center of Robot

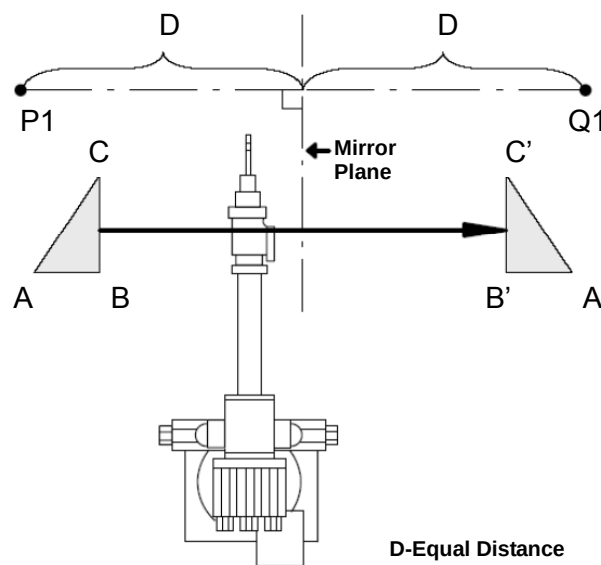


Fig. 9.42.1 (b) Parallel Mirror Image with Mirror Plane Offset from Center of Robot



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 Fig. 9.42.1 (c).

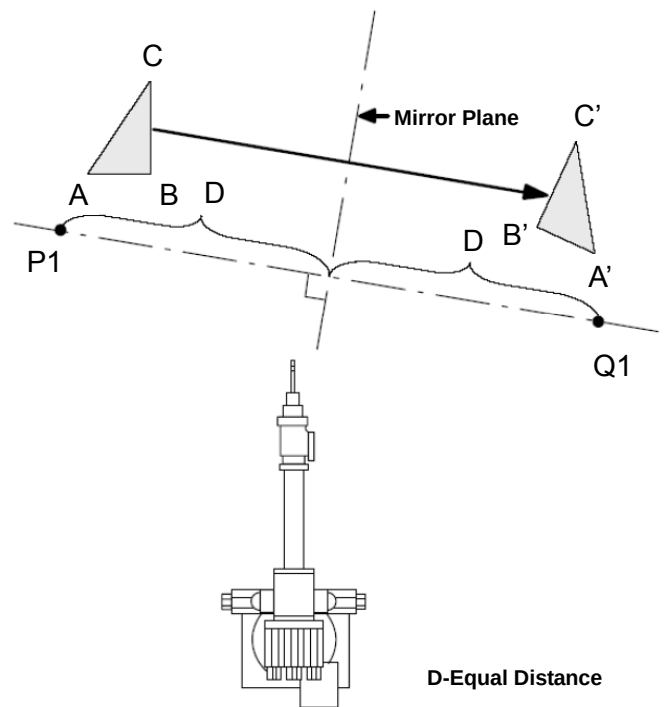


Fig. 9.42.1 (c) Parallel Mirror Image with Offset

Mirror Image Example

Fig. 9.42.1 (d) displays a mirror image of A, B, and C to A', B', and C' when P1 and Q1 are taught positions. Fig. 9.42.1 (d) 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''.

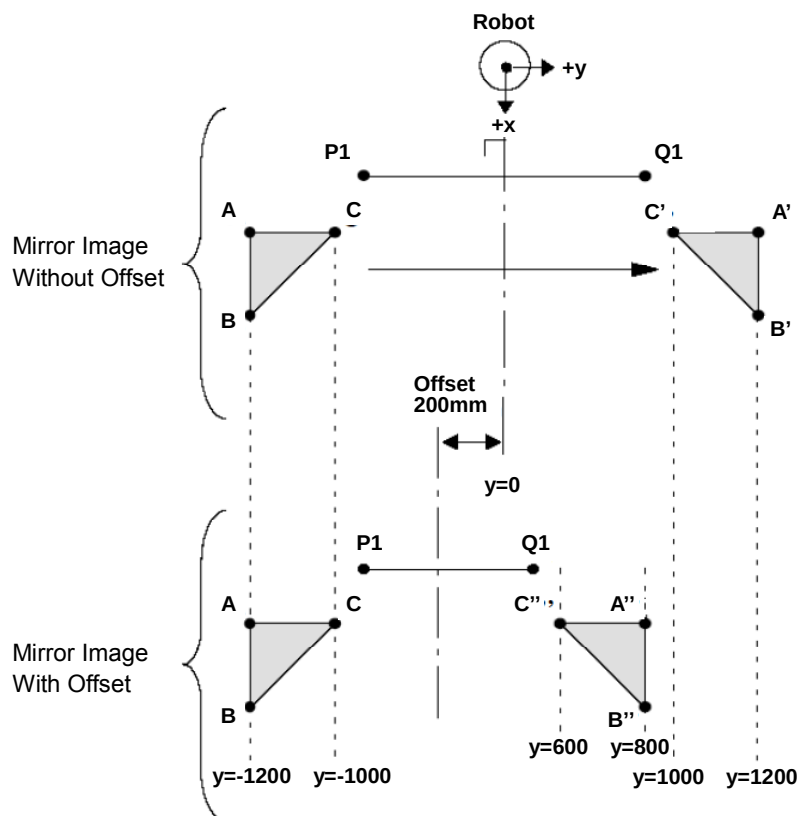


Fig. 9.42.1 (d) Positional Mirror Image

9.42.2 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 Fig. 9.42.2 (a).

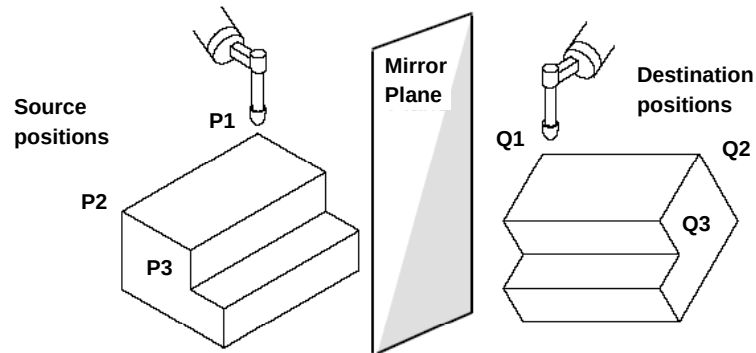


Fig. 9.42.2 (a) Rational Mirror Image

In Fig. 9.42.2 (b) 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'.

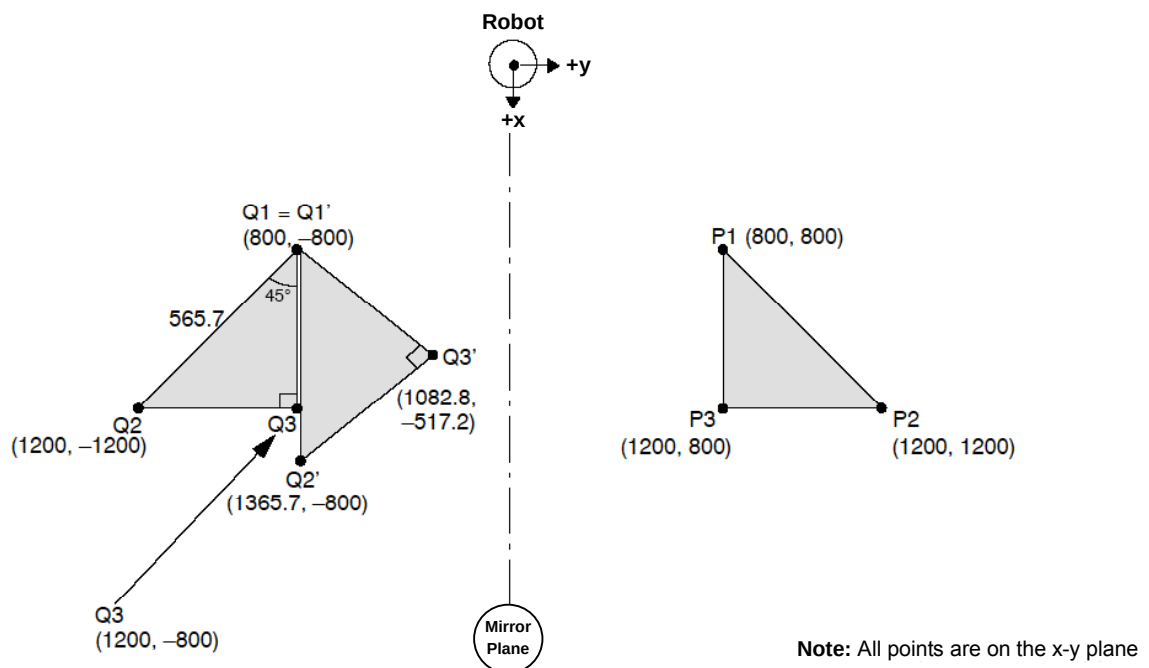


Fig. 9.42.2 (b) Rational Mirror Image

9.42.3 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 Fig. 9.42.3 (a).

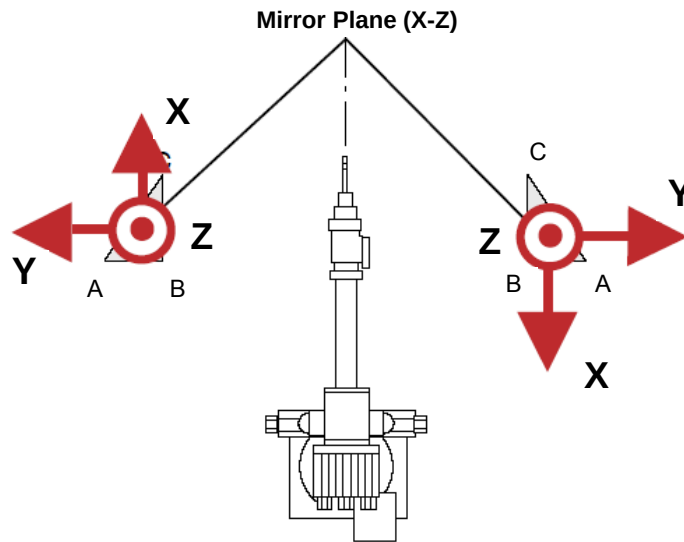


Fig. 9.42.3 (a) Mirror Image Shift with Orientation Mirrored (Mirror Method)

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 Fig. 9.42.3 (a), or "Fixture", as shown in Fig. 9.42.3 (b).

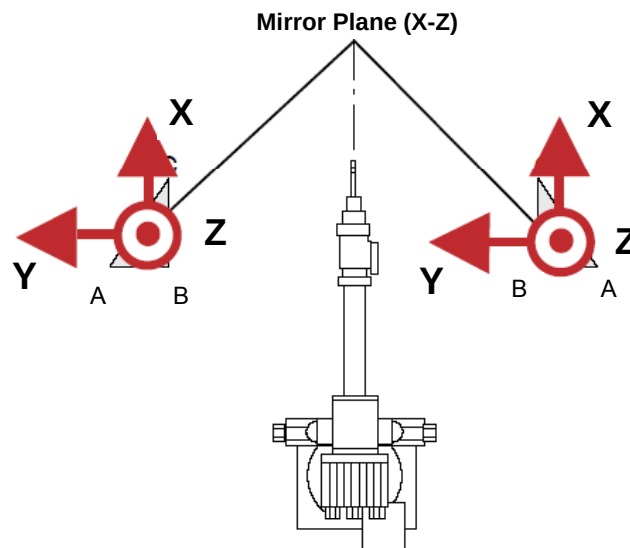


Fig. 9.42.3 (b) Enhanced Mirror Image with Orientation Controlled (Fixture Method)

9.42.4 Mirror Image of Extended Axes

Extended (Ext) axes determine how the mirror image function will translate the program when you are using extended axes. Fig. 9.42.4 (a) displays a key for use with Fig. 9.42.4 (b) through Fig. 9.42.4 (d).

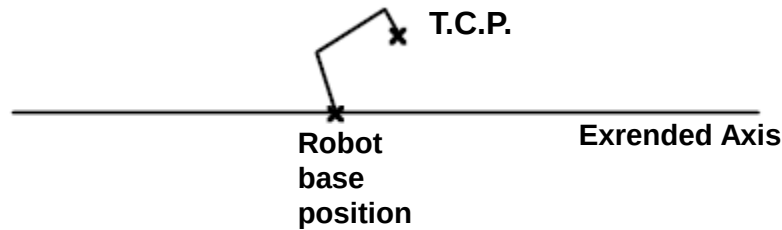


Fig. 9.42.4 (a) Mirror Image Key

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). See Fig. 9.42.4 (b).

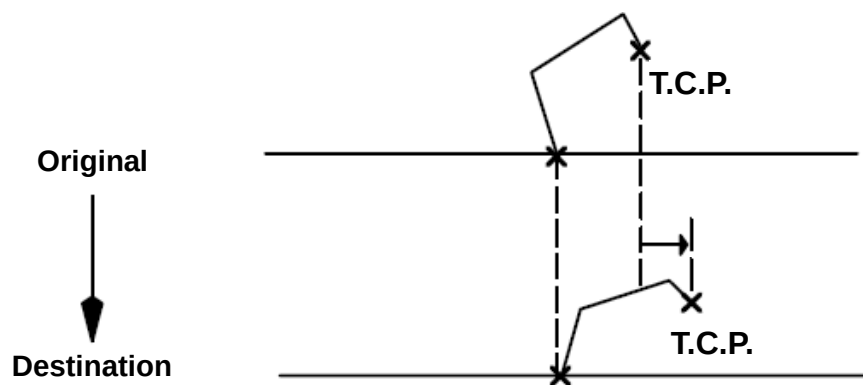


Fig. 9.42.4 (b) Example of Robot Axes Only Mirror Image

- **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 Fig. 9.42.4 (c).

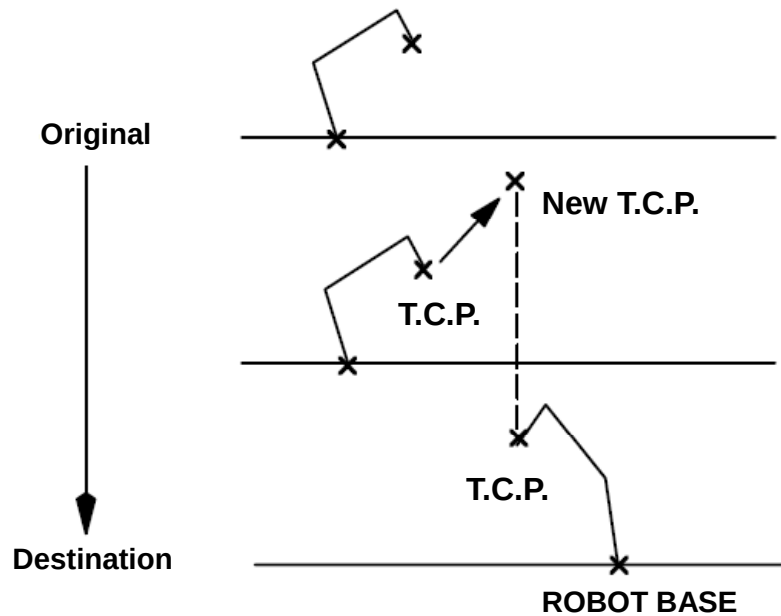


Fig. 9.42.4 (c) Example of Extended Axes Integrated Mirror Image

- **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 Fig. 9.42.4 (d).

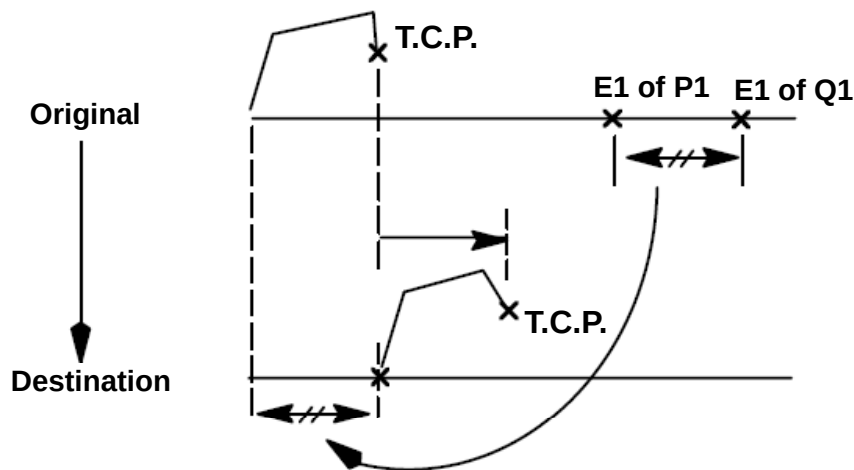


Fig. 9.42.4 (d) Example of With Extended Axes Mirror Image

Procedure 9-48 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.

Steps

1. Press MENUS.
2. Select UTILITIES.
3. Press F1, [TYPE].
4. Select Mirror Image Shift. You will see the MIRROR IMAGE SHIFT screen.

MIRROR IMAGE SHIFT		JOINT 10%
Settings		1/9
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		JOINT 10%
Settings		1/9
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

**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		JOINT 10%
Settings		1/9
Original Program :		
1		
2	Range:	PART
3	Start line:(not used)	0
4	End line:(not used)	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 Fig. 9.40.4 (e) and Fig. 9.40.4 (f).

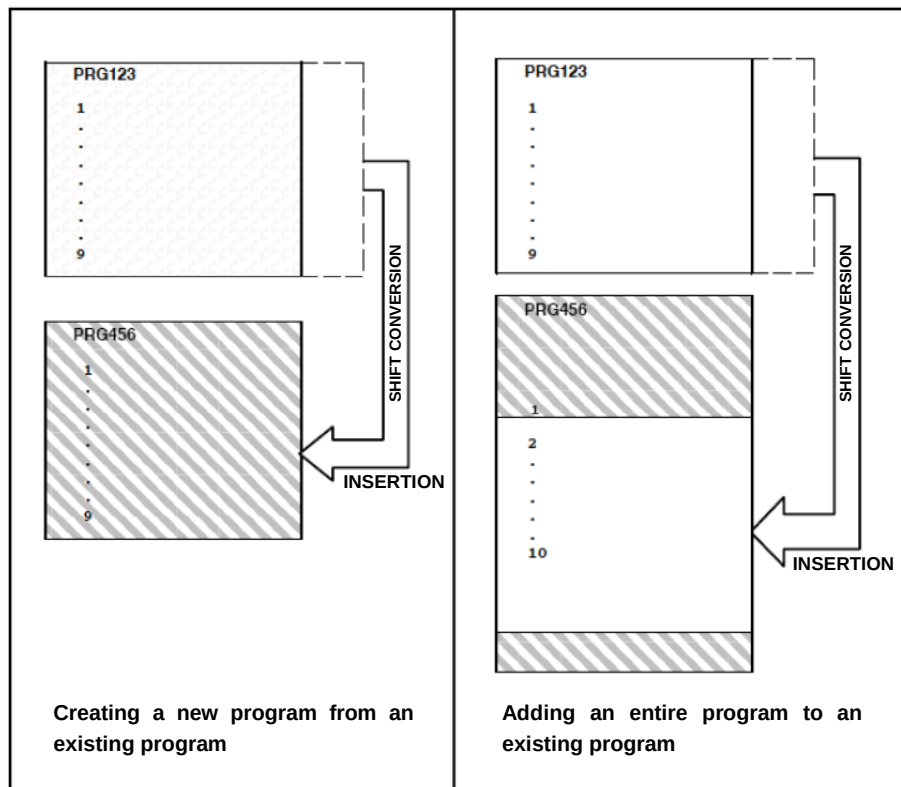


Fig. 9.42.4 (e) Mirroring an Entire Program

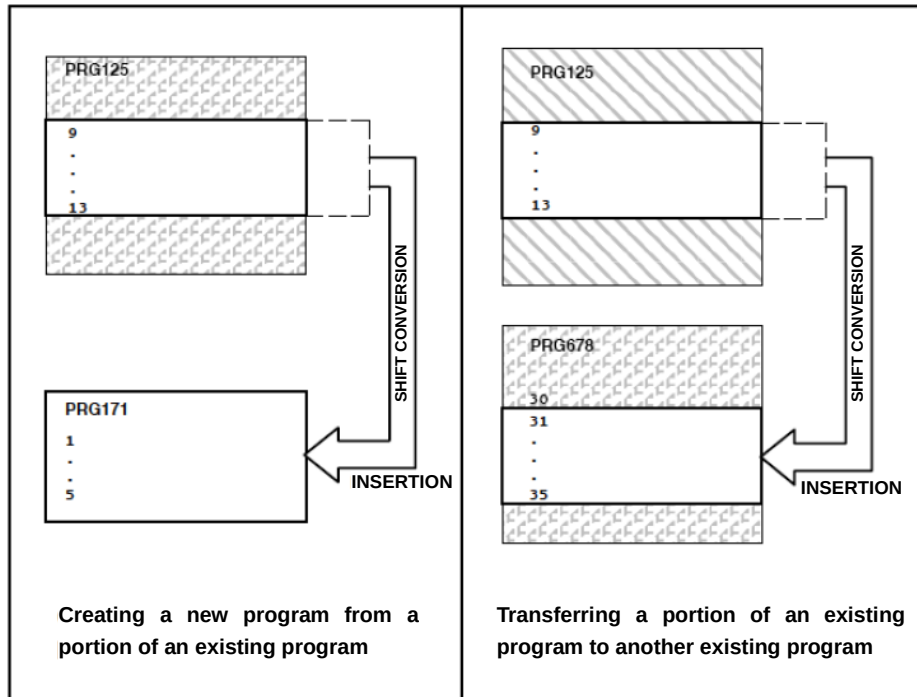


Fig. 9.42.4 (f) Mirroring a Portion of a Program

**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].
- **To mirror the image by teaching points to define the mirror plane**, press F5, [TEACH]. This method also mirrors the orientation.

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		JOINT 10%
Shift amount/Teach		1/3
Position data		
X :*****	Y :*****	Z :*****
1 Rotation		OFF
2 Source position		P1:
3 Destination position		Q1:

If you do not want to rotate the positions, continue to Step 12.

If you want to rotate the positions, continue to Step 13.

- **If you selected FRAME** for the Image Shift Method, you will see a Shift Frame Settings screen similar to the following:

MIRROR IMAGE SHIFT		JOINT 10%
Shift Frame Settings		1/3
1 Mirror Frame:		WORLD
2 Frame No. : (not used)		****
3 Mirror Plane:		XY

Continue to Step 14.

12. If you do not want to rotate the positions,

- Press F5, [OFF]. You will see a screen similar to the following.

MIRROR IMAGE SHIFT		JOINT 10%
Shift amount/Teach		1/3
Position data		
X :*****	Y :*****	Z :*****
1 Rotation		OFF
2 Source position		P1:
3 Destination position		Q1:

- Move the cursor to source position (see Fig. 9.40.1 (a)).
- 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. You will see a screen similar to the following.

MIRROR IMAGE SHIFT		JOINT 10%	
Shift amount/Teach		1/3	
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.

13. If you want to rotate the positions,

- a. Press F4, [ON]. You will see a screen similar to the following.

MIRROR IMAGE SHIFT		JOINT 10%	
Shift amount/Teach		1/3	
Position data			
X :*****	Y :*****	Z :*****	
1 Rotation		ON	
2 Source position		P1:	
		P2:	
		P3:	
3 Destination position		Q1:	
		Q2:	
		Q3:	

- b. Move the cursor to source position (see Fig. 9.42.1 (a)).
- 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. You will see a screen similar to the following.

```

MIRROR IMAGE SHIFT          JOINT 10%
Shift amount/Teach          1/3
Position data
X :***** Y :***** Z :*****

1 Rotation                  ON

2 Source position           P1:  P[1]
                           P2:  P[2]
                           P3:  P[3]

3 Destination position      Q1:
                           Q2:
                           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.

14. If you selected FRAME as the Image Shift Method, you will see a screen similar to the following:

```

MIRROR IMAGE SHIFT          JOINT 10%
Shift Frame Settings        1/3

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.

15. If you selected USER or JOG, move the cursor to Frame No. and type the number of the existing frame you want to use.
16. Move the cursor to Mirror Plane and press F4, [CHOICE]. Select the plane you want to use and press ENTER.
17. **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].

18. Wait until software has finished processing the mirror image.

- PROCESSING -

If the turn number will be changed, you will see a message similar to the following.

Select P[1]:J6 angle (deg -234)
 deg -234 deg 486 *Uninit* QUIT

The above example will be displayed when J6 turn number will be changed.

- If you do not want to change the turn number, press F1, [deg -234].
- If you want to change the turn number, press F2, [deg 486].
- If you want to make the position as un-initialized, press F3, [*Uninit*].
- If you want to interrupt mirror image, press F4, [QUIT].



CAUTION

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 reattach each position that did not mirror correctly.

9.43 KAREL USE SUPPORT FUNCTION

This section describes about KAREL Use Support Function, option J971.

9.43.1 Overview

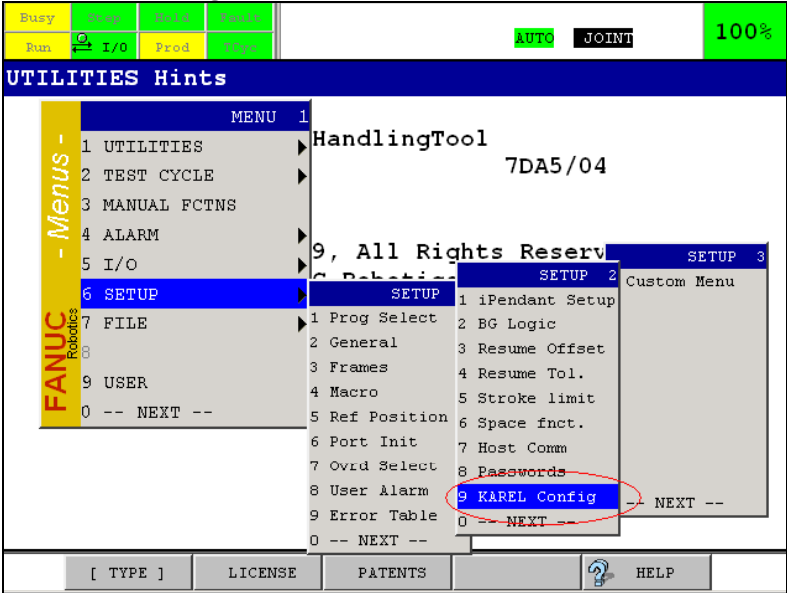
This function supplies the interface for KAREL execution, abort, status output and so on. It enables following operations.

- Executing or aborting KAREL programs.
- Auto running the KAREL programs at the time of power on.
- Periodically monitor the specified KAREL program and outputting the status.
- Set the configuration of custom menu setting.

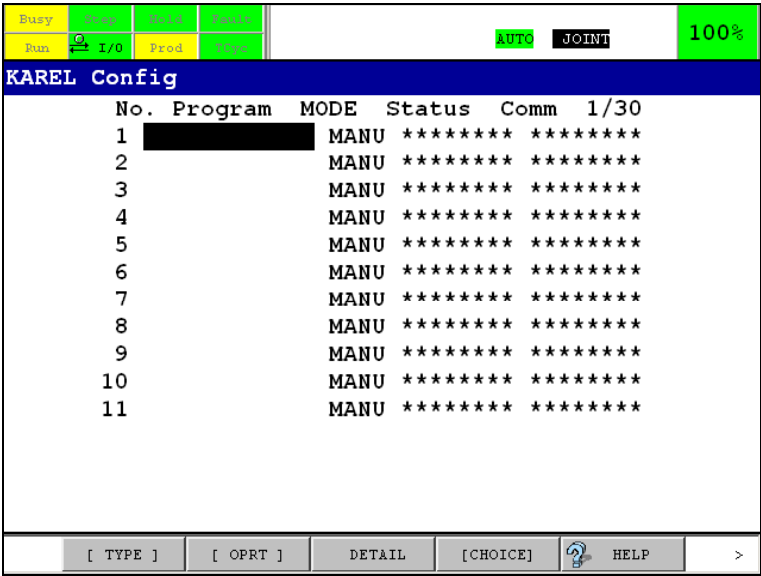
9.43.2 KAREL Config

9.43.2.1 Start the KAREL Config screen

Select menu – setup – KAREL Config.



Like following screen is displayed.



9.43.2.2 Use KAREL Config screen

Select F4[CHOICE], then you can see the list of KAREL program to be registered to run.

Busy	Scan	Hold	Pause	AUTO		JOINT	100%
Run	I/O	Prod	Stop				
KAREL Config							
No. Program MODE Status Comm 1/30							
1		MANU	*****	*****			
1		MANU	*****	*****			
1 C1		MANU	*****	*****			
2 C10		MANU	*****	*****			
3 C11		MANU	*****	*****			
4 C12		MANU	*****	*****			
5 C13		MANU	*****	*****			
6 C14		MANU	*****	*****			
7 C15		MANU	*****	*****			
8 --next page--		MANU	*****	*****			
10		MANU	*****	*****			
11		MANU	*****	*****			
[TYPE] [OPRT] DETAIL [CHOICE] ? HELP >							

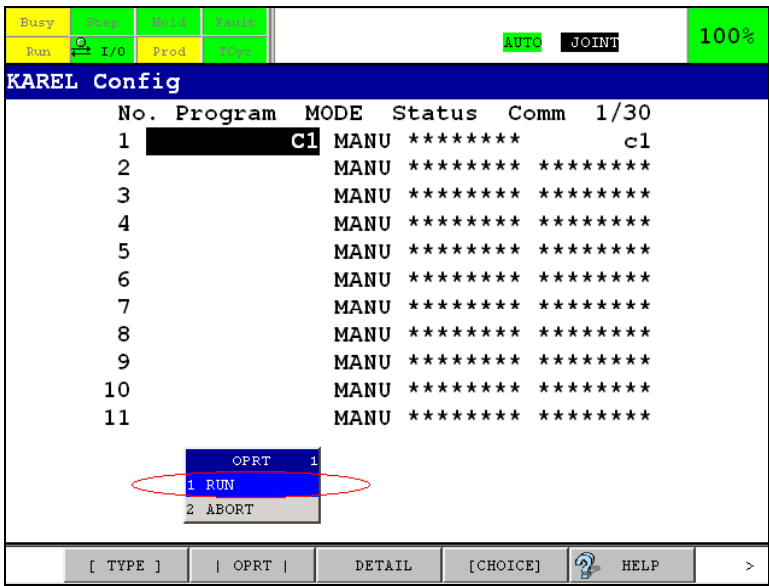
In this section, C1 is used as the sample to explain. Set the cursor onto the C1, and press enter key.

Busy	Run	Hold	Pause		AUTO	JOINT	100%
Run	I/O	Prod	Stop				
KAREL Config							
No. Program MODE Status Comm 1/30							
1	c1	MANU	*****	c1			
2		MANU	*****	*****			
3		MANU	*****	*****			
4		MANU	*****	*****			
5		MANU	*****	*****			
6		MANU	*****	*****			
7		MANU	*****	*****			
8		MANU	*****	*****			
9		MANU	*****	*****			
10		MANU	*****	*****			
11		MANU	*****	*****			
[TYPE] [OPRT] DETAIL [CHOICE] ? HELP >							

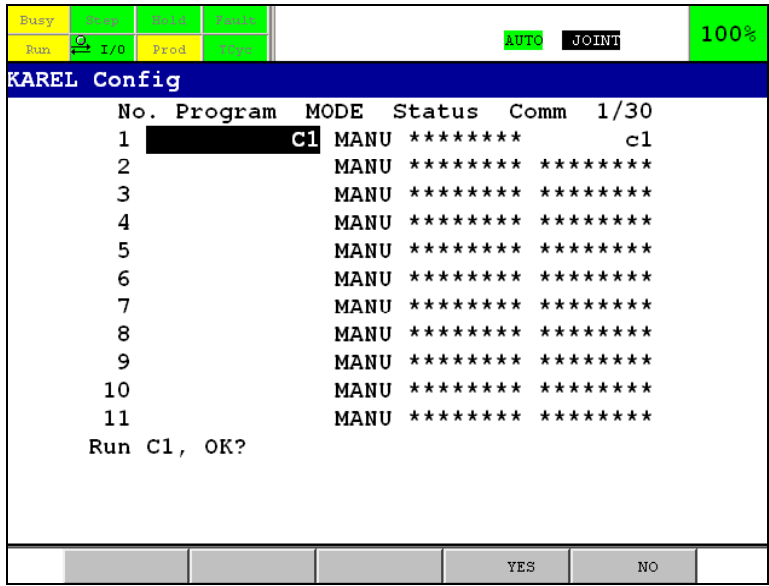
There are four status, running, paused, aborted and *****. ***** means that status is not other 3 ones.
Comm is the comment that is described in the KAREL source file as %COMMENT.

9.43.2.3 Run KAREL program

Set the cursor onto C1 in Program row and press F2[OPRT].



Select 1. RUN. Then confirmation message is displayed.



Program is run after selecting F4[YES].

Busy	Comp	Hold	Ready			AUTO	JOINT	100%
Run	I/O	Prod	Stop					

KAREL Config

```

No. Program  MODE  Status  Comm  1/30
1  C1  MANU  RUNNING  c1
2  MANU  *****  *****
3  MANU  *****  *****
4  MANU  *****  *****
5  MANU  *****  *****
6  MANU  *****  *****
7  MANU  *****  *****
8  MANU  *****  *****
9  MANU  *****  *****
10 MANU  *****  *****
11 MANU  *****  *****

C1 was run

```

[TYPE]	[OPRT]	DETAIL	[CHOICE]	 HELP	>

9.43.2.4 Abort the program

Set the cursor onto C1 in Program row and press F2[OPRT].

Busy	Stop	Hold	Ready				
Run	I/O	Prod	Rec		AUTO	JOINT	100%

KAREL Config

No.	Program	MODE	Status	Comm	1/30
1	c1	MANU	RUNNING		c1
2		MANU	*****	*****	
3		MANU	*****	*****	
4		MANU	*****	*****	
5		MANU	*****	*****	
6		MANU	*****	*****	
7		MANU	*****	*****	
8		MANU	*****	*****	
9		MANU	*****	*****	
10		MANU	*****	*****	
11		MANU	*****	*****	

OPRT	1
1 RUN	
2 ABORT	

[TYPE]	OPRT	DETAIL	[CHOICE]	HELP	>

Set the cursor onto 2. **ABORT** and press enter key.
Confirmation message is displayed.

Busy	Stop	Hold	Reset			100%
Run	I/O	Prod	Stop	AUTO	JOINT	
KAREL Config						
No.	Program	MODE	Status	Comm	1/30	
1	C1	MANU	RUNNING	c1		
2		MANU	*****	*****		
3		MANU	*****	*****		
4		MANU	*****	*****		
5		MANU	*****	*****		
6		MANU	*****	*****		
7		MANU	*****	*****		
8		MANU	*****	*****		
9		MANU	*****	*****		
10		MANU	*****	*****		
11		MANU	*****	*****		
Abort C1, OK?						
				YES	NO	

Select F4[YES], then KAREL program is aborted.

Busy	Stop	Hold	Reset			100%
Run	I/O	Prod	Stop	AUTO	JOINT	
KAREL Config						
No.	Program	MODE	Status	Comm	1/30	
1	C1	MANU	ABORTED	c1		
2		MANU	*****	*****		
3		MANU	*****	*****		
4		MANU	*****	*****		
5		MANU	*****	*****		
6		MANU	*****	*****		
7		MANU	*****	*****		
8		MANU	*****	*****		
9		MANU	*****	*****		
10		MANU	*****	*****		
11		MANU	*****	*****		
C1 was aborted						
[TYPE]		[OPRT]		DETAIL	[CHOICE]	HELP >

9.43.2.5 Start mode config

Set the cursor onto MODE row. F4 and F5 keys are turns to AUTO and MANUAL

Busy	Stop	Hold	Reset	AUTO JOINT		100%
Run	 I/O	Prod	Clear			

KAREL Config

No.	Program	MODE	Status	Comm	1/30
1	c1	MANU	ABORTED		c1
2		MANU	*****	*****	
3		MANU	*****	*****	
4		MANU	*****	*****	
5		MANU	*****	*****	
6		MANU	*****	*****	
7		MANU	*****	*****	
8		MANU	*****	*****	
9		MANU	*****	*****	
10		MANU	*****	*****	
11		MANU	*****	*****	

[TYPE]	[OPRT]	DETAIL	AUTO	MANUAL

Default is MANUAL. If you select AUTO, KAREL program is run after power on the controller

Busy	Setup	Hold	Ready			AUTO	JOINT	100%
Run	I/O	Prod	Stop					

KAREL Config

No.	Program	MODE	Status	Comm	1/30
1	C1	AUTO	ABORTED		c1
2		MANU	*****	*****	
3		MANU	*****	*****	
4		MANU	*****	*****	
5		MANU	*****	*****	
6		MANU	*****	*****	
7		MANU	*****	*****	
8		MANU	*****	*****	
9		MANU	*****	*****	
10		MANU	*****	*****	
11		MANU	*****	*****	

C1 is run after cycle power

[TYPE]	[OPRT]	DETAIL	AUTO	MANUAL

There is a limit that the number of running the KAREL simultaneously. It depends on the configuration of the robot system. “PROG-014 Max task number exceed” will be posted when the number of running KAREL is going to be over max number.

9.43.2.6 Detail of KAREL Config

Press F3[DETAIL].

Busy	Setup	Hold	Reset	AUTO		JOINT	100%																																																																								
Run		I/O	Prod	Clear																																																																											
KAREL Config																																																																															
<table><tr><td>No.</td><td>Program</td><td>MODE</td><td>Status</td><td>Comm</td><td>1/30</td></tr><tr><td>1</td><td>c1</td><td>MANU</td><td>ABORTED</td><td></td><td>c1</td></tr><tr><td>2</td><td></td><td>MANU</td><td>*****</td><td>*****</td><td></td></tr><tr><td>3</td><td></td><td>MANU</td><td>*****</td><td>*****</td><td></td></tr><tr><td>4</td><td></td><td>MANU</td><td>*****</td><td>*****</td><td></td></tr><tr><td>5</td><td></td><td>MANU</td><td>*****</td><td>*****</td><td></td></tr><tr><td>6</td><td></td><td>MANU</td><td>*****</td><td>*****</td><td></td></tr><tr><td>7</td><td></td><td>MANU</td><td>*****</td><td>*****</td><td></td></tr><tr><td>8</td><td></td><td>MANU</td><td>*****</td><td>*****</td><td></td></tr><tr><td>9</td><td></td><td>MANU</td><td>*****</td><td>*****</td><td></td></tr><tr><td>10</td><td></td><td>MANU</td><td>*****</td><td>*****</td><td></td></tr><tr><td>11</td><td></td><td>MANU</td><td>*****</td><td>*****</td><td></td></tr></table>								No.	Program	MODE	Status	Comm	1/30	1	c1	MANU	ABORTED		c1	2		MANU	*****	*****		3		MANU	*****	*****		4		MANU	*****	*****		5		MANU	*****	*****		6		MANU	*****	*****		7		MANU	*****	*****		8		MANU	*****	*****		9		MANU	*****	*****		10		MANU	*****	*****		11		MANU	*****	*****	
No.	Program	MODE	Status	Comm	1/30																																																																										
1	c1	MANU	ABORTED		c1																																																																										
2		MANU	*****	*****																																																																											
3		MANU	*****	*****																																																																											
4		MANU	*****	*****																																																																											
5		MANU	*****	*****																																																																											
6		MANU	*****	*****																																																																											
7		MANU	*****	*****																																																																											
8		MANU	*****	*****																																																																											
9		MANU	*****	*****																																																																											
10		MANU	*****	*****																																																																											
11		MANU	*****	*****																																																																											
[TYPE]		[OPRT]		DETAIL		[CHOICE]	HELP																																																																								
							>																																																																								

Like following screen is displayed.

Busy	Stop	Reset	Pause		AUTO	JOINT	100%
Run	I/O	Prod	Clear				

KAREL Config

1/6

Program: [C1]

 1 No. **1**

 2 Paused alarm DISABLE

 3 Start Config Restart

 4 Start DI DISAB DI[0] +

 5 Abort DI DISAB DI[0] +

 6 Status DISAB DO[0] DO[0]

 Running ON ON

 Paused ON OFF

 Aborted OFF ON

[TYPE]	PREV	NEXT	?	HELP	>

If “2 Paused alarm” is enable, warning is posted when specified program, ex C1, is paused or aborted.

In the case of paused, “SYST 294 Paused (program name)” and “SYST 296 Specified karel was paused” are posted.

In the case of aborted, “SYST 295 Aborted (program name)” and “SYST 297 Specified karel was aborted” are posted.

“3 Start Config” set the mode of restart. “Restart” is once aborting the specified program and start from the top of the program. “Resume” is not once aborted. Specified program is restarted from the paused point.

“4 Start DI ” sets the start operation. Detecting the rising edge (+) or falling edge (-) of specified DI starts the specified program. When “Start DI” is set enable, DI port number and Up or Down can not be

changed. If you would like to set them, please set “Start DI” disable once. After setting them, please set it enable. Then this configuration is enabled after cycle power.

“5 Abort DI” sets the abort operation. How to set is the same of “4 Start DI”.

“6 Status” is to monitor the specified program and output its status with two DOs. As the default, program is monitored every 100msec. Its config is enabled after cycle power. Following is the sample of configuration.

Busy	Run	Pause	Abort		AUTO	JOINT	100%
Run	I/O	Prod					

KAREL Config 6/6

Program: [C1]

1 No. 1

2 Paused alarm ENABLE

3 Start Config Restart

4 Start DI ENABL DI[1] +

5 Abort DI ENABL DI[2] +

6 Status **ENABL** DO[1] DO[2]

Running	ON	ON
Paused	ON	OFF
Aborted	OFF	ON

[TYPE]	PREV	NEXT	ENABLE	DISABLE

On this sample, DO[1] and DO[2] are on while program C1 is running. DO[1] is on and DO[2] is off while it is pausing.

DO[1] is off and DO[2] is on while aborting. If status is not each three status, DO[1] and DO[2] are both off.

To switch previous screen, press F2[PREV]. In this case, screen is switched to No.30.

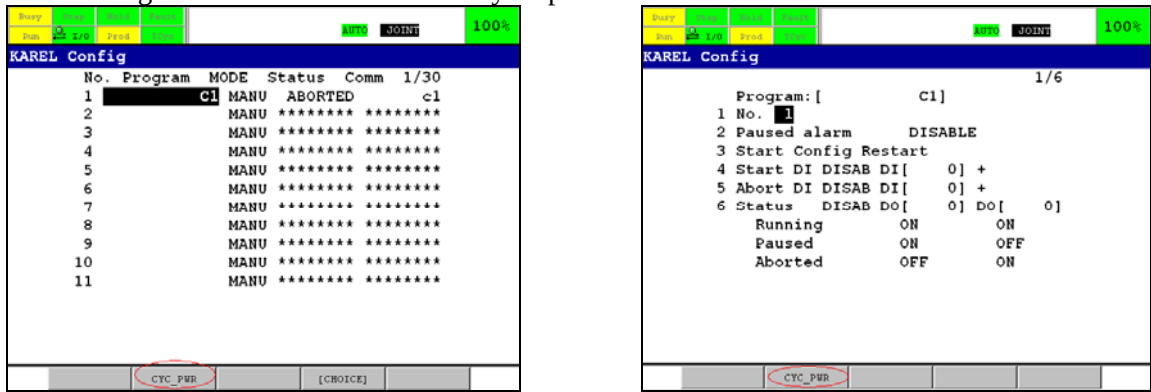
To switch next detail screen, press F3[NEXT]. In this case, screen is switched to No.2.

9.43.2.7 Limitation and Caution of KAREL Config

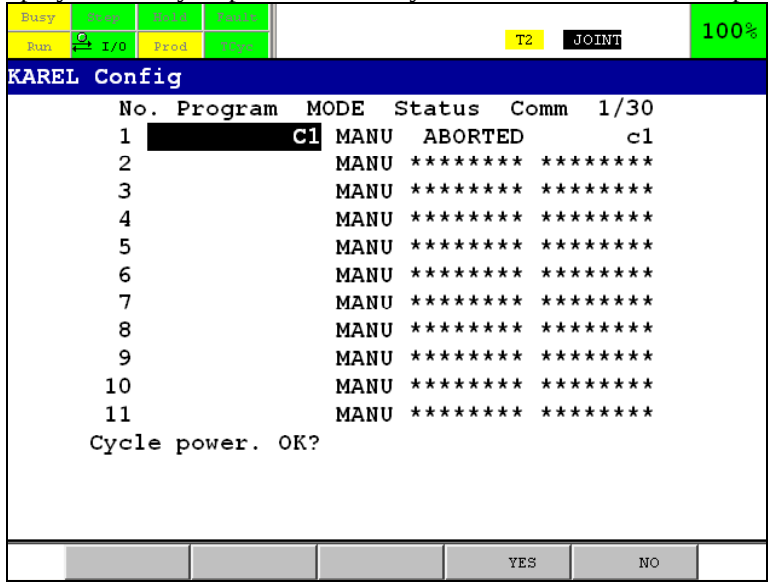
- Status is monitored at a constant frequency. Thus status is not changed at a real time. For example, program status is changed from running to paused, and aborted in one monitoring cycle, paused status is not detected at next monitoring cycle. Therefore status is changed from running to aborted in this case.
- Monitoring constant frequency can be changed. You can change it with system variable \$KRL_NUM.\$UPDT_TIME. It is able to change in the range 8 to 5000 msec. If you set it out of its range under 8 or above 5000, it is automatically set to 8 or 5000msec.
- In the case of above sample, after you set the detail of C1, return to KAREL Config screen with prev key and change the registered No.1 program from C1 to other program, such as C2, C1 does not start if DI[1] is turned on. C2 is run instead. For prevention of miss operation, please do not change program after set the detail.

9.43.2.8 Cycle Power

Both kare config and detail screen enable to cycle power.



F2[CYC_PWR] is displayed when you press NEXT key. Set the TP to enable and press F2[CYC_PWR].

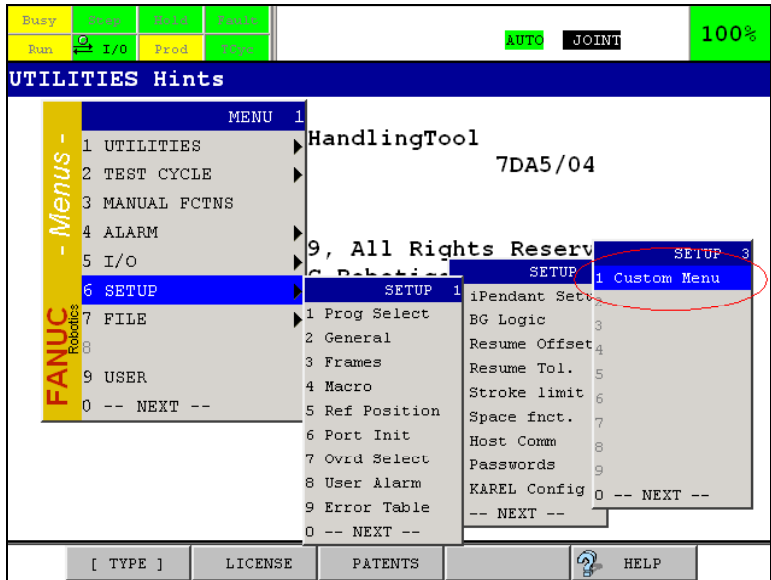


If F4[YES] is selected, the warning message “SYST-266 Please cycle power” is displayed. Please cycle power.

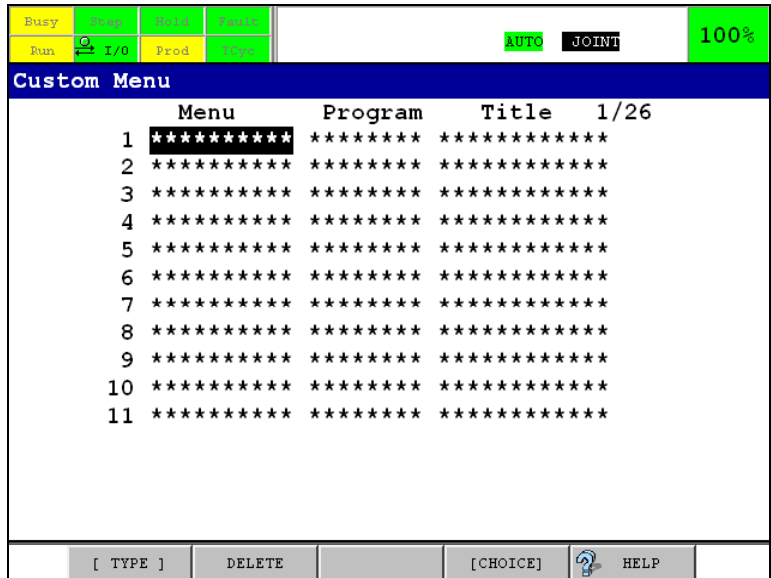
9.43.3 Custom Menu

9.43.3.1 Starting custom menu

Select menu – setup – Custom Menu.



Following is displayed.



9.43.3.2 Set custom menu

Press F4[CHOICE].

Busy	Run	Prod	100%
Run	I/O	Prod	

Custom menu

Menu

Program

426

1 ***

1 UTILITIES1

2 UTILITIES2

3 TEST CYCLE

4 MANUAL-1

5 MANUAL-2

6 ALARM-1

7 ALARM-2

8 --next page--

2 I/O-1

3 I/O-2

4 SETUP-1

5 SETUP-2

6 FILE

7 DATA-1

8 DATA-2

9 --next page--

1 STATUS-1

2 STATUS-2

3 STATUS-3

4 STATUS-4

5 SYSTEM

6 --next page--

10 *****

11 *****

[TYPE]

DELETE

[CHOICE]

HELP

Select item that you would like to add. Select FCTN -1 as an example.

Busy	Run	Prod	100%
Run	I/O	Prod	

Custom Menu

Menu

Program

Title

1/26

1 FCTN-1

2 *****

3 *****

4 *****

5 *****

6 *****

7 *****

8 *****

9 *****

10 *****

11 *****

[TYPE]

DELETE

[CHOICE]

HELP

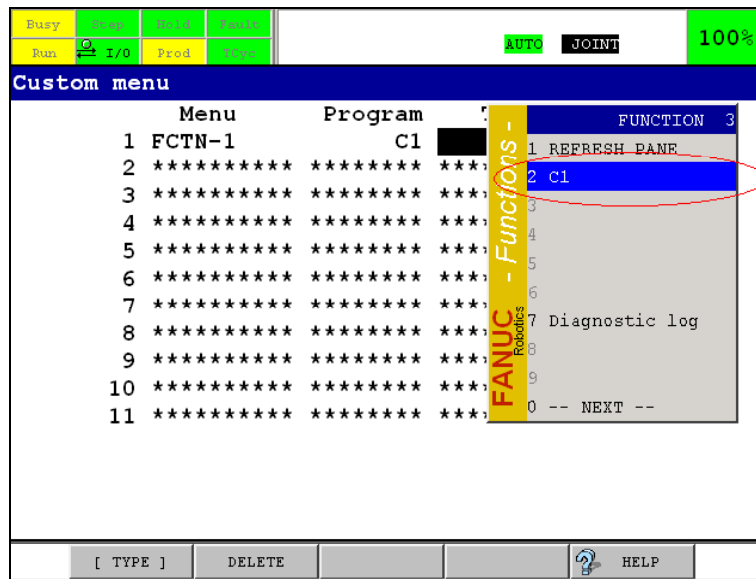
Set the cursor onto program row and press F4[CHOICE] key. As an example C1 is selected.

Busy	Scan	Hold	Ready			100%
Run	I/O	Prod	Stop	AUTO JOINT		
Custom Menu						
Menu		Program	Title	1/26		
1 FCTN-1		*****	*****			
1 C1		*****	*****			
2 C10		*****	*****			
3 C11		*****	*****			
4 C12		*****	*****			
5 C13		*****	*****			
6 C14		*****	*****			
7 C15		*****	*****			
8 --next page--		*****	*****			
10 *****		*****	*****			
11 *****		*****	*****			
[TYPE]		DELETE		[CHOICE]	?	HELP

Next, set the cursor onto title row and press enter key. C1 is input for instance.

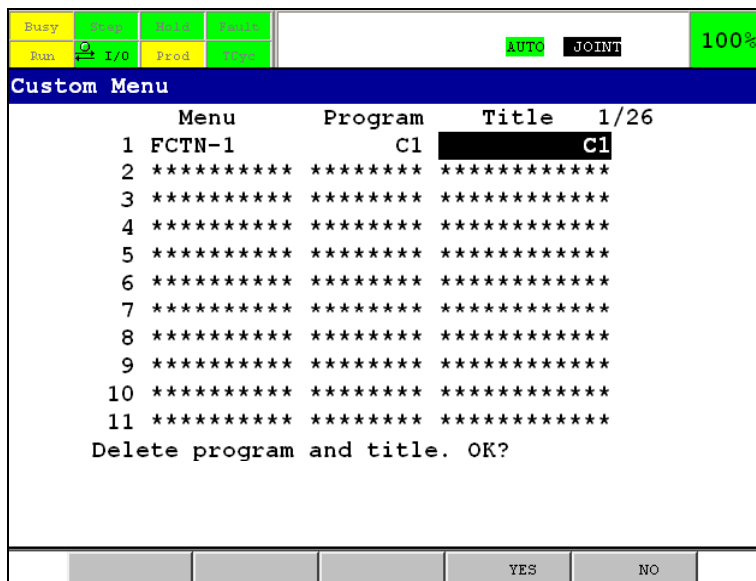
Busy	Scan	Hold	Ready			100%
Run	I/O	Prod	Stop	AUTO JOINT		
Custom Menu						
Menu		Program	Title	1/26		
1 FCTN-1		C1	C1			
2 *****		*****	*****			
3 *****		*****	*****			
4 *****		*****	*****			
5 *****		*****	*****			
6 *****		*****	*****			
7 *****		*****	*****			
8 *****		*****	*****			
9 *****		*****	*****			
10 *****		*****	*****			
11 *****		*****	*****			
Old Value:		Alpha input 1 Upper Case Lower Case Punctuation Options				
ABCDEF		GHIJKL	MNOPQR	STUVWX	YZ_@*.	

Select FCTN – 0 – NEXT -- - 0 –NEXT -- . Verify C1 is added to FCTN menu.



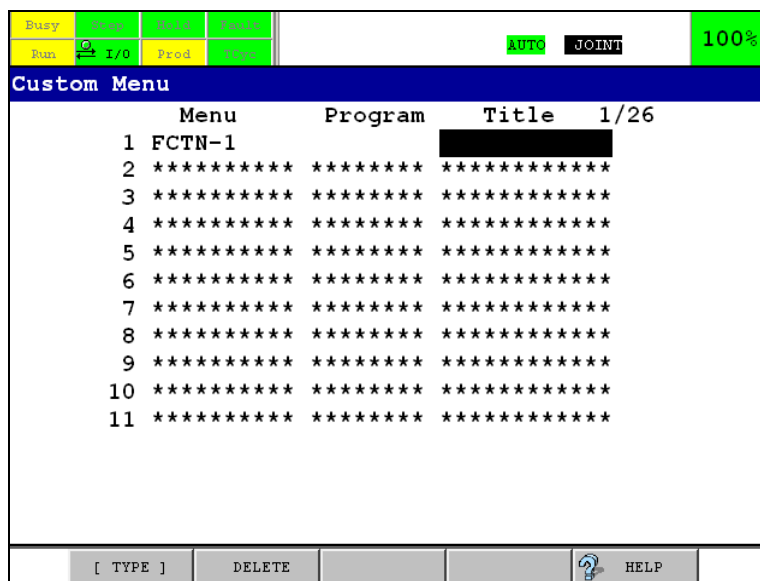
9.43.3.3 Delete set

In the case of deleting set, press F2[DELETE]. Confirmation message is displayed like following.

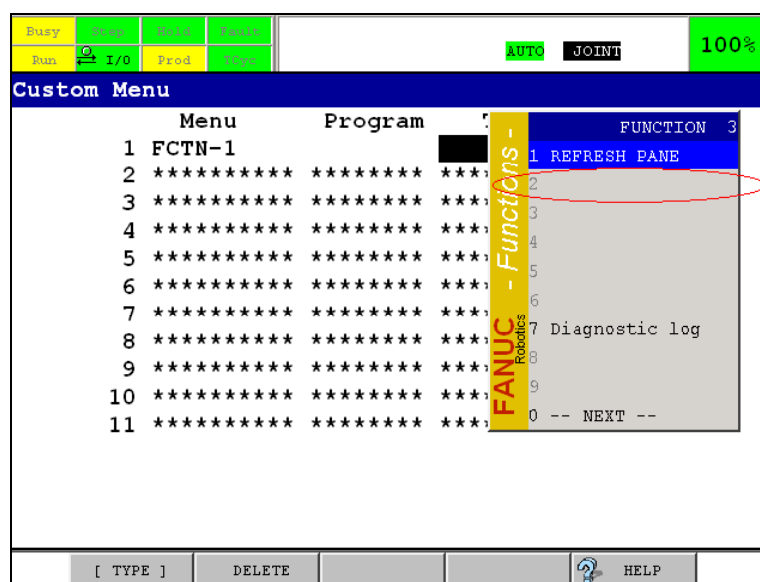


Select F4[YES].

Specified program and title are deleted.



Select FCTN – 0 – NEXT -- - 0 –NEXT -- . Verify C1 is deleted from FCTN menu.



9.44 KAREL PROGRAM EXECUTION HISTORY RECORD

9.44.1 Overview

KAREL program execution history record is a tool used to diagnose user application or controller system software problems. This option generates logs that show details about events, including the sequence in which they occur.

This option allows you to log certain events that occur during KAREL and TPP program execution. It also allows you to generate ASCII files that contain all or part of the logged data. The ASCII files contain detailed information about each event that is logged.

For example, you can log the following types of events.

- Start of execution of KAREL statements or TPP lines.

- Routine calls or returns.
- Internal system events for diagnosis of system software problems.

Refer to the following for additional information.

- For information about related hardware and software, refer to 9.44.2 Hardware And Software.
- For information about installing and setting up the debugging option, refer to 9.44.3 Setup And Operations.
- For information about events, refer to chapter 9.44.4 logging events.
- For examples of this option as related to KAREL programming, teach pendant programming, and the ASCII file format, refer to 9.44.5.

9.44.2 Hardware and Software

9.44.2.1 Hardware and software requirements

There are certain hardware and software requirements that must be met before you can run the KAREL Program Execution History Record.

NOTE

The hardware requirements in this section apply to all robot models.

9.44.2.2 Hardware

The KAREL Program Execution History Record requires the following hardware for all robot models:

- A FANUC Robot Controller
- A teach pendant
- 42,000 bytes of Temporary memory
- 18,000 bytes of CMOS
- 30000 bytes of FROM file system space

9.44.2.3 Software

The KAREL Program Execution History Recode Option can be installed in controllers with KAREL software version 7DA5 or later.

A series of teach pendant is used to control the logging of diagnostic information. The teach pendant screens allow you to:

- Specify the kinds of events you want to log.
- Specify the tasks for which events are to be logged.
- Generate ASCII files containing the logged data.

NOTE

For information about how to install the KAREL Program Execution History Record, refer to the Software Installation Manual.

9.44.2.4 Performance

The KAREL Program Execution History Record Option can affect the performance of your system.

- This option reduces execution speed of KAREL and teach pendant program logic by approximately 1 to 2 %.

NOTE

You can select “Disable all logging” on the Diagnostic Logging main menu to eliminate this reduction.

Refer to Subsection 9.44.3.8 for more information about system performance.

- Actual logging of events adds about .08 ms per event logged.
- Motion execution is not affected.

9.44.3 Setup and Operations

9.44.3.1 Setting up the KAREL program execution history record

Before you can log information and write data log files, you must set up the KAREL Program Execution History Record Option. This subsection contains information about how to set up the KAREL Program Execution History Record Option, including a detailed procedure for each setup screen.

The following screen are used to control diagnostic logging:

- Clear event log deletes all data from the log.
- Dump Selections screen allows you to write log data to an ASCII file, Subsection 9.44.3.2.
- Task Selection screen allows you to add tasks for which events are logged, Subsection 9.44.3.3.
- Stop Logging a task Selection screen allows you to select tasks to be removed from the list of currently logged tasks, Subsection 9.44.3.4.
- Currently Selected tasks screen lists tasks currently selected for logging, Subsection 9.44.3.5.
- Event Class Selections screen allows you to select or de-select classes or groups of related event types for logging, Subsection 9.44.3.6.
- Event Detail selections screen allows you to select or de-select individual event types for logging, Subsection 9.44.3.7.
- Enable or Disable all logging allows you to start or stop all event logging, Subsection 9.44.3.8.

For example, you can log the following types of events.

- Start of execution of KAREL statements or TPP lines.
- Routine calls or returns.
- Internal system events for diagnosis of system software problems.

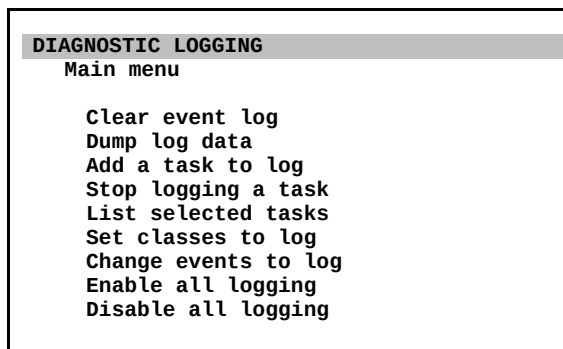
9.44.3.2 Dump selections screen

The Dump Selection Screen allows you to select a range of records to be dumped or logged. If the log is currently empty, an error message is displayed. Use Procedure 9-49 to dump records to the debug log.

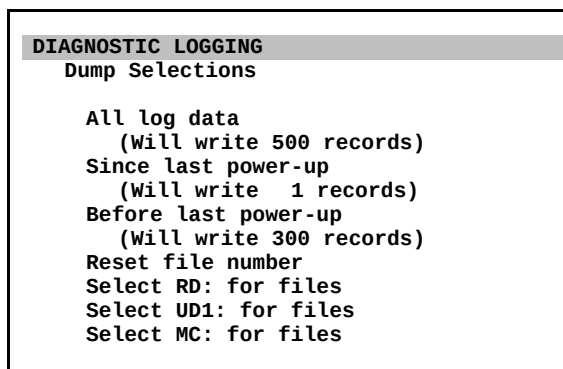
Procedure 9-49 Dumping Records to the Debug Log

Steps

1. Press MENU.
2. Select TEST CYCLE.
3. Press F1, [TYPE].
4. Select Debug Ctl, You will see a screen similar to the following.



5. Select Dump log data. The number of records in each range is displayed on the screen. See the following screen for an example.



6. Select the range of data to be dumped or logged.
 - To request a dump of all event information currently in the log , select All log data.
 - To request a dump of all event information logged since the most recent power-up, select Since last power-up.
 - To request a dump of all event information between the last two power-ups, select Before last power-up. This will log all events prior to the last power-down.

The information will be dumped to an ASCII file.

NOTE

By default, successive log files are written to MC:PGDBG201.DT, MC:PGDBG202.DT, and so forth.

7. To reset the file number in the file naming convention , select Reset file number. The starting point is MC:PGDBG201.DT.
8. To change the default saving device to RAM disk , select RD: for files. All files will be automatically saved to RAM disk.
9. To change the default saving device to UD1:, select UD1: for files. All files will automatically be saved to UD1:.

NOTE

UD1: is the device for R-30iA controller. Please do not select UD1: in R-30iA Mate controller.

10. To change the default saving device to memory card, select MC: for files. All files will automatically be saved to a memory card.

9.44.3.3 Task selection screen

The Task Selection screen allows you to select additional tasks for which you want to log information. Use Procedure 9-50 to select tasks to log.

Procedure 9-50 Selecting Tasks to Log

Steps

1. Press MENUS.
2. Select TEST CYCLE.
3. Press F1, [TYPE].
4. Select Debug Ctl. You will see a screen similar to the following.

```

DIAGNOSTIC LOGGING
Main Menu

Clear event log
Dump log data
Add a task to log
Stop logging a task
List selected tasks
Set classes to log
Change events to log
Enable all logging
Disable all logging
  
```

5. Select Add a task to log. You will see a screen similar to the following.

```

DIAGNOSTIC LOGGING
Task Selection                                1/1

Use CHOICE to select task to add
or PREV if none
  
```

6. To display a list of tasks from which you can select, press F4, [CHOICE]. You will see a screen similar to the following.

NOTE

Selecting a program means that events will be logged for a task that has the selected program as its MAIN program.

```

1 PRG001          5 SUB0012
2 PRG002          6 SUB0013
3 PRG003          7 SUB0021
4 SUB0011
DIAGNOSTIC LOGGING
Task selection
  
```

7. Use the arrow keys to select the task for which you want events logged.
8. Press ENTER when you have chosen a task. You will see a screen similar to the following.

```

DIAGNOSTIC LOGGING
Task selection                                1/1
Use CHOICE to select task to add,
or PREV if none

                                PROG003
  
```

9. To request logging for the displayed task, press F3, OK.

⚠ CAUTION

Do not exit this screen by pressing PREV, F1, [TYPE], or any other key before you press F3, OK; otherwise, the selected task will not be logged.

9.44.3.4 Stop Logging Tasks Screen

The Stop Logging Tasks screen allows you to select tasks to be removed from the list of currently logged tasks. Use Procedure 9-51 to stop logging tasks.

Procedure 9-51 Stop Logging Tasks

Steps

1. Press MENU.
2. Select TEST CYCLE.
3. Press F1, [TYPE].
4. Select Debug Ctl. You will see a screen similar to the following.

```

DIAGNOSTIC LOGGING
Main Menu

Clear event log
Dump log data
Add a task to log
Stop logging a task
List selected tasks
Set classes to log
Change events to log
Enable all logging
Disable all logging
  
```

5. Select Stop logging a task. You will see a screen similar to the following.

```

DIAGNOSTIC LOGGING
Stop Logging

Use CHOICE to select task not to log
or EXIT if none
  
```

6. To display a list of tasks, from which you can select, press F4, [CHOICE]. You will see a screen similar to the following.

NOTE

When you select a program, the events will no longer be logged for any tasks that have the selected program as their MAIN program.

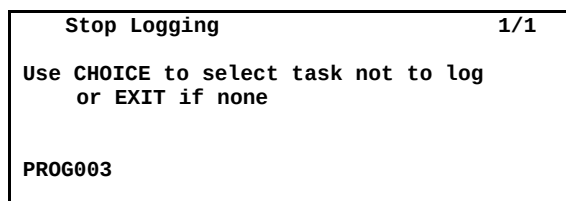
```

1 PRG001          5 SUB0012
2 PRG002          6 SUB0013
3 PRG003          7 SUB0021
4 SUB0011
DIAGNOSTIC LOGGING
Stop logging
  
```

7. Use the arrow keys to select the task for which you want to stop logging events.
8. Press ENTER when you have chosen a task. You will see a screen similar to the following.

```

DIAGNOSTIC LOGGING
  
```



9. To stop logging for the task that is current displayed, press F3, OK.

**CAUTION**

Do not exit this screen by pressing PREV, F1, [TYPE] , or any other key before you press F3, OK; otherwise, logging for the selected task will not stop.

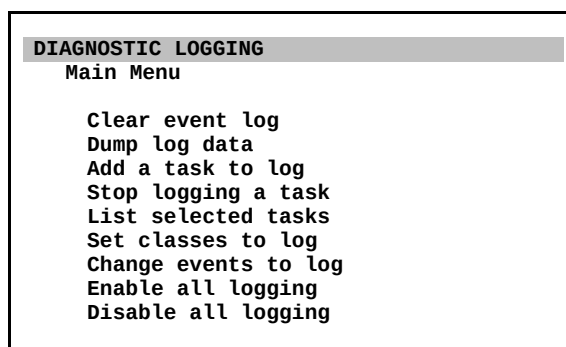
9.44.3.5 List selected tasks screen

The List Selected Tasks Screen allows you to displays the tasks that are currently selected and being logged. Use Procedure 9-52 to list selected tasks.

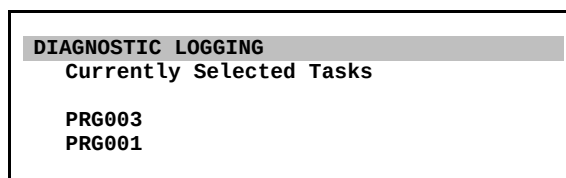
Procedure 9-52 List Selected Tasks

Steps

1. Press MENUS.
2. Select TEST CYCLE.
3. Press F1, [TYPE].
4. Select Debug Ctl. You will see a screen similar to the following.



5. Select List selected tasks. You will see a screen similar to the following.



This screen displays the tasks that are currently selected and being logged.

6. Press PREV on the teach pendant to return to the main menu.

9.44.3.6 Event class selection screen

The Event Class Selection Screen allows you to determine which classes of events should be logged (Procedure 9-55). Each class of events enables or disables one or more detailed event types. The following is a list of event classes and the associated event types. Refer to Subsection 9.44.4.1 for more detailed information on the types of events that can be logged.

NOTE

An asterisk (*) indicates logging is for internal use only.

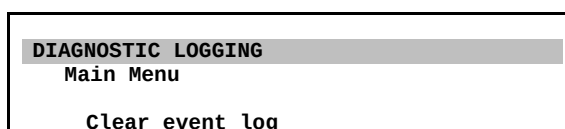
- Execute KAREL statement
 - Execute KAREL statement
- Execute TPP line
 - Execute TPP line
- Call/return
 - KAREL or TPP routine called
 - KAREL or TPP routine returned
- Motion *
 - Motion started *
 - Motion planned (teach pendant motion only) *
 - Motion cancel issued *
 - Motion stop issued *
 - Resume move *
 - Motion done received *
 - Motion completed normally *
 - MMR received*
- Condition Handler
 - Condition handler triggered
 - Condition handler enabled
 - Condition handler disabled
- Interrupt rtn
 - Before transferring to an ISR (Interrupt Sub-Routine)
 - After transferring to an ISR (Interrupt Sub-Routine)
 - Return from ISR (Interrupt Sub-Routine)
- Task start/end
 - KAREL or TPP task starts execution
 - KAREL or TPP task aborts
- Packet_rcd *
 - Interpreter receives packet *
- Pcode exec *
 - Starting execution of p-code *
- AMR activity *
 - Normal AMR recvd by AMGR *
 - Start AMR recvd by AMGR *
 - Stop AMR recvd by AMGR *
 - AMR sent to AX *
 - AMR rcvd from AX *

Use Procedure 9-53 to enable or disable event classes for logging.

Procedure 9-53 Enable or Disable Event Classes for Logging

Steps

1. Press MENU.
2. Select TEST CYCLE.
3. Press F1, [TYPE].
4. Select Debug Ctl. You will see a screen similar to the following.



```

Dump log data
Add a task to log
Stop logging a task
List selected tasks
Set classes to log
Change events to log
Enable all logging
Disable all logging

```

5. Select Set classes to log. You will see a screen similar to the following.

```

DIAGNOSTIC LOGGING
Event Class Selections          1/10

NO Execute KAREL line
NO Execute TPP line
NO Call/return
NO Motion
NO Condition Handler
NO Interrupt rtn
NO Task start/end
NO Packet_rcd
NO Pcode exec

```

6. Enable or disable classes of events to be logged.
- To turn on logging for a class of events , press F4, YES.
 - To turn off logging for a class of events , press F5, NO.
- The changes take effect immediately. The current status of whether an event is being logged is displayed to the left of the event.

NOTE

If you specify an entire class of events, you will enable or disable logging for all event types in that class.

9.44.3.7 Event detail selection screen

The Event Detail Selections Screen allows you to enable and disable logging of specific events. Use Procedure 9-54 to enable or disable a specific event type for logging.

Procedure 9-54 Enable or Disable a Specific Event Type for Logging

Steps

1. Press MENUS.
2. Select TEST CYCLE.
3. Press F1, [TYPE].
4. Select Debug Ctl. You will see a screen similar to the following.

```

DIAGNOSTIC LOGGING
Main Menu

Clear event log
Dump log data
Add a task to log
Stop logging a task
List selected tasks
Set classes to log
Change events to log
Enable all logging
Disable all logging

```

5. Select Change events to log. You will see a screen similar to the following.

DIAGNOSTIC LOGGING	
Event Detail Selections	1/30
YES	Controller power-up
YES	Logging enabled
YES	Logging disabled
NO	Execute KAREL line
NO	KAREL or TPP routine called
NO	KAREL or TPP routine returned
YES	Motion started
YES	Motion planned
YES	Motion cancel issued

6. Enable or disable a specific event type to be logged.
 - To turn on logging for an event type , press F4, YES.
 - To turn off logging for an event type , press F5, NO.

The changes take effect immediately.

9.44.3.8 Enable or disable all event logging

The Enable/Disable All Logging Screen allows you to start or stop the automatic logging of all selected events. When the Program Diagnostic Option is installed, logging of selected events in selected tasks is enabled. The *Disable all logging* menu selection turns off all logging. It also eliminates the overhead added when logging is enabled (refer to Subsection 9.44.2.4 "Performance.") The *Enable all logging* menu selection can be used to start logging again.

NOTE

Event and task selections are not changed by Disable all logging or Enable all logging .

Use Procedure 9-55 to enable and disable all event logging.

Procedure 9-55 Enabling and Disabling All Event Logging

Steps

1. Press MENUS.
2. Select TEST CYCLE.
3. Press F1, [TYPE].
4. Select Debug Ctl. You will see a screen similar to the following.

DIAGNOSTIC LOGGING	
Main Menu	
Clear event log	
Dump log data	
Add a task to log	
Stop logging a task	
List selected tasks	
Set classes to log	
Change events to log	
Enable all logging	
Disable all logging	

5. Select either Enable or Disable all logging.

⚠ CAUTION

If you Enable or Disable all logging, you must perform a Cold Start for the change to take effect.

- If you select Enable all logging, selected events in selected tasks will automatically start being logged after you perform a Cold Start.
- If you select Disable all logging, selected events in selected tasks will automatically stop being logged after you perform a Cold Start.

6. Perform a Cold Start.

9.44.4 Logging Events

Events are conditions or situations that occur in a KAREL or teach pendant program while the program is running. The KAREL Program Execution History Record Option allows you to record certain events, as they take place, to help you in debugging programs. The logged events can be dumped to an ASCII file.

9.44.4.1 Setting up events

An event will be recorded in the log when all of the following are true:

- You have selected Enable all logging from the main menu.
- You have enabled event logging for the task.
- You have selected the event type to log either,
 - Specifically, using the Change Events To Log screen to select a specific event.
 - Generally, using the Set Classes To Log screen to select a class of events.

Types of Events that can be Logged

With the Program Diagnostic Option, the following kinds of events can be recorded. Refer to Section 9.44.4.4 for additional information.

- Controller power-up
- Logging enabled
- Logging disabled
- Execute KAREL line
- KAREL or TPP routine called
- KAREL or TPP routine returned
- Motion started
- Motion planned
- Motion cancel issued
- Motion stop issued
- Resume move
- Motion done received
- Motion completed normally
- Condition handler triggered
- Condition handler enabled
- Condition handler disabled
- Before processing ISR
- Return from KAREL ISR
- KAREL or TPP task starts execution
- KAREL or TPP task aborts
- Interpreter receives packet
- Starting execution of p-code
- Start of execution of TPP line
- MMR_RCVD (MMR received)
- MMR received
- Normal AMR received by AMGR

- Start AMR received by AMGR
- Stop AMR received by AMGR
- AMR sent to AX
- AMR received from AX

9.44.4.2 Logging events to an ASCII file

The last 500 events that are recorded can be written to an ASCII file. This can be done by requesting to write the logged data to a file on the Dump Selections screen (refer to Subsection 9.44.3.2).

9.44.4.3 ASCII file general event information

Specific information about each event is logged to the ASCII file. Table 9.44.4.4 describes the following information for each event:

- **Event name:** Event description displayed in the Event Detail Selections screen.
- **When recorded:** Conditions under which the event is recorded. For example, the STRT_K_LINE event is recorded when a KAREL statement is about to be executed.
- **Information recorded:** List of values logged with each event. The following are reported for all events (except as noted).
 - Event time (seconds) since power-up. This is always a multiple of 4 milliseconds (.004 second).
 - Number of the task associated with the event.
 - Name of the KAREL routine or TPP program, in which the statement was triggered.
 - Program line number at which statement event was triggered.
- **Enabled default:** YES if logging of the specified event type is Enabled by default.
- **Comments:** Additional information, if appropriate.

9.44.4.4 ASCII file specific event information

Table 9.44.4.4 contains each event and the corresponding information that is logged.

Table 9.44.4.4 Event Logging Information

Event Name	When Recorded	Information Recorded	Enabled Default	Comments
Controller power-up	At every controller COLD start	Routine name, line number		
		Clock time: at which power-up occurred, in "DD-MMM-YY HH:MM" format.		
		Task number will always be 1.		
Logging enabled	When logging is started. Typically, this is when the task starts.	Standard information only	YES	
Logging disabled	When logging is stopped. Typically, this is when the task ends.	Standard information only	YES	
Execute KAREL line	At beginning of execution of KAREL statement.	Internal information	NO	
		Internal information		
		Internal information		

Event Name	When Recorded	Information Recorded	Enabled Default	Comments
KAREL or TPP routine called	When a KAREL routine or KAREL or TPP program is called. This is either directly or as a result of a condition handler action.	Standard information only	NO	Routine name is name of called routine; line number is line number from which call was made or interrupt occurred.
KAREL or TPP routine returned	When a KAREL PROGRAM or ROUTINE or TP program returns to the calling or interrupted program	Standard information only	NO	The ROUTINE name is the name of the returning program or routine; the line number is the line in the calling or interrupted program to which the task is returning, generally the same as the line number shown for in the CALL event.
Motion started	When a request to initiate a motion is issued.	Group mask	NO	Intended for system level analysis.
		MMR address, in hexadecimal		
		MMR status, in hexadecimal; normally FFFFFFFF		
Motion planned	For TPP motion commands, this occurs typically before the motion is actually started.	Group mask	NO	Intended for system level analysis.
		MMR address, in hexadecimal		
		MMR status, in hexadecimal; normally FFFFFFFF		
Motion cancel issued	When a CANCEL statement or condition handler action is executed, a MOVE... UNTIL condition is satisfied, a CANCEL severity error is posted, or a program is aborted with a motion planned or in progress.	Group mask	NO	Intended for system level analysis.
		MMR address, in hexadecimal		
		MMR status, in hexadecimal; normally FFFFFFFF		
Motion stop issued	When a STOP statement or condition handler action is executed or a STOP severity error is posted.	Group mask	NO	Intended for system level analysis.
		MMR address, in hexadecimal		
		MMR status, in hexadecimal; normally FFFFFFFF		
Resume move	When a stopped motion is resumed; typically when a RESUME statement or condition handler action is executed; also when a program is CONTINUED following a STOP error condition.	Group mask	NO	Intended for system level analysis.
		MMR address, in hexadecimal		
		MMR status, in hexadecimal; normally FFFFFFFF		

Event Name	When Recorded	Information Recorded	Enabled Default	Comments
Motion done received	When the termination type is satisfied for a KAREL or TPP motion statement is executed. For KAREL MOVE ... NOWAIT statements, this event is recorded when the motion starts.	Group mask	NO	Intended for system level analysis.
		MMR address, in hexadecimal		
		MMR status, in hexadecimal; normally FFFFFFFF		
Motion completed normally	When a motion is completed or cancelled. This is generally after MTN_DONE is recorded.	Group mask		Intended for system level analysis.
		MMR address, in hexadecimal		
		MMR status, in hexadecimal; normally FFFFFFFF		
Condition handler triggered	When a global condition handler triggers	Intended for system level analysis.	NO	Intended for system level analysis.
Condition handler enabled	When a global condition handler is enabled by a ENABLE statement or condition handler action.	Condition handler number	NO	Intended for system level analysis.
Condition handler disabled	When a global condition handler is disabled by a DISABLE statement or condition handler action. It is not recorded when a condition handler triggers.	Condition handler number	NO	Intended for system level analysis.
Before processing ISR	Before calling a KAREL PROGRAM from a condition handler.	Internal information	NO	The routine name and line number indicate the code that was executing when the interrupt routine request was received.
		Internal information		
		Internal information		
After processing ISR	When the interrupt routine is ready to run.	Standard information only	NO	Routine name is the name of the interrupt routine. Line number will always be 1.
Return from KAREL ISR	When exiting from an interrupt routine.	Standard information only	NO	Routine name and line number are the interrupt routine name and the line number from which it returned.
KAREL or TPP task starts execution	When a task selected for logging starts executing.	Standard information only	YES	Routine name is the main program name; line number is always 1.

Event Name	When Recorded	Information Recorded	Enabled Default	Comments
KAREL or TPP task aborts	When a task ends.	Standard information only	YES	Routine and line number indicate the last statement executed by the task. This might be an END statement of a KAREL program or the last line of a TPP program.
Interpreter receives packet	When an interpreter task receives a packet.	Packet address	NO	Intended for system level analysis.
		Packet status		
		Request code (including sub-system code)		
		Requestor id		
		ITR-level		
Starting execution of p-code	At start of execution of KAREL p-code instruction.	Number of words on the routine stack	NO	Intended for system level analysis.
		Number of words on the data stack		
		P-code mnemonic		
Start of execution of TPP line	At start of execution of a line of a TPP program.	Number of words on the data stack	NO	
		Number of words on the routine stack		
		Number of words available for combined ROUTINE and DATA stacks		
MMR received	When an MMR is received back from the the motion sub-system. This occurs when the motion completes, is cancelled, or is stopped.	Group mask	NO	
		MMR address, in hexadecimal		
		MMR status, in hexadecimal; normally FFFFFFFF		
Normal AMR recvd by AMGR	An AMR is received by AMGR.	Address of AMR	NO	Intended for system level analysis.
		AMR number		
		AMR AMGR_wk		
		AMR ax_phase		
Start AMR recvd by AMGR	A start AMR request is processed by AMGR.	Address of AMR	NO	Intended for system level analysis.
		AMR number		
		AMR AMGR_wk		
		AMR ax_phase		

Event Name	When Recorded	Information Recorded	Enabled Default	Comments
Stop AMR recvd by AMGR	A stop AMR request is processed by AMGR.	Address of AMR	NO	Intended for system level analysis.
		AMR number		
		AMR AMGR_wk		
		AMR ax_phase		
AMR sent to AX	When an AMR is sent from AMGR to an AX task.	Address of AMR	NO	Intended for system level analysis.
		AMR number		
		AMR AMGR_wk		
		AMR ax_phase		
AMR rcvd from AX	When an AMR is receivedack from an AX task.	Address of AMR	NO	Intended for system level analysis.
		AMR number		
		AMR AMGR_wk		
		AMR ax_phase		

9.44.5 Examples

9.44.5.1 Overview

This appendix contains example programs and the resulting log data, that show how the KAREL Program Execution History Record Option operates.

- Subsection 9.44.5.2 contains a KAREL program (T) which calls a teach pendant program (TPP).
- Subsection 9.44.5.3 contains the teach pendant program that is called (TTT).
- Subsection 9.44.5.4 contains the log file that is generated after running the KAREL program (T) with the following Event class selections:
- YES Execute KAREL line
- YES Execute TPP line
- YES Call/return
- YES Motion
- YES Condition Handler
- YES Interrupt rtn
- YES Task start/end
- NO Packet_rcd
- NO Pcode exec

9.44.5.2 KAREL program example

The KAREL program in Example 9.44.5.2 is the MAIN program from which the TPP program is called and executed.

Example 9.44.5.2 KAREL Program Example (T.KL)

```

1 program t
2 var
3   i,j,k: INTEGER
4
5 routine ttt from ttt
6 routine tt
7 begin
8   i = i + 1

```

```

9 end tt
10
11 begin
12
13 condition[1]:
14   when k >= 500 DO
15     k = 0
16     tt
17     enable condition[1]
18   endcondition
19 k = 0
20 i = 0
21 connect timer to k
22 enable condition[1]
23 wait for i=2
24 disconnect timer k
25 disable condition[1]
26 ttt
27 DELAY 1000
28 end t

```

9.44.5.3 Teach Pendant Program Example

The TPP program in Example 9.44.5.3 is called by the KAREL program (T) in Example 9.44.5.2.

Example 9.44.5.3 Teach Pendant Program Example (TTT.TP)

```

1:J P[1] 100% FINE
2: WAIT 0.00(sec)
3: R[1]=0
4: LBL[1]
5: R[1]=R[1]+1
6: IF R[1] <= 3,JMP LBL[1]
7: WAIT .50(sec)

```

9.44.5.4 ASCII File Example

With logging task T enable and event classes selected as indicated in Example 9.44.5.2 , if T is executed and the log dumped, the file show in Example 9.44.5.4 is generated.

NOTE

For detailed information about the fields on the ASCII report shown in Example 9.44.5.4, refer to Subsection 9.44.4.4.

Example 9.44.5.4 ASCII File Example

Event	Time	TID	Routine	Line				
POWER-UP	.104	1	***	***	09- 9-28	11:53		
TASK-START	84.888	8	T	1				
LOG-ENABLE	84.888	8	T	1				
STRT-K-LINE	84.888	8	T	13	0	0	19200	
STRT-K-LINE	84.904	8	T	19	0	0	19200	
STRT-K-LINE	84.904	8	T	20	0	0	19200	
STRT-K-LINE	84.904	8	T	21	0	0	19200	
STRT-K-LINE	84.904	8	T	22	0	0	19200	
STRT-K-LINE	84.904	8	T	23	0	0	19200	
CH-ENABLE	84.904	8	T	23	1			
CH-TRIGGER	85.912	8	T	23	1			
PRE-ISR	85.912	8	T	23	0	128	35	
CALL	85.912	8	TT	23				
IN-ISR	85.912	8	TT	1				
STRT-K-LINE	85.912	8	TT	8	0	68	1792	
STRT-K-LINE	85.928	8	TT	9	0	68	1792	
RETURN	85.928	8	TT	23				
RTN-INT-RTN	85.928	8	TT	23				
CH-TRIGGER	86.920	8	T	23	1			
PRE-ISR	86.920	8	T	23	0	128	35	
CALL	86.920	8	TT	23				
IN-ISR	86.920	8	TT	1				
STRT-K-LINE	86.920	8	TT	8	0	68	1792	
STRT-K-LINE	86.936	8	TT	9	0	68	1792	
RETURN	86.952	8	TT	23				
RTN-INT-RTN	86.952	8	TT	23				
STRT-K-LINE	86.952	8	T	24	0	0	19200	
STRT-K-LINE	86.952	8	T	25	0	0	19200	
STRT-K-LINE	86.952	8	T	26	0	0	19200	
CH-DISABLE	86.952	8	T	26	1			
CALL	86.952	8	TTT	26				
STRT-T-LINE	86.952	8	TTT	1	0	16	15104	
PLAN-MOVE	86.952	8	TTT	1	405E61A8	FFFFFFFF	6	
START-MOVE	86.952	8	TTT	1	405E61A8	FFFFFFFF	6	
MTN-DONE	88.088	8	TTT	1	405E61A8	FFFFFFFF	3	
STRT-T-LINE	88.088	8	TTT	2	0	16	15104	
MMR_RCVD	88.104	8	TTT	1	405E61A8	00000000	3	
MTN_ENDED	88.104	8	TTT	1	405E61A8	00000000	3	
STRT-T-LINE	88.104	8	TTT	3	0	16	15104	
STRT-T-LINE	88.104	8	TTT	4	0	16	15104	
STRT-T-LINE	88.104	8	TTT	5	0	16	15104	
STRT-T-LINE	88.104	8	TTT	6	0	16	15104	
STRT-T-LINE	88.136	8	TTT	4	0	16	15104	
STRT-T-LINE	88.136	8	TTT	5	0	16	15104	
STRT-T-LINE	88.136	8	TTT	6	0	16	15104	
STRT-T-LINE	88.168	8	TTT	4	0	16	15104	
STRT-T-LINE	88.168	8	TTT	5	0	16	15104	
STRT-T-LINE	88.168	8	TTT	6	0	16	15104	
STRT-T-LINE	88.200	8	TTT	4	0	16	15104	
STRT-T-LINE	88.200	8	TTT	5	0	16	15104	
STRT-T-LINE	88.200	8	TTT	6	0	16	15104	
STRT-T-LINE	88.200	8	TTT	7	0	16	15104	
RETURN	89.208	8	TTT	26				
STRT-K-LINE	89.208	8	T	27	0	0	19200	
STRT-K-LINE	91.224	8	T	28	0	0	19200	
TASK-END	91.240	8	T	28				
LOG-DISABLE	91.240	8	T	28				

9.45 TORQUE LIMIT FUNCTION

9.45.1 Introduction

Using a TP instruction, Torque Limit function can limit the maximum motor torque as desired. Then you can realize followings.

This function requires optional software "Torque limit function" (A05B-2500-J611).

And there is a optional software "Torque limit multi-axis setup function" to set up torque limit for multi axes.

"Torque limit multi-axis setup function".

This function requires optional software "Torque limit function" (A05B-2500-J611) and "Torque limit multi-axis setup function"(A05B-2500-J969).

This section describes Torque Limit function using a Servo Hand which has a servo driven gripper.

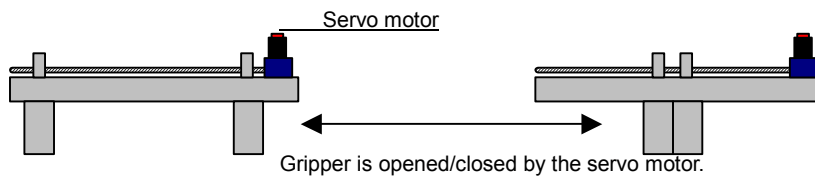
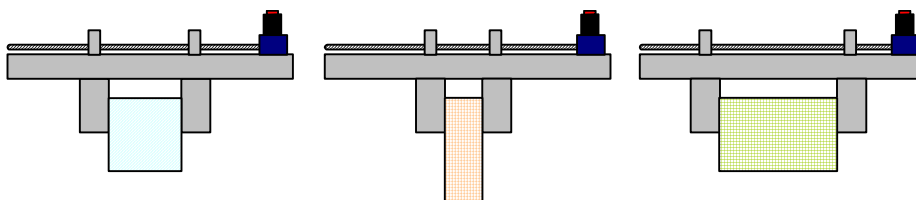


Fig. 9.45.1 (a) Example of Servo Hand

(1) Easy to teach the gripper position

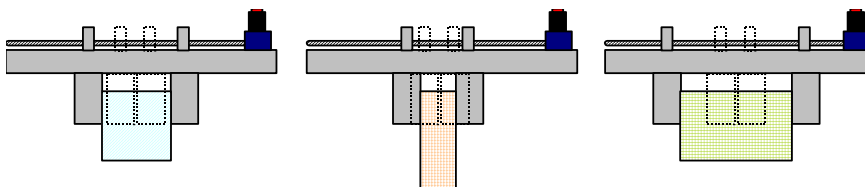
Servo Hand can control the Open/Close position of gripper, but it is necessary to teach gripper position according to the work piece length (size). Then it is necessary to teach position for each work piece type, and it is not easy to operate and maintain the Servo Hand.



It is necessary to teach the gripper positions according to the size of work piece.

Fig. 9.45.1 (b) Teaching by Regacy style

This function can stop the gripper movement when the external force (load) of the servo motor exceed the specified motor torque. Therefore if you teach a closed gripper position and use Torque Limit for this motion instruction, the gripper will stop automatically at the position fit to the work length.



Merit1: Easy to teach

One teaching point (where gripper is closed) can be used for all work pieces.

Fig. 9.45.1 (c) Teaching by torque limit function

9.45.2 Torque limit function feature

Torque Limit function can limit the maximum torque of servo motor which is controlled by a robot controller.

You can set Torque Limit value in % of the maximum motor torque.

Syntax ... TORQ_LIMIT t% (t: 0.0 to 100.0)

Using following system variables, you can select the axes to be limited the motor torque. (By default, J7 axis in Group 1 is enabled.)

-Group number

\$TORQUE_LMT.\$GROUP[i] ... Select the group number including the servo motor using Torque Limit.

i : Group number

-Axis number

\$TORQUE_LMT.\$GAi[j] ... Select the group number and axis number including the servo motor using Torque Limit.

i : Group number, j : Axis number

For example, J1 axis in Group 2 uses Torque Limit, set \$G2[1], and set others FALSE.

\$TORQUE_LMT.\$GROUP[2] = TRUE

\$TORQUE_LMT.\$GA2[1] = TRUE

\$TORQUE_LMT.\$GA2[2] = FALSE

:

\$TORQUE_LMT.\$GA2[9] = FALSE

You can set multiple axes to TRUE.



CAUTION

Usually GA1[1] - [6] are robot axes. So, please do NOT set these variables to TRUE.

It is necessary for you to decide Torque Limit value.

It is necessary to decide Torque Limit value by gripping actual work piece.

After a TORQ_LIMIT instruction is executed, the maximum motor torque will not changed until another TORQ_LIMIT instruction will be executed.

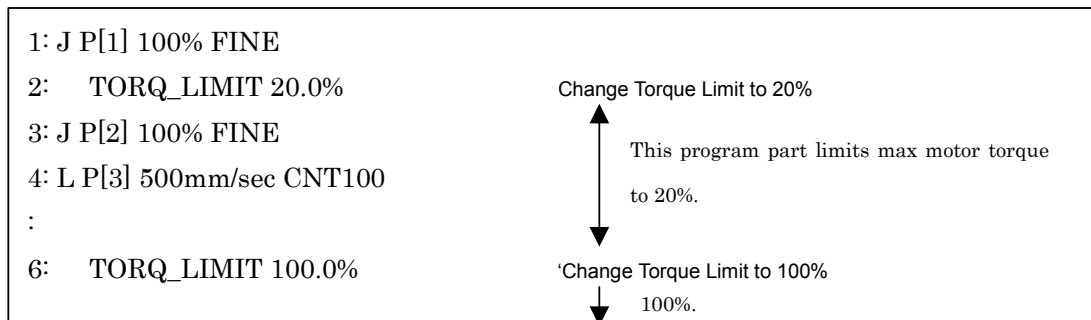


Fig.9.45.2 Example program of torque limit function

After decreasing maximum motor torque by TORQ_LIMIT instruction, the gripper axis may not reach the destination point.

It is necessary to set following system variables to large value in advance.

\$PARAM_GROUP[group].\$STOPTOL[axis]

\$PARAM_GROUP[group].\$STOPERLIM[axis]

\$PARAM_GROUP[group].\$MOVER_OFFST[axis]

Setting the upper limit of Torque Limit

The upper limit of Torque Limit can be specified by the following system variables.

\$TORQUE_LMT.\$MAX_TRQ_LMT = Upper limit (0.1 - 100.0)

The lower limit is fixed to 0.1.

9.45.3 Torque Limit Multi-axis Setup Function

This function requires optional software "Torque limit function" (A05B-2500-J611) and "Torque limit multi-axis setup function"(A05B-2500-J969).

9.45.3.1 Torque limit multi-axis setup function

Please set system variable \$karel_enb to TRUE in order to select karel at "Call" instruction.
You can set torque limit to multi-axis using "CALL" from TP program.
Following explains how to use from TP program.

CALL TPTRQLIM(group number, axis number, torque limit value(%))

Fig.9.45.3.1 (a) How to call "Torque limit multi-axis setup function"

- group number : Set the group number including the servo motor using Torque Limit.
- axis number : Set the axis number including the servo motor using Torque Limit.
- torque limit value : Set Torque Limit value in % of the maximum motor torque.
(You can set 0.1% to 100.0%. You can also use integer.)



CAUTION

Do not set robot axis to axis number.

Following is a example TP program.(This example TP program set 70% torque limit to axis 7 of group 1)

CALL TPTRQLIM(1, 7, 70)

Fig.9.45.3.1 (b) Example program of Torque limit multi-axis setup function

9.45.3.2 Error code of torque limit multi-axis setup function

INTP-657 STOP Option software not installed

[Cause] Option software "Torque limit function" or "Torque limit multi-axis setup function" is not installed.

[Remedy] Install both "Torque limit function" and "Torque limit multi-axis setup function".

INTP-658 STOP Invalid group value

[Cause] Group number is not within 1 to group number of the system or invalid value.

[Remedy] Input a correct value for group number.

INTP-659 STOP Axis not exist or is robot axis

[Cause] Axis number is not within 1 to max value of axis number or robot axis or invalid value.

[Remedy] Input a correct value for axis number.

INTP-660 STOP Invalid torque limit value

[Cause] Torque limit value is not within 0.1 to 100.0 or invalid value.

[Remedy] Input a correct value for torque limit value.

9.45.4 Limitations

- This function limits the maximum motor torque of the specified servo motor, but this function does NOT guarantee the motor torque while rotating the servo motor.
- Because this function is designed for FANUC servo motor, if you want to use this function to non FANUC servo motor, please consult to FANUC LTD.

9.45.5 Caution

- It is possible to use Torque Limit for robot axes, but please do NOT enable this function for any robot axes which may be fell down by the gravity.
- When an axis is stopped by Torque Limit, the external force is equal to the limited motor torque, and the distance between current position and destination position is considered as position error. (Please see the highlighted part in the figure below.) If you changed Torque Limit to 100%, the axis moves to the destination with 100% of motor torque, therefore you may break Servo Hand or a work piece. If you want to change Torque Limit to 100% (or increase motor torque), you have to move the axis where above problem will not occur before changing Torque Limit.

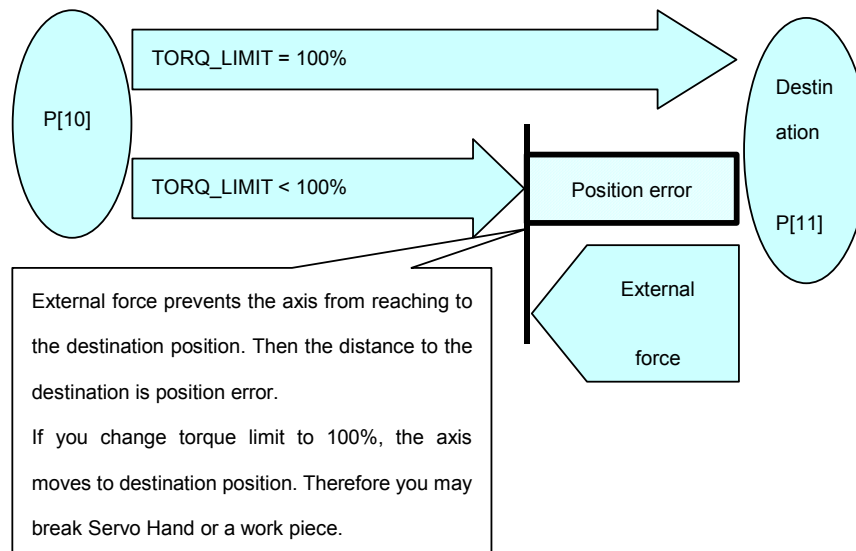


Fig. 9.45.5 Caution for torque limit function

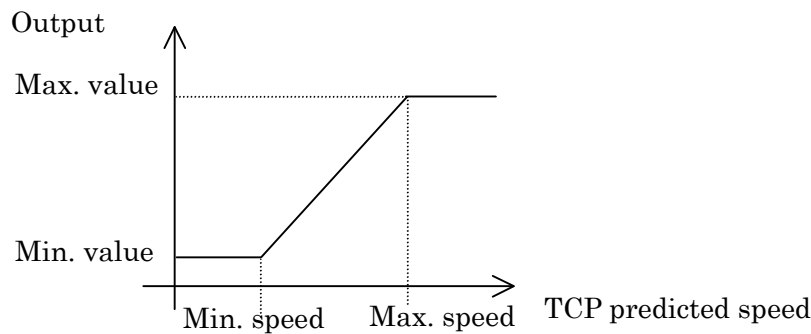
9.46 TCP SPEED OUTPUT

9.46.1 Overview

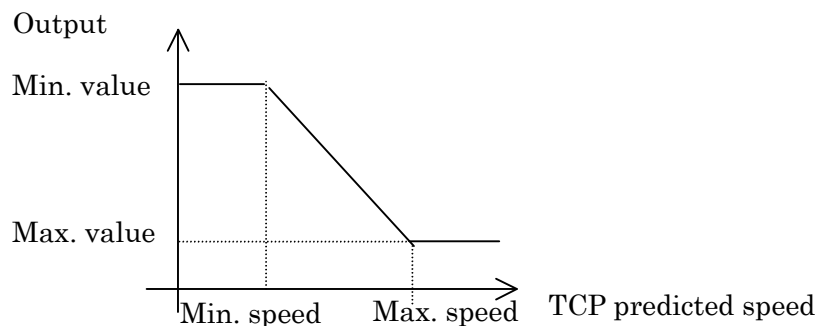
The TCP speed output function outputs the predicted speed of tool center point (TCP) via analog output signals, group output signals, or registers. A software option, TCP speed output function, is required to use this function. The TCP predicted speed is converted on the output condition that is defined by the following setup items:

- Min. value
- Max. value
- Min. speed (mm/sec)
- Max. speed (mm/sec)

The following figure shows the relationship between TCP predicted speed and output.



The Min value is output when TCP predicted speed is the specified Min speed or below. The Max value is output when TCP predicted speed is the specified Max speed or above. The following figure shows the relationship between TCP predicted speed and output when you set the Max value to a value of less than Min value.



9.46.2 Limitations

There are limitations on use of the TCP speed output function.

- This function is not available on DispenseTool or Dispense-enabled SpotTool+.
- This function is not available with tracking functions e.g. Line Tracking.
- TCP predicted speed is output only while the robot is moving by program execution.
- TCP predicted speed reflects only group 1 motion.
- TCP predicted speed does not reflect the extended axis motion except the integrated axis one.
- Accuracy of TCP predicted speed may lower when the motor speed limit alarms are posted.
- Accuracy of TCP predicted speed may lower while the frame is being switched.
- Accuracy of TCP predicted speed may lower while the motion is being switched between the normal and the remote TCP motion.
- Accuracy of the TCP predicted speed may lower while one of the following instructions is being executed:
 - WAIT instruction
 - Joint motion instruction
 - Motion instruction with Approach/Retract (Linear Distance) instruction
 - Motion instruction with Process Speed instruction
 - Motion instruction with Max Speed instruction
 - Motion control statement in KAREL program
- Up to 10 output conditions can be defined.
- Only one prediction time can be specified. Different prediction times cannot be specified among output conditions.

9.46.3 Setting up TCP speed output

To use the TCP speed output function, you must set up the output condition using the procedure 9-56.

Procedure 9-56 Setting up TCP speed output conditions

Step

1. Press MENU.
2. Select 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 OFF	AO[0]	0/	0	0/	0
2 OFF	AO[0]	0/	0	0/	0
3 OFF	AO[0]	0/	0	0/	0
4 OFF	AO[0]	0/	0	0/	0
[TYPE] SETUP DETAIL					

5. Press F2, SETUP. You will see a screen similar to the following.

TCP speed output		
Setup		1/1
1 Prediction time (msec):	0	
[TYPE] LIST		

6. Set the value for Prediction time.
7. Press F2, LIST. You will see a screen similar to the following.

TCP speed output					
List					1 /10
Enable	Target	Min (Val/Spd)		Max	
1 OFF	AO[0]	0/	0	0/	0
2 OFF	AO[0]	0/	0	0/	0
3 OFF	AO[0]	0/	0	0/	0
4 OFF	AO[0]	0/	0	0/	0
[TYPE] SETUP DETAIL					

NOTE

Each line in this screen indicates the status of one output condition.

8. Move the cursor to an output condition that you want to set up and press, F3, DETAIL. You will see a screen similar to the following.

TCP speed output		
TCP_SPD[1]		1/6
1 Enable:	OFF	
2 Target:	AO [0]	
3 Min. value:	0.000	
4 Max. value:	0.000	
5 Min. speed (mm/sec):	0.000	
6 Max. speed (mm/sec):	0.000	
[TYPE] LIST		

9. Move the cursor to Target and select the output target.
10. Move the cursor to each item and set the value.
11. To start the output, set Enable to ON.

NOTE

You can also set Enable and Target by using an instruction. Refer to Procedure 9-57 for more information.

12. Press F3, LIST. You will see a screen similar to the following.

TCP speed output						
List	1 / 10					
Enable	Target	Min (Val/Spd)		Max		
1 ON	A0[1]	100/	10	4095/	2000	
2 OFF	A0[0]	0/	0	0/	0	
3 OFF	A0[0]	0/	0	0/	0	
4 OFF	A0[0]	0/	0	0/	0	
[TYPE] SETUP DETAIL						

NOTE

You can set Enable for each output condition on this screen.

9.46.4 TCP speed output instruction

You can set the Enable and Target of TCP speed output function by using an instruction. The following instruction is used to start the output.

Target type[number]=(TCP_SPD[output condition number])

The following instruction is used to terminate the output.

Target type[number]=constant value

Procedure 9-57 directs how to insert the TCP speed output instruction for a register type target.

Procedure 9-57 Inserting TCP speed output instruction

Step

1. Open the program edit screen.
2. Press NEXT.
3. Press F1, [INST].

Instruction	
1 Register	5 JMP/LBL
2 I/O	6 CALL
3 IF/SELECT	7 Palletizing
4 WAIT	8 --- next page ---
[End]	
[INST] [EDCMD]>	

4. Select Registers.

REGISTER statement	
1 ...=...	5 ...=.../...
2 ...=...+...	6 ...=...DIV...
3 ...=...-...	7 ...=...MOD...
4 ...=...*...	8 ...=(...)
[End]	
[CHOICE]	

5. Select ...=(...).

Mixed Logic	
1 R[]	5 GO[]
2 PR[I,j]	6 AO[]
3 DO[]	7 UO[]
4 RO[]	8 --- next page ---
[End]	
[CHOICE]	

6. Select R[].

1 R[...]=(...)
[End]
DIRECT INDIRECT [CHOICE] [LIST]

7. Set the register number for the output.

Mixed Logic	
1 UI[]	5 WI[]
2 UO[]	6 WO[]
3 RI[]	7 TCP_SPD[]
4 RO[]	8 --- next page ---
1 R[1]=(...)	
[End]	

8. Select TCP_SPD[...].

1 R[1]=(TCP_SPD[...])
[End]
DIRECT INDIRECT [CHOICE]

9. Set the output condition number.

9.47 TP DRAM/FILE STORAGE FUNCTION

TP DRAM/FILE Storage function enhances the capacity of program space by providing for storages outside of CMOS. In 1 motion group system, this function extends the available program space from 15,000 points now to 100,000 points. However, the maximum points will be fewer than 100,000 points depending on system configuration.

“R709 TP DRAM/FILE Storage” option is necessary to use this function.

9.47.1 Storages

There are 4 kinds of storages, SHADOW, SHADOW ONDEMAND, FILE, and the existing storage CMOS. The “temporary” storage states mean the transition state when a program is moved between memories for edit or load of a program.

The following tables are for the storages and its description.

Table 9.47.1(a) Storages

Storage	Description
CMOS	Stored into CMOS (non-volatile memory) in the same way as normal controller.
SHADOW	Program is in FROM (non-volatile memory). Loaded into DRAM (volatile memory) on startup.
SHADOW ONDEMAND	Program is in FROM. Loaded into DRAM when accessed.
FILE	Program is in a TP file in the directory specified in “File Path” in Program Configuration screen. Loaded into DRAM when accessed.

Table 9.47.1(b) Temporary storage status

“temporary” storage state	Description
SHAD (DRAM) ONDE	SHADOW ONDEMAND program is loaded into 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.

9.47.1.1 CMOS programs

CMOS programs use CMOS (non-volatile memory) as the storage memory.

The largest program size is dependent on CMOS size.

9.47.1.2 SHADOW programs

SHADOW programs use FROM and DRAM as storage memory. As the size of DRAM is larger than that of CMOS, more programs than before will be available by using SHADOW. The capacity for programs that can be used for SHADOW is dependent on the rest capacity of FROM and DRAM.

A program stored in SHADOW is loaded from FROM into DRAM on start up. A program will be executable by loaded into DRAM.

The start up time will be long if there are too many SHADOW programs.

When SHADOW program is edited, it is moved to CMOS and to be “temporary” storage state SHAD (CMOS).

9.47.1.3 SHADOW ONDEMAND programs

SHADOW ONDEMAND programs use FROM and DRAM as storage memory. SHADOW ONDEMAND is different from SHADOW in that SHADOW ONDEMAND is loaded when a program is accessed for edit or execution of the program. So a SHADOW ONDEMAND program has no effect to start up time.

The capacity of programs that can be used for SHADOW ONDEMAND is also dependent on the rest capacity of FROM and DRAM as with SHADOW.

When a SHADOW ONDEMAND program is loaded, its storage is moved to “temporary” storage state, SHAD(DRAM) ONDE. There is a slight time lag in the first access, but after that there is no time lag because SHAD(DRAM) ONDE, once loaded into DRAM, remains in DRAM until the power is off.

When a SHADOW ONDEMAND program is edited, its storage is moved to CMOS and to be “temporary” storage state SHAD (CMOS) ONDE.

9.47.1.4 FILE programs

FILE programs use storage devices such as memory card as storage memory. In FILE storage, much more programs than in SHADOW can be available by using external storage devices that have large capacity. A FILE program is loaded into DRAM when first accessed as with SHADOW ONDEMAND.

The path to external storage device is specified in “File Path” in Program Configuration screen.

When a FILE program is loaded, its storage is moved into “temporary” storage state, FILE (DRAM).

The time lag in the first access is longer than that of SHADOW ONDEMAND programs, but after that there is no time lag as with SHADOW ONDEMAND because FILE (DRAM) remains in DRAM until the power is off.

When a FILE program is edited, its storage is moved to “temporary” storage state, FILE (CMOS).

9.47.2 Storage Configuration

Procedure 9-58 Confirm storage device

Step

1. Press Select key. Such screen as follows is displayed. “Not Loaded” in the comment means that a SHADOW ONDEMAND or FILE program is not loaded into DRAM.

Select		JOINT 10%	
700000 bytes free		4/20	
No.	Program name	Comment	
3	TEST01	[Not Loaded	]
4	TEST02	[	]
5	TEST03	[	]
6	TEST04	[	]
7	TEST05	[	]
8	TEST06	[	]
9	TEST07	[Not Loaded	]
[TYPE] CREATE DELETE MONITOR [ATTR]			

2. Press F5, [ATTR], and select “Storage”. Such screen as follows is displayed.

Select		JOINT 10%
700000 bytes free		4/20
No.	Program name	Storage
3	TEST01	[SHADOW ONDEMAND]
4	TEST02	[SHAD 1 Comment
5	TEST03	[SHAD 2 Protection
6	TEST04	[SHAD 3 Last Modified
7	TEST05	[SHAD 4 Size
8	TEST06	[FILE 5 Copy Source
9	TEST07	[FILE 6 Storage
		7 Name Only
[TYPE] CREATE DELETE MONITOR [ATTR]		

Procedure 9-59 Display Program Configuration screen

Step

1. Press SELECT key.
2. Press F1, [TYPE], and select "Config...".
3. Such screen as follows is displayed.

Program Configuration				JOINT 10%
				1/8
Storage	Total	Loaded	Used	Memory
CMOS	8	8	44k	700k
SHADOW	8	8	28k	28k
SHAD ONDE	20	1	12k	240k
FILE	30	0	0k	
File Path:				
MC:\				
2	Move From: [CMOS]		To: [SHADOW]	
3	Default Storage:		[SHADOW]	
4	Force defaults:		[OFF]	
5	Backup FILE programs:		[OFF]	
6	Remaining memory limit:		[4096k]	
Limit of -1 means use minimum limit				
7	Memory limit SHADOW:		[-1k]	
8	Memory limit FILE:		[-1k]	
END MOVE REFRESH				

The table below describes the each item in Program Configuration screen.

Table 9.47.2 Program configuration screen

Item	Description
Storage	For the 4 storages, the number of programs and memory usage are displayed. Total : The number of programs that exist in the storage. Loaded: The number of programs that are actually loaded in the storage. Used : The memory usage for loaded programs. Memory: For CMOS, the capacity for programs is displayed. For others, the total size of the programs that exist in the storage.
File Path	Specify the device where FILE programs are stored.
Move From To	To and from locations for moving programs between storage types.
Default Storage	Set the storage for the new programs.
Force defaults	If valid, loaded into the storage specified in the "Default Storage". If invalid, loaded into the storage specified in the program file.
Backup FILE programs	Exclude or include FILE program types in backup.

Item	Description
Remaining memory limit	If remaining DRAM goes lower than the value set in "Remaining memory limit" in some operations such as creating a new program, an alarm occurs.
Memory limit SHADOW	This item decides the total memory size of SHADOW and SHADOW ONDEMAND programs that can be loaded into DRAM. If the total memory size goes over the setting value, an alarm occurs. You can set the value from -1 to 10000. The default value is -1. So there is no limitation on SHADOW. Usually, please do not change this value.
Memory limit FILE	This item decides the total memory size of FILE programs that can be loaded into DRAM. If the total memory size goes over the setting value, an alarm occurs. You can set the value from -1 to 10000. The default value is -1. So there is no limitation on FILE. Usually, please do not change this value.
END	Finish Program Configuration screen and go back to Select screen.
MOVE	Move programs at a time according to the storage types set in "Move From To".
REFRESH	Update FILE program list according to TP files in the directory specified in "File Path". After this operation, there are updated FILE programs displayed in Select screen.

Note

The usage of TEMP is different from that displayed in Program Configuration screen. When you load a program into DRAM, it uses three times as much TEMP as the value of the memory displayed in Program Configuration screen.

Note

These settings are reflected soon after changed.

Procedure 9-60 Move Storage**Step**

- You can move all programs in the same storage to other storage at a time by using "MOVE" in Program Configuration screen.
- In the item "Move From To", select the original storage in "From", and select the new storage in "To".
 - Press F2, [MOVE].
 - Press F4, [YES].
 - Such screen as follows is displayed and moving the storage finished.

Program Configuration				JOINT 10%
				1/8
Storage	Total	Loaded	Used	Memory
CMOS	8	8	44k	700k
SHADOW	8	8	28k	28k
SHAD ONDE	20	1	12k	240k
FILE	30	0	0k	
File Path:				
MC:\				
2	Move From: [CMOS]		To: [SHADOW]	
3	Default Storage:		[SHADOW]	
4	Force defaults:		[OFF]	
5	Backup FILE programs:		[OFF]	
Moved 2 programs				
END	MOVE	REFRESH		

Note

Please confirm that there is enough memory space available in the new storage. If not, an alarm occurs and stops moving. Then, programs that are already moved do not go back the original storage. When you move large number of programs, please make backup in advance.

- You can also change the storage of a program individually.
1. Press Select key and press Next, [>].
 2. Press F2, [DETAIL] and such screen as follows is displayed.
 3. Move the cursor on Storage and press ENTER, and select the storage.

```

Program name:
1 TEST1
2 Sub Type:      [None      ]
3 Comment:      [          ]
4 Group Mask:   [1,*,*,*,*,* ]
5 Write protect: [OFF      ]
6 Storage:      [SHADOW    ]
7 Ignore pause: [OFF      ]
8 Stack size:   [300      ]
  
```

Note

Sometimes, the storage goes back to original storage soon after you change the storage. In such case, please confirm if the program is in edit, there is enough memory space in the new storage, or "File Path" is specified if FILE.

Procedure 9-61 Update FILE program list

Step

- You can update the FILE program list. This operation is used to display the deleted FILE programs again or to use programs in other external storage devices.
1. Input the path of the device where FILE programs are stored, and press ENTER.
 2. Press F3, REFRESH.
 3. Such screen as follows is displayed, and FILE program list is updated.

```

Program Configuration      JOINT 10%
                             1/8
Storage  Total  Loaded  Used  Memory
CMOS      8      8      44k   700k
SHADOW     8      8      28k    28k
SHAD ONDE 20      1      12k   240k
FILE      30      0       0k
File Path:
MC:\
2 Move From: [CMOS  ] To: [SHADOW ]
3 Default Storage:      [SHADOW ]
4 Force defaults:       [OFF    ]
5 Backup FILE programs: [OFF    ]
File program list refreshed
END      MOVE      REFRESH
  
```

Updated FILE programs are listed in Select screen.

The diagram illustrates the effect of the 'REFRESH' button. It shows two side-by-side screens. The left screen displays a list of programs: TEST01 and TEST02, both with a status of [SHADOW]. The right screen displays a list of programs: TEST01, TEST02, TEST03, TEST04, TEST05, and TEST06. TEST01 and TEST02 have a status of [SHADOW], while TEST03, TEST04, TEST05, and TEST06 have a status of [FILE]. A box highlights the text 'FILE programs are added on the list' at the bottom of the right screen. A 'REFRESH' button is shown between the two screens, with an arrow pointing from the left screen to the right screen.

9.47.3 Save / Load programs

9.47.3.1 Save / Load TP files

When a program is saved as TP file, information for storage is also preserved in it. So, when a TP file is loaded, the program is stored in the saved storage according to the information.

When “Force defaults” is “ON” in Program Configuration screen, the program is stored in the storage specified in “Default Storage”.

When the existing TP file that has no information for storage is loaded, the program is stored in the storage specified in “Default Storage”.

9.47.3.2 Copy programs

When a program is copied, the storage of copied program is the same as that of original program.

However, when “Force defaults” is “ON” in Program Configuration screen, copied program is stored in the storage specified in “Default Storage”.

9.47.3.3 Save / Load LS files

When a program is saved in ASCII format, a LS file is created. Then, new item “STORAGE” is added in the LS file. The storage of the program is printed in this item. When this LS file is loaded, the program is stored in the storage specified in the “STORAGE” item.

When a existing LS file that does not have the item “STORAGE” is loaded, the program is stored in the storage specified in “Default Storage”.

Note

“R507 ASCII UPLOAD” option is necessary to load LS files.

Note

“R709 TP DRAM/FILE Storage” option is necessary to load LS files that include “STORAGE” item. To load LS files into the controller without this option, you need to delete “STORAGE” item in advance.

9.47.3.4 Make backup of programs

In File screen, you can make backup of SHADOW and SHADOW ONDEMAND programs as well as CMOS programs.

However, when you make backup of FILE programs, you need to set “Backup FILE programs” “ON” in Program Configuration screen.

9.47.4 Loading process in program execution

The programs that are not loaded on startup such as SHADOW ONDEMAND and FILE programs have time lag until they are loaded, because they are loaded on the start of edit or execution.

For example, SHADOW program TEST01 execute FILE programs TEST02~TEST05 by using CALL instruction.

TEST01		JOINT 10%
		1/5
1:	CALL TEST02	
2:	CALL TEST03	
3:	CALL TEST04	
4:	CALL TEST05	
[End]		

When TEST01 is executed, all the programs (TEST02~TEST05) that are called from TEST001 are loaded into DRAM.

If the size of a FILE program is large, it takes time to load it. So there is a time lag.

If these FILE programs are loaded in advance, TEST01 will start immediately.

Select			Execute TEST01			JOINT 10%		
700000 b						700000 bytes free		
						4/20		
No.	Program name	Comment	No.	Program name	Comment			
3	TEST01	[SHADOW]	3	TEST01	[SHADOW]			
4	TEST02	[FILE]	4	TEST02	[FILE(DRAM)]			
5	TEST03	[FILE]	5	TEST03	[FILE(DRAM)]			
6	TEST04	[FILE]	6	TEST04	[FILE(DRAM)]			
7	TEST05	[FILE]	7	TEST05	[FILE(DRAM)]			
8	TEST06	[FILE]	8	TEST06	[FILE]			
9	TEST07	[FILE]	9	TEST07	[FILE]			
[TYPE] CREATE DELETE MONI			[TYPE] CREATE DELETE MONITOR [ATTR]			TEST02 ~ TEST05 are loaded at a time		

9.47.5 Program exchange function without enough memory space

An error usually occurs when you load SHADOW ONDEMAND or FILE programs without enough memory space in DRAM. This function replaces old programs with newly loaded program in the same storage automatically.

Until there is enough memory space in DRAM for the newly loaded program, old programs are deleted from DRAM in order automatically.

Old programs are deleted, but they still exist in the original storage. So they can be loaded again.

The following figure indicates the behavior of loading a new SHADOW ONDEMAND program, TEST07 when there is not enough memory space in DRAM. When TEST07 is selected, TEST03 moves back to SHADOW ONDEMAND and TEST07 can move into the "temporary" storage state SHAD(DRAM) ONDE.

So, there is a time lag when you load TEST03 again.

You can use this function by setting the value of system variable \$TPDRAM_CFG.\$MANAGE_MEM to "1". By default this value is "0" and this function is invalid.

Select			JOINT 10%		
700000 bytes free			700000 bytes free 7/20		
No.	Program		No.	Program	
3	TEST01	[SHADOW]	3	TEST01	[SHADOW]
4	TEST02	[SHADOW]	4	TEST02	[SHADOW]
5	TEST03	[SHAD(DR]	5	TEST03	[FILE(DRAM]
6	TEST04	[SHAD(DR]	6	TEST04	[FILE(DRAM]
7	TEST05	[SHAD(DR]	7	TEST05	[FILE(DRAM]
8	TEST06	[SHAD(DR]	8	TEST06	[FILE]
9	TEST07	[SHADOW]	9	TEST07	[FILE]
[TYPE] CREATE DELETE MONI			[TYPE] CREATE DELETE MONITOR [ATTR]		

9.47.6 Precaution

9.47.6.1 Cause and Remedy for alarm occurrence

Cause and remedy for alarms that occur when you use this function are listed.

TPIF-036 Not enough memory

Cause:

1. Performed such operations as creating a new program, edit, or copy a program without enough memory space in DRAM and CMOS.
2. Performed such operations as creating, editing, or copying a SHADOW or SHADOW ONDEMAND program in the following situation.
 “Memory limit SHADOW” is valid and the memory usage exceeds the limit.
 There is not enough memory space in CMOS.
3. Performed such operations as creating, editing, or copying a FILE program in the following situation.
 “Memory limit FILE” is valid and the memory usage exceeds the limit.
 There is not enough memory space in CMOS.

Remedy:

1. Delete unnecessary programs using CMOS and DRAM, and make enough memory space in CMOS and DRAM.
2. Set larger value in “Memory limit SHADOW” or set the limit invalid. Delete unnecessary programs using CMOS and make enough memory space in CMOS.
3. Set larger value in “Memory limit FILE” or set the limit invalid. Delete unnecessary programs using CMOS and make enough memory space in CMOS.

MEMO-126 No more available TPP memory

Cause:

1. Performed such operations as editing, copying a program or creating a new CMOS program without enough memory space in CMOS.

Remedy:

1. Delete unnecessary programs using CMOS and make enough memory space in CMOS.

MEMO-178 No more available DRAM (SHADOW/FILE)

Cause:

1. Performed such operations as creating a new program, edit, or copy a program without enough memory space in DRAM.

Remedy:

1. Delete unnecessary programs using DRAM and make enough memory space in DRAM.

9.47.6.2 Effect of alarm occurrence

When you create or copy programs without enough memory space in DRAM or CMOS, an alarm occurs. Then, there are some cases that programs are created in unexpected storage.

Effect 1:

When “Default Storage” is SHADOW, the storage of newly created program will be SHADOW. But when you create a new program without enough memory space in DRAM, the storage of the program will be CMOS automatically, and you cannot select other storages.

Remedy 1:

Delete unnecessary programs using DRAM until there is enough memory space in DRAM. After that, create a program again or change the storage individually.

Effect 2:

When “Default Storage” is SHADOW and “Force Default” is ON, the storage of the program loaded in File screen will be SHADOW. But when you load a program in File screen without enough memory space in DRAM, the storage of the program will be CMOS automatically.

Remedy 2:

Delete unnecessary programs using DRAM until there is enough memory space in DRAM. After that, load the program again or change the storage individually.

Effect 3:

When a SHADOW program is copied without enough memory space in DRAM, SHAD(CMOS) program is created.

When in SHAD(CMOS), the program is not executable.

Remedy 3:

Delete unnecessary programs using DRAM until there is enough memory space in DRAM. The program can move from SHAD(CMOS) to SHADOW when there is enough memory space and then it will be executable.

9.48 CYCLE TIME TRACKING

9.48.1 Overview

Cycle Time Tracking function records and displays the cycle time of programs. Cycle times are recorded based on individual cycles and hourly averages. Multiple cycles can be observed to view trends and analyze data. The cycle time data is displayed graphically and raw cycle data in text format can also be displayed. By default, the cycle time is tracked from the start of a teach pendant program until the program ends.

This is optional function and *i*Pendant is necessary to use this function.

9.48.1.1 Cycle Time Categories

Cycle Time is recorded based on multiple categories. The following table describes each category.

Table 9.48.1.1 Cycle time categories

Category	Description
MOTION	The amount of motion time in the cycle.
PROCESS	The amount of process time including welding time.
APPL	The amount of application-specific time executed such as ARC instruction.
WAIT	The amount of time spent waiting for I/O or a data event. Condition specified WAIT instruction is recorded.

Category	Description
DELAY	Time specified WAIT instruction is recorded.
KAREL	The amount of time spent executing KAREL logic.
LOGIC	The amount of time spent executing instructions except categories above (ex: Label, I/O).
HOMIO	The amount of time spent executing HOME_IO program. HOME_IO is automatically executed when the robot reaches the home position set in Reference Position screen and the program ran by SOP, UOP, or PROD_START ends. To execute HOME_IO, Reference position should be set correctly. If you do not use HOME_IO, home position setting is not necessary.
IDLE	The time, until the next cycle starts after the robot reaches home position and the program ends, is recorded.
FAULT	The amount of time spent during a program is paused by alarms or some other causes.
TOTAL	The total cycle time of categories above except HOMIO, IDLE and FAULT.

NOTE

The cycle time PROCESS, APPL, HOMIO, and IDLE is recorded only when the program is run by SOP cycle start, UOP START, or PROD START.

9.48.1.2 Cycle Time Screen Items

The following table describes the items displayed in STATUS Cycle Time screen.

Table 9.48.1.2 Items in cycle time screen

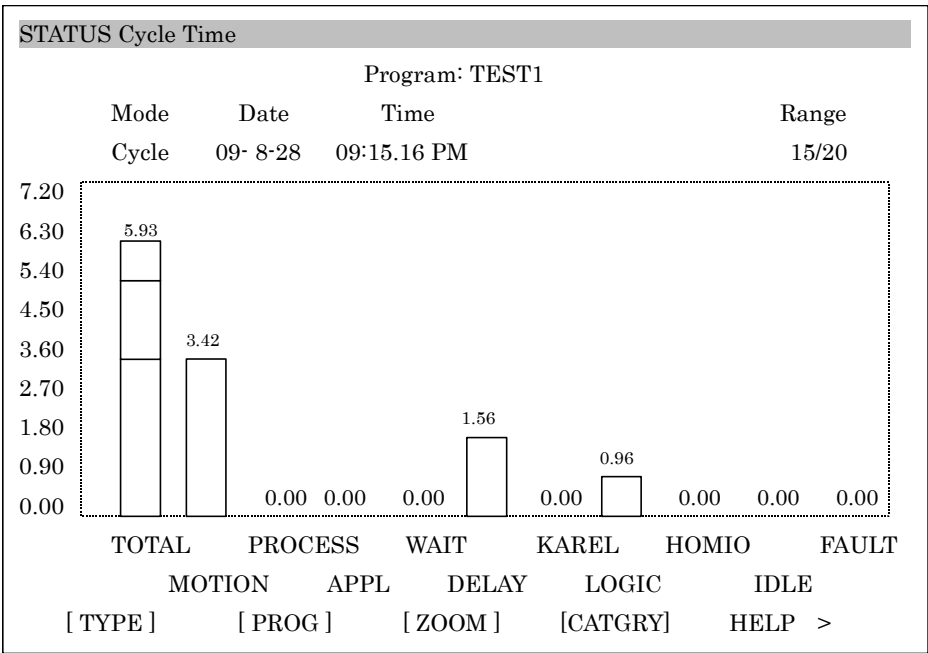
Item	Description
Program	The name of currently tracked program.
Mode	This item indicates whether the cycle is tracked per cycle or per hour. one, ten, or one hundred cycles can be displayed. It can also display one, ten, one hundred hours.
Date	The date of the currently highlighted cycle or hour recorded.
Time	The time of the currently highlighted cycle was executed.
Range (Total)	This indicates the total number of cycles as n/m where m is the total number of cycles or hours, and n is the currently displayed cycle or hour.
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 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, one hundred hours, one cycle, ten cycles, or one 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 of right corner, and cycle time information for that category will be only displayed. You can choose to display the following categories: TOTAL, MOTION, PROCESS, APPL, WAIT, DELAY, KAREL, LOGIC, HOMIO, IDLE, FAULT.
[PAGE]	This function key allows you to display cycle time data in a variety of formats. To display cycle time data on the iPendant, Cycle Graphics should be displayed. Other formats include Cycle Text, and Average Text. Line Text is only available in line-by-line mode. If you select Cycle Text or Average Text, the .DG files will be displayed. CYCLES.DG displays the cycle data, and CYCAVE.DG displays the hourly data. These files can be displayed using the Web Server on the iPendant.
[UPDATE]	This function key is used for the 4 operations in the following. Each detail is described in 9.48.3.3. Load Cycle Data: Load the cycle time data. Clear Cycle Data: Clear the cycle time data. Clear Line Data: Clear the line data in line-by-line mode. Collect Line Data: Get the line data in line-by-line mode.
RECORD	This function key records the specific cycle time of a program. "RECORD" is displayed only in cycle mode when an individual category is selected. Detail is described in 9.48.3.5.

9.48.2 Display Modes of Cycle Time

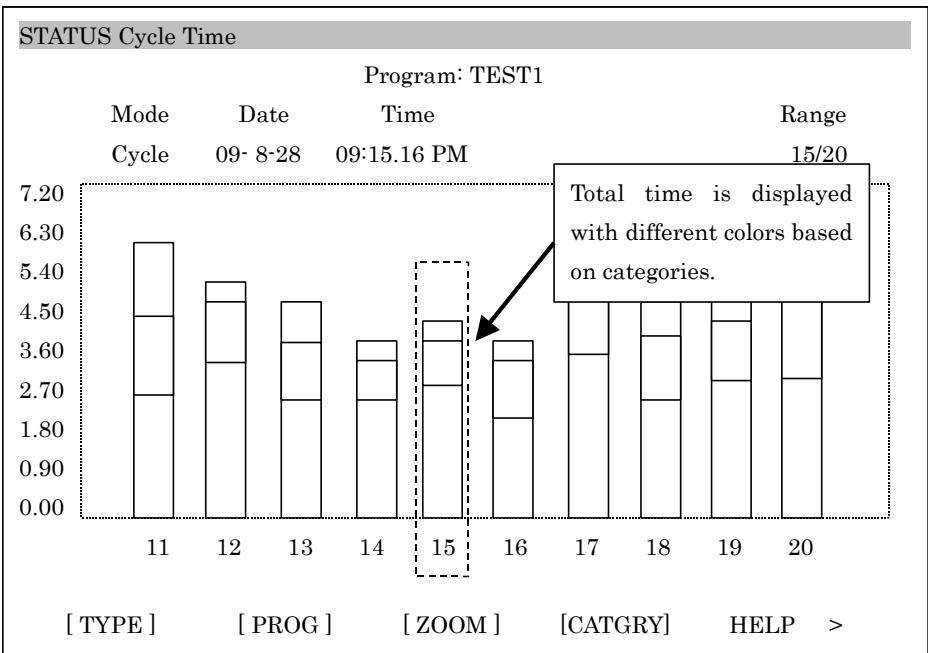
Cycle time data can be displayed in three modes by cycles, by hours, or line-by-line.

9.48.2.1 Cycle Mode

In cycle mode, cycle time is displayed by cycle. You can display cycle time data by one cycle, ten cycles, or one hundred cycles at a time. The following figure 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.

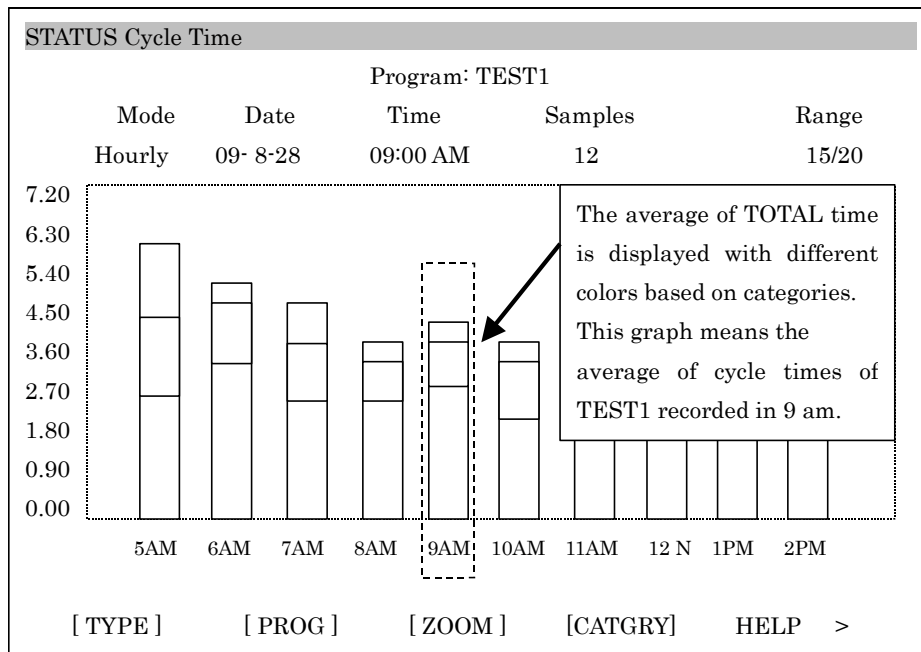


The following figure displays cycle times in ten cycles mode.



9.48.2.2 Hourly Mode

In hourly mode, the hourly average cycle time is displayed. You can display cycle time data by one hour, ten hours, or one hundred hours at a time. Hourly mode is available only when hourly data exists. The following figure displays the cycle time in ten hours mode.



9.48.2.3 Line-by-Line Mode

In line-by-line mode, the cycle time in each line in a program is recorded. So you can see the cycle time of specific line number in the program. The data is saved in CYCLINE.DG in DRAM. So it is cleared when the power is off.

To use line by line mode, \$CY_CONFIG.\$LINEENABLE and \$CY_CONFIG.\$NUMLINES should be configured.

\$CY_CONFIG.\$LINEENABLE Valid when it is TRUE (FALSE by default)

\$CY_CONFIG.\$NUMLINES Set the number you want to record (0 by default)

After that,

1. In STATUS Cycle Time screen, press NEXT, >, press F3, UPDATE and select Collect Line Data.
2. Execute a program of which you want the cycle time.
3. In STATUS Cycle Time screen, press NEXT, >, press F2, [PAGE] and select Line Text.
4. The cycle time of each line is displayed in the Web Server as follows.

Record	Program	Line	Mncode	Duration-uS
0	TEST1	1	254	592680
1	TEST1	2	123	998858
2	TEST1	3	254	1208431

[TYPE] [PAGE] [HELP]

9.48.3 Tracking Cycle Time

Procedure 9-62 Tracking Cycle Time

Condition

- 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.

Step

1. Press MENU.
2. Press STATUS.
3. Press F1, [TYPE].
4. Select Cycle Time.
5. Set the name of the program whose cycle time you want to display. 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 cycle time you want to display. 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 and backward quickly, press SHIFT and the left or right arrow key. If you are in ten cycle mode it will move by 5. If in one hundred cycle mode, it will move by 30.
7. To display cycle time for a specific program category, press F4, [CATGRY] and select the category you want to display.

9.48.3.1 Change Display Format of Cycle Time Data

The format of displaying cycle time data can be changed. In STATUS Cycle Time screen, press NEXT, >, press F2, [PAGE], and select one from the following items.

- Cycle Graphics Cycle time data is displayed graphically.
- Cycle Text Cycle time data is displayed as raw text. The data is saved in CYCLES.DG.
- Average Text Hourly average cycle time is displayed as raw text. The data is saved in CYCAVE.DG.
- Line Text Line data of cycle time in line-by-line mode is displayed as raw text. The data is saved in CYCLINE.DG and is cleared with power off. Line text can be displayed only when line-by-line mode is valid.

NOTE

Text data is displayed on Web Server screen.

9.48.3.2 Save Cycle Time Data

To save the cycle time data, press FCTN key and select SAVE. The cycle time data of cycle mode is saved in PGCYCCYC.TXT and that of hourly mode is saved in PGCYCHR.TXT. They are saved in the device selected in File screen.

F4, [BACKUP] in File screen can be also used.

9.48.3.3 Update Cycle Time Data

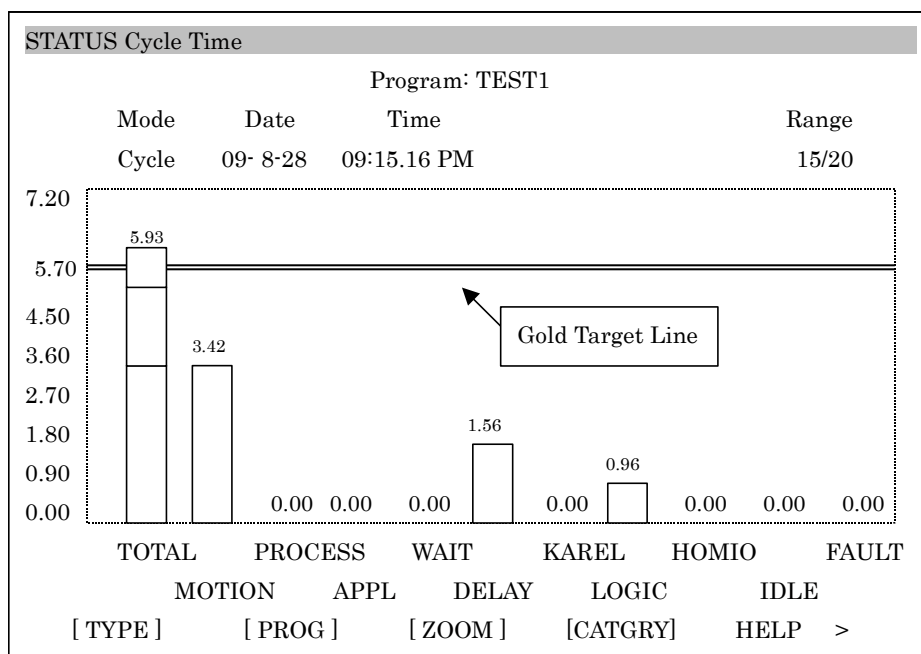
For update of cycle time data, there are 4 operations. In STATUS Cycle Time screen, press NEXT, >, press F3, UPDATE, and select the operation you want to execute. The detail of each operation is described as follows.

- **Load Cycle Data**
Clear the currently recorded cycle time data and load the previous cycle time data from the device that is selected in FILE screen.
Loading PGCYCCY.TXT and PGCYCHR.TXT also can display the previous cycle time data.
- **Clear Cycle Data**
Clear the currently recorded cycle time data.
- **Clear Line Data**
Clear the data for line by line mode. When you want to get a new line data, you should select Collect Line Data again.
- **Collect Line Data**
Collect Line Data is an operation necessary for getting the line data for line-by-line mode. After this operation, execute a program whose cycle time you want to record. Then, in STATUS Cycle Time screen, press NEXT, > and select Line Text. The cycle time for each line is displayed on Web Server screen.

9.48.3.4 Display Target Cycle Time Line

Target cycle time can be displayed as a gold line.
Target cycle time is displayed only in 1 cycle mode.

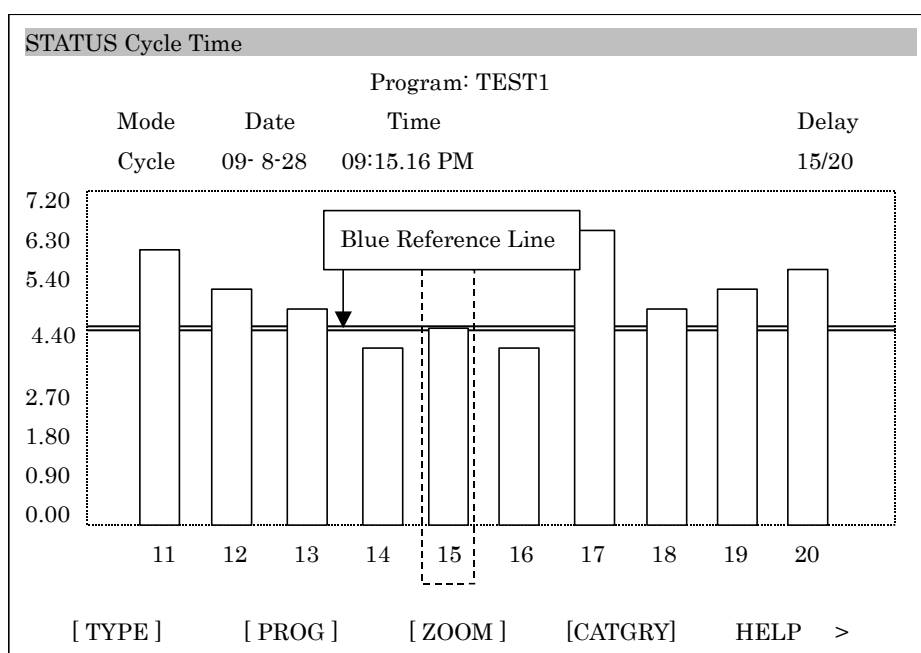
1. Press SELECT and select a program.
2. Press NEXT, > and press F2, DETAIL.
3. Press F3, NEXT until Target Cycle Time (seconds) is displayed.
4. Press ENTER and type a new target cycle time.
5. Press F1, END.
6. Press MENU, select STATUS, press F1, [TYPE] and select Cycle Time if it is not already displayed.
7. Press F2, [PROG] and select the program for which target cycle time is set.
8. Press F3, [ZOOM] and select One Cycle. Such screen as follows will be displayed.



9.48.3.5 Display Reference Cycle Time Line

The cycle time of the current program can be recorded and be used as a reference. The recorded cycle time is displayed as a blue line. The reference cycle time line is displayed only in cycle mode and when an individual category is selected.

1. In STATUS Cycle Time screen, press F2, [PROG] and select the program whose cycle time you want to record.
2. Press NEXT, > and confirm that F4, RECORD is displayed and press it.
3. You will be prompted to answer whether or not you want to overwrite. Select YES and the cycle time will be recorded.
4. Press F4, [CATGRY] and select a category you want to display. DELAY is selected in the following example.
5. The recorded cycle time is displayed as a blue line. (not displayed when TOTAL is selected)



9.48.4 Execution example of Spot program

In this section, as an execution example of Cycle Time Tracking function, the cycle time is recorded by executing the following Spot program, WELDTTEST.

The cycle time of Spot program is not recorded by SHIFT-FWD execution. SOP cycle start, UOP_START, or PROD_START should be used.

WELDTTEST				
1:J	P[1]	100%	FINE	
2:L	P[2]	2000mm/sec	CNT50	
:	SPOT	[SD=1,P=1,S=1,ED=1]		
3:L	P[3]	2000mm/sec	CNT50	
:	SPOT	[SD=1,P=1,S=1,ED=1]		
4:L	P[4]	2000mm/sec	CNT50	
:	SPOT	[SD=1,P=1,S=1,ED=1]		
5:L	P[5]	2000mm/sec	CNT50	
:	SPOT	[SD=1,P=1,S=1,ED=1]		
6:J	P[1]	100%	FINE	
[End]				
POINT	SPOT	GunTchup	AutTchup	TOUCHUP

WELDTTEST has 4 Spot instructions. In this example, the setting of welding time is 250 ms per one Spot instruction. So the total welding time is expected to 1 second and recorded as PROCESS cycle time.

The following HOME_IO program is also used. It has only one instruction of “WAIT 1.00 (sec)” and other statements are all comment. This is to make HOMIO cycle time easy to estimate. HOME_IO is automatically executed when the robot reaches the home position set in Reference Position screen and the program ran by SOP, UOP, or PROD_START ends. The execution time of HOME_IO is recorded as HOMIO cycle time. So, in this example, HOMIO is expected to be 1 second.

HOME_IO	
1:	WAIT 1.00(sec)
2:	!*****
3:	!MACRO SET AT HOME I-O
4:	!Example Housekeeping Routine
5:	!Insert Appropriate Commands
6:	!Where Necessary
7:	!*****
8:	

To execute HOME_IO, Reference position should be set correctly.

REF POSN

Reference Position

Ref.Position Number:1

1:Comment:[*****]

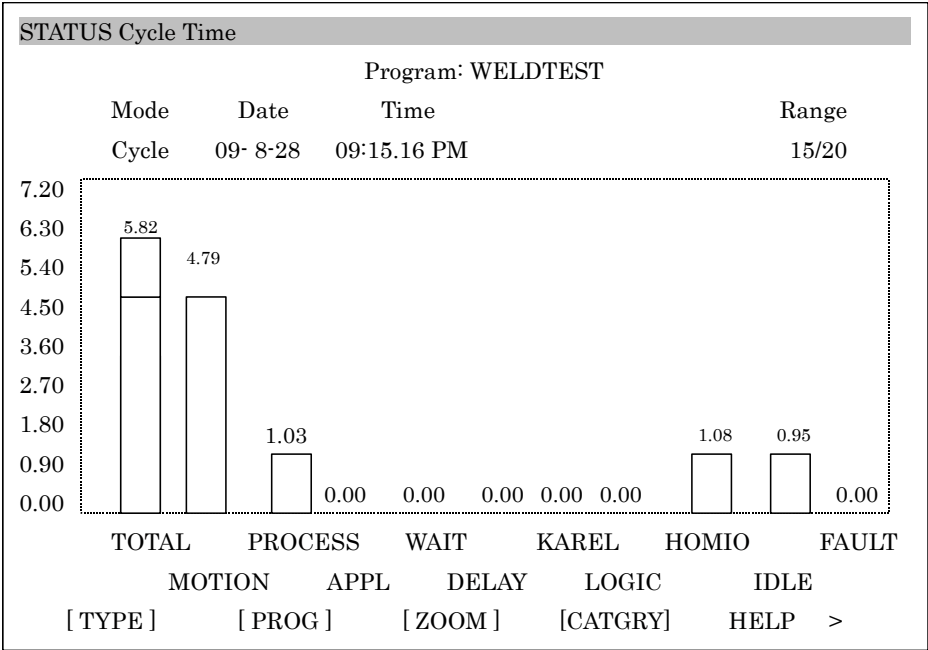
2:Enable/Disable:ENABLE

3:Is a valid HOME:TRUE

Choose ENABLE for "Enable/Disable".

Choose TRUE for "Is a valid HOME".

The following figure is the cycle time of WELDTEST.



As expected, PROCESS cycle time is 1.03 second and HOMIO cycle time is 1.08 second. Here, IDLE cycle time is recorded with the value 0.95 second. IDLE cycle time records the time that the robot stays idle until next program starts after the program ends. In this example, WELDTEST was restarted soon after it ended. This cycle time data shows that it took 0.95 second to restart WELDTEST after it ended.

The Total cycle time includes MOTION and PROCESS cycle times. HOMIO, IDLE, and FAULT are not included in TOTAL cycle time.

9.49 ROBOT TOOLS OF ROBOT HOMEPAGE

R-30iA and R-30iA Mate controller have Web server function by default. The web server allows you to access to robot home page. Please refer to “R-30iA/R-30iA Mate controller Ethernet Function OPERATOR’S MANUAL” (B-82974EN).

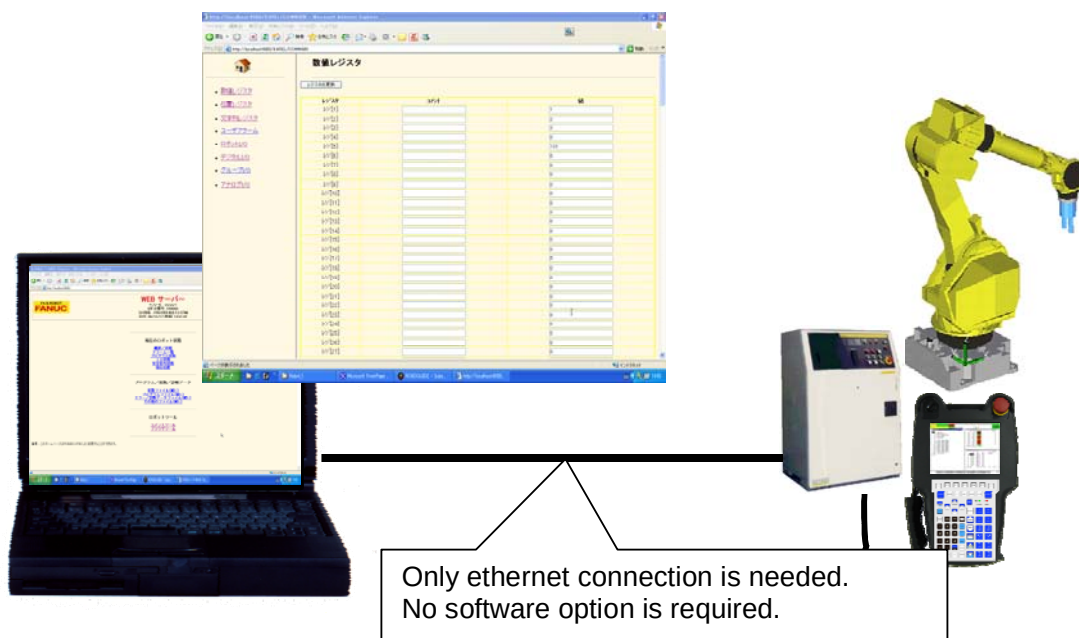
The links on the default home page (7DA7 series or later) called “ROBOT TOOLS” allow you to enter text information easily using a keyboard.

NOTE

ROBOT TOOLS can be used only when system software is 7DA7 series or later.

- PC requires only web browser(Internet explorer) to use ROBOT TOOLS
- Robot controller does not need software option

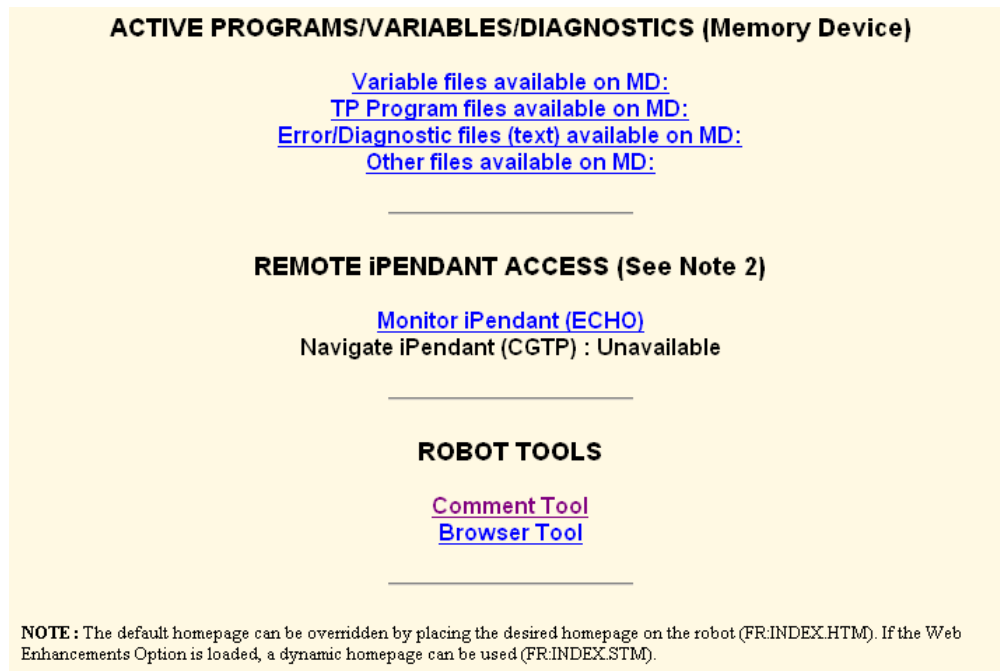
Web Browser



NOTE

- 1 Before using ROBOT TOOLS, you have to allow web server to run KAREL program. Otherwise controller requests you to input username and password even though you have not setup them. Please refer to section 6.5 "HTTP AUTHENTICATION" of "R-30iA/R-30iA Mate controller Ethernet Function OPERATOR'S MANUAL" (B-82974EN). If you don't need authentication, display HTTP AUTHENTICATION screen and set "KAREL" to "U" (Unlock). Please refer to 9.49.1.2.
- 2 You can display robot homepage by *iPendant*. In this case, ROBOT TOOLS are not displayed.

Following figure shows part of robot homepage, including ROBOT TOOLS.



This section first describes outlines of setup required to display ROBOT TOOLS by PC. Then ROBOT TOOLS are described.

9.49.1 Setup for ROBOT TOOLS

This subsection explains setup of IP address and HTTP authentication. Please refer to “R-30iA/R-30iA Mate controller Ethernet Function OPERATOR’S MANUAL” (B-82974EN) for more detail.

9.49.1.1 Setup TCP/IP

- 1 Setup TCP/IP on robot controller.
 - (a) Press MENUS.
 - (b) Select SETUP.
 - (c) Press F1 [TYPE].
 - (d) Select Host Comm. You will see a screen similar to the following.

Busy	Stop	Hold	Pause			AUTO	JOINT	100%
Run	I/O	Prod	Test					

SETUP Protocols		1/8
Protocol	Description	
1 TCP/IP	TCP/IP Detailed Setup	
2 TELNET	Telnet Protocol	
3 SM	Socket Messaging Device	
4 PROXY	Proxy Server	
5 PPP	Point to Point Protocol	
6 PING	Ping Protocol	
7 HTTP	HTTP Authentication	
8 FTP	File Transfer Protocol	

[TYPE]	DETAIL	[SHOW]

(e) Select TCP/IP and press F3 DETAIL. You will see a screen similar to the following.

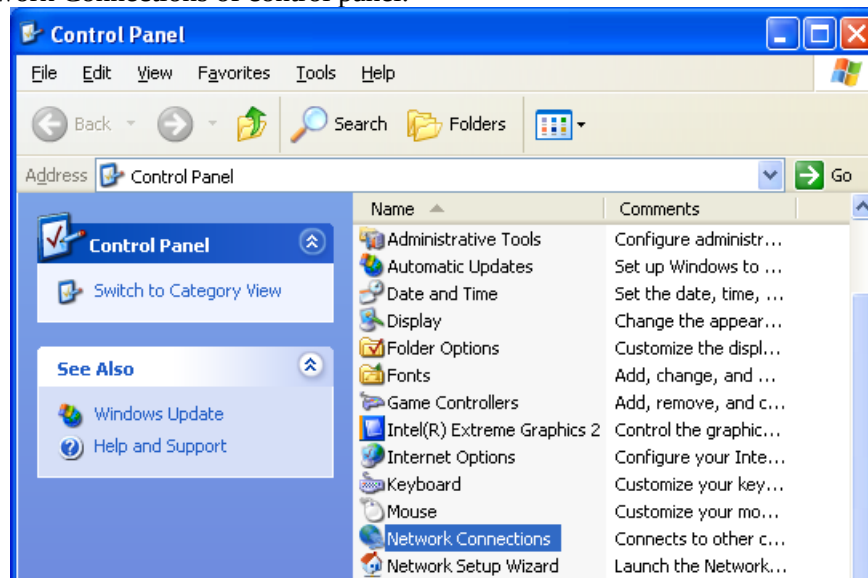
Busy	Stop	Hold	Pause			AUTO	JOINT	100%
Run	I/O	Prod	Test					

SETUP Host Comm		2/40
TCP/IP		
Robot name:	ROBOT	
Port#1 IP addr:	192.168.0.4	
Subnet Mask:	255.255.255.0	
Board address:	*****	
Router IP addr:	*****	
Host Name (LOCAL) Internet Address		
1	*****	*****
2	*****	*****
3	*****	*****
4	*****	*****

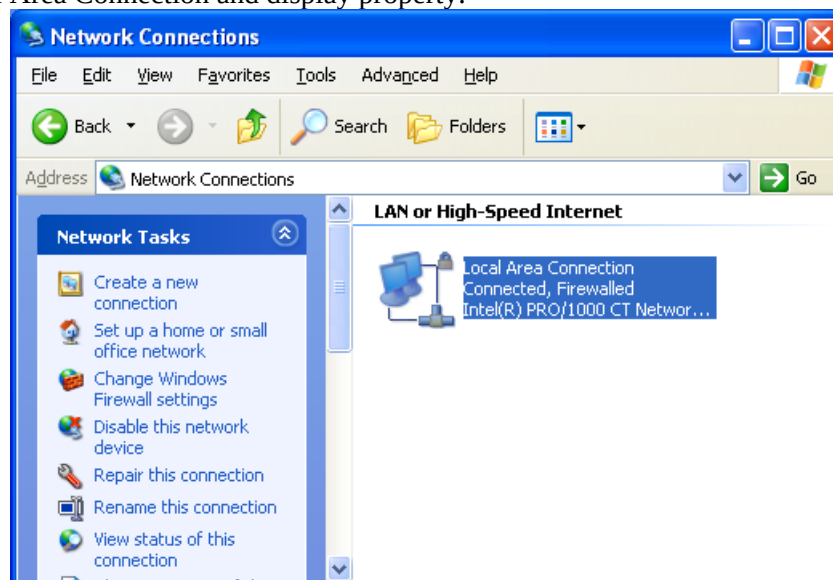
[TYPE]	PORT	PING	HELP	>

- 2 Set IP address.
Set Subnet Mask and Router IP address as required. Please refer to "R-30iA/R-30iA Mate controller Ethernet Function OPERATOR'S MANUAL" (B-82974EN) for more detail.
- 3 PC needs setup to communicate by TCP/IP.
If fixed IP address is available and Windows XP is installed to your PC, following procedure may help you. You have to setup TCP/IP network according to your environment.

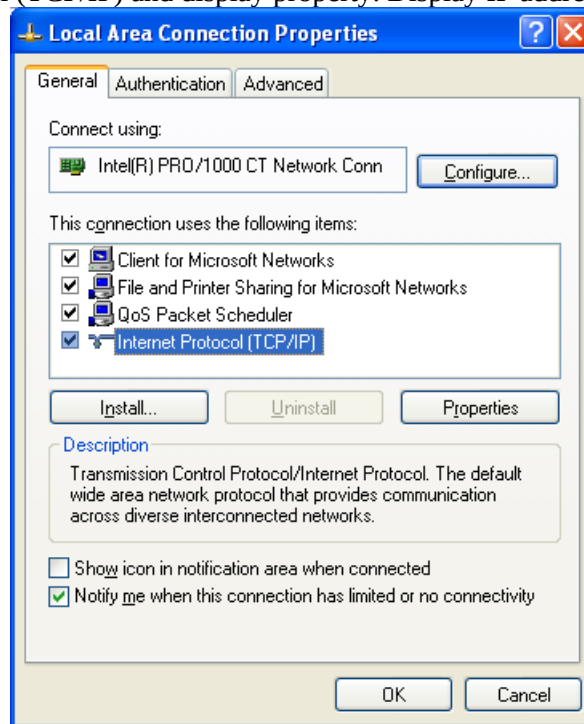
Select Network Connections of control panel.



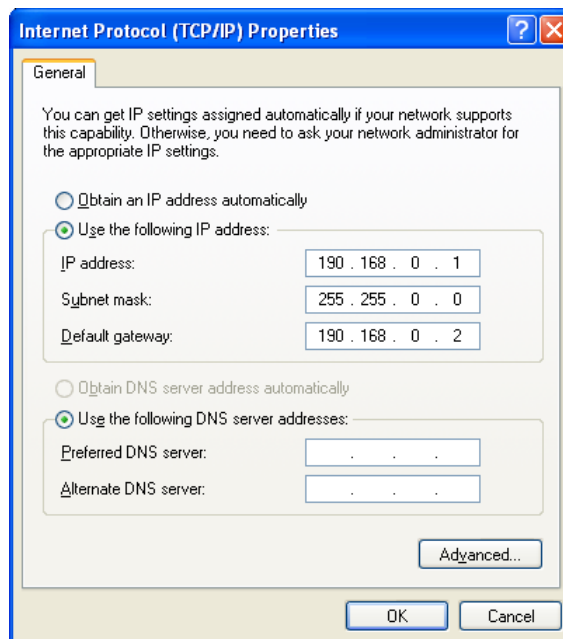
Select Local Area Connection and display property.



Select Internet Protocol (TCP/IP) and display property. Display IP address.



Input IP address.



9.49.1.2 Setup HTTP AUTHENTICATION

Before using ROBOT TOOLS, you have to allow web server to run KAREL program. Otherwise controller requests you to input username and password even though you have not setup them. Please refer to section 6.5 “HTTP AUTHENTICATION” of “R-30iA/R-30iA Mate controller Ethernet Function OPERATOR’S MANUAL” (B-82974EN). If you don’t need authentication, display HTTP AUTHENTICATION screen and set “KAREL” to “U” (Unlock).

- 1 Press MENU.
- 2 Select SETUP.

- 3 Press F1[TYPE] and select Host Comm.

Busy	Stop	Hold	Pause			AUTO	JOINT	100%	
Run	I/O	Prod	Stop						
Prog Select									
						1/13			
1 Program select mode:						PNS			
2 Production start method:						UOP			
Production checks:									
3 At home check:						DISABLED			
4 Re. TYPE 2 on toler.:						DISABLED			
						DISABLED			
						DISABLED			
1 iPendant Setup						ide < 100%: DISABLED			
2 BG Logic						< 100%: DISABLED			
3 Resume Offset						DISABLED			
4 Resume Tol.						DISABLED			
5 Stroke limit						DISABLED			
6 Space fnct.						DISABLED			
7 Host Comm						DISABLED			
8 Passwords									
9									
0 -- NEXT --									
TYPE				DETAIL		[CHOICE]		?	HELP

- 4 Move cursor to HTTP and press F3 DETAIL.

Busy	Stop	Hold	Pause			AUTO	JOINT	100%	
Run	I/O	Prod	Stop						
SETUP Protocols									
						7/8			
Protocol						Description			
1 TCP/IP						TCP/IP Detailed Setup			
2 TELNET						Telnet Protocol			
3 SM						Socket Messaging Device			
4 PROXY						Proxy Server			
5 PPP						Point to Point Protocol			
6 PING						Ping Protocol			
7 HTTP						HTTP Authentication			
8 FTP						File Transfer Protocol			
[TYPE]				DETAIL		[SHOW]			

- 5 Move cursor to head of “KAREL” line and press F3 UNLOCK.

BusyStopHoldFault

RunI/OProdStop

AUTCJOINT100%

HTTP Setup2/8

PROTECTED RESOURCES

	Name	Pwr	Resource
A	*****	*****	iPendant
U	*****	*****	KAREL:*
A	*****	*****	KCL:*
U	*****	*****	VISION SETUP
A	*****	*****	*****
A	*****	*****	*****
A	*****	*****	*****
A	*****	*****	*****

[TYPE]

LOCK

UNLOCK

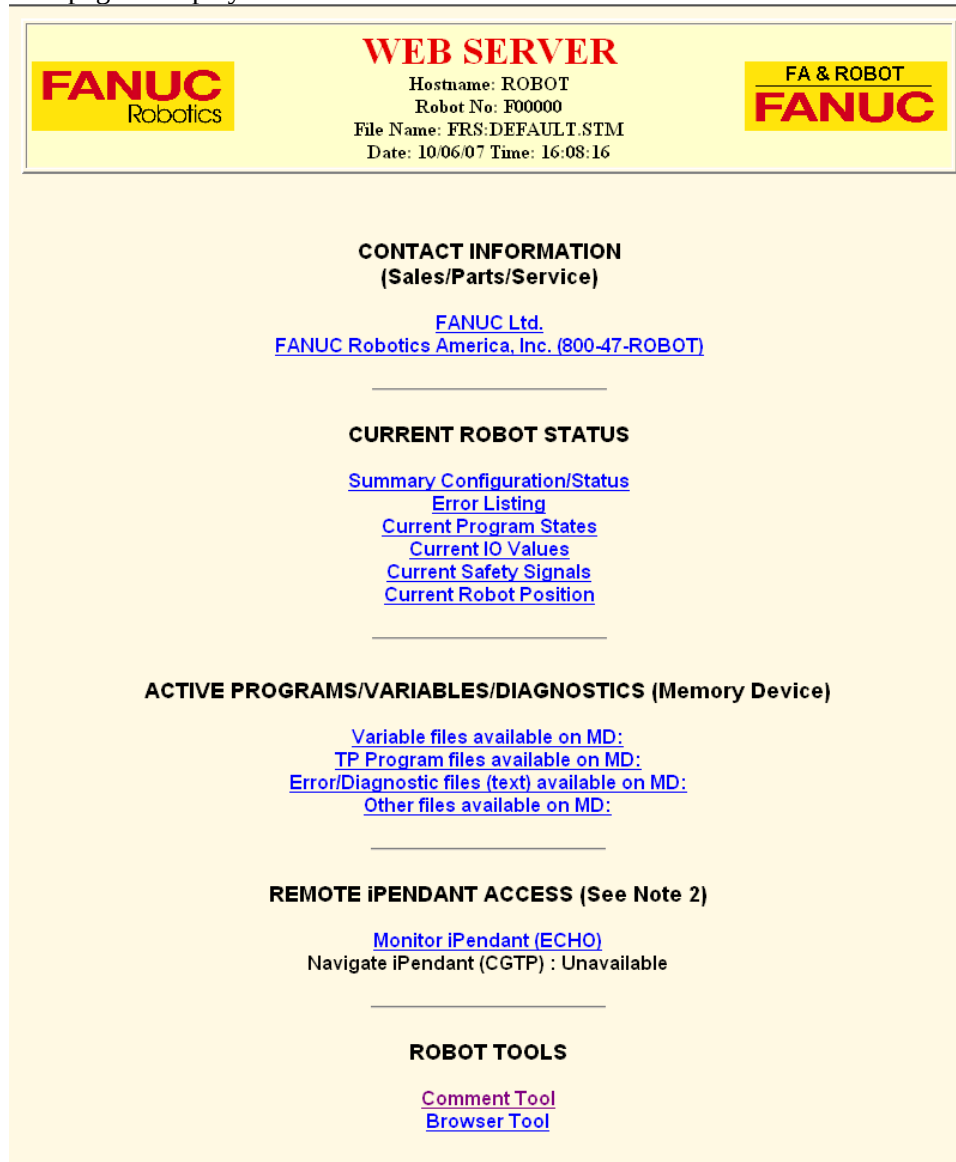
AUTH

?

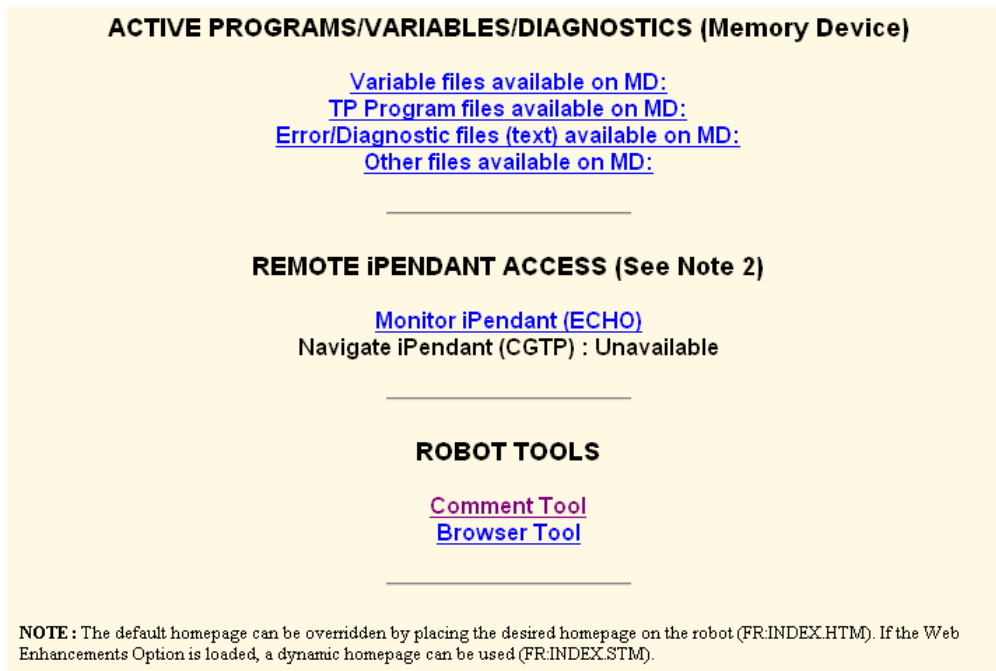
HELP

9.49.2 Usage of ROBOT TOOLS

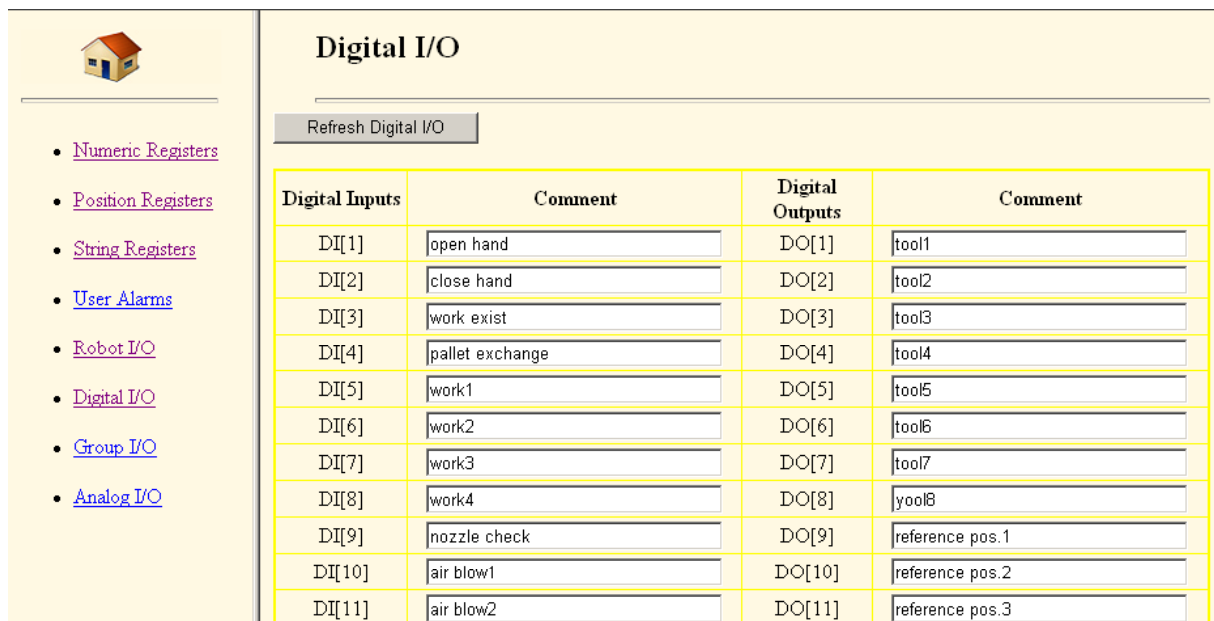
- 1 Invoke Internet Explorer.
- 2 Set “http://(IP address of ROBOT)/” to URL and display it.
Robot home page is displayed.



The links on the default home page called “ROBOT TOOLS” allow you to enter text information easily using a keyboard.

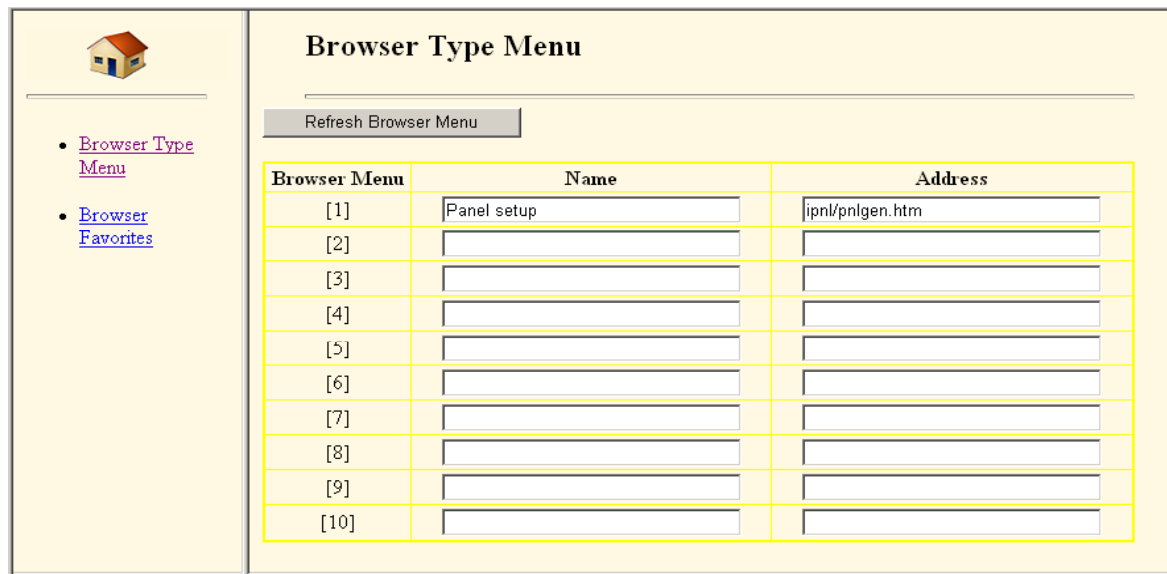


“Comment Tool” will split the screen as shown in following figure. The left side links determine the type of robot information displayed on the right side. The data is retrieved from the robot controller but it is not dynamically updated so you must press the “Refresh” button if you want to refresh the contents of the page. When you are done entering text in the box, click the mouse outside the box or press the tab key. The new data is immediately entered to the robot. When you wish to return home, press the HOME icon on the upper left of the screen.



Digital Inputs	Comment	Digital Outputs	Comment
DI[1]	open hand	DO[1]	tool1
DI[2]	close hand	DO[2]	tool2
DI[3]	work exist	DO[3]	tool3
DI[4]	pallet exchange	DO[4]	tool4
DI[5]	work1	DO[5]	tool5
DI[6]	work2	DO[6]	tool6
DI[7]	work3	DO[7]	tool7
DI[8]	work4	DO[8]	tool8
DI[9]	nozzle check	DO[9]	reference pos.1
DI[10]	air blow1	DO[10]	reference pos.2
DI[11]	air blow2	DO[11]	reference pos.3

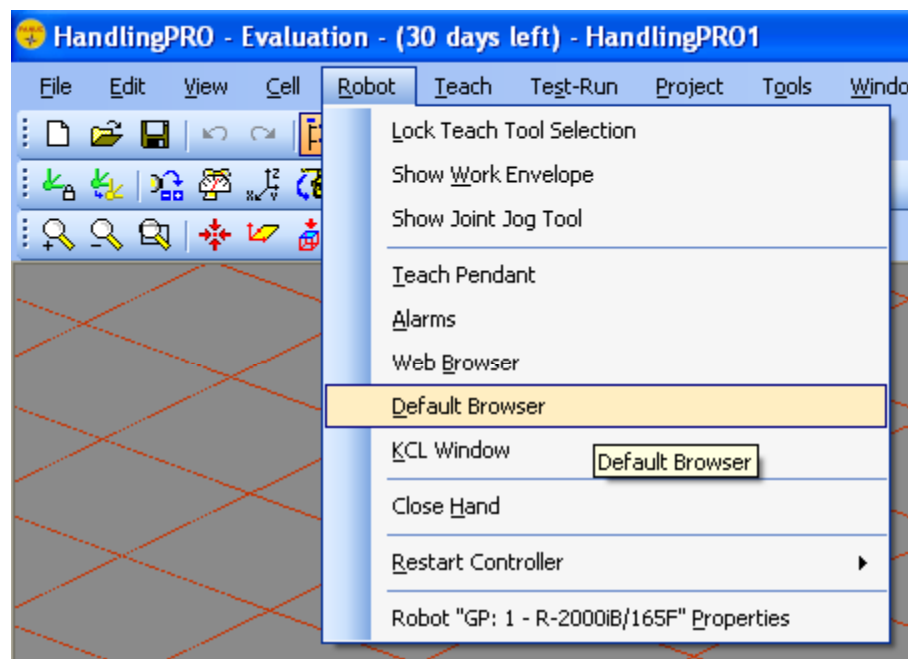
“Browser Tool” will split the screen as shown in following figure.



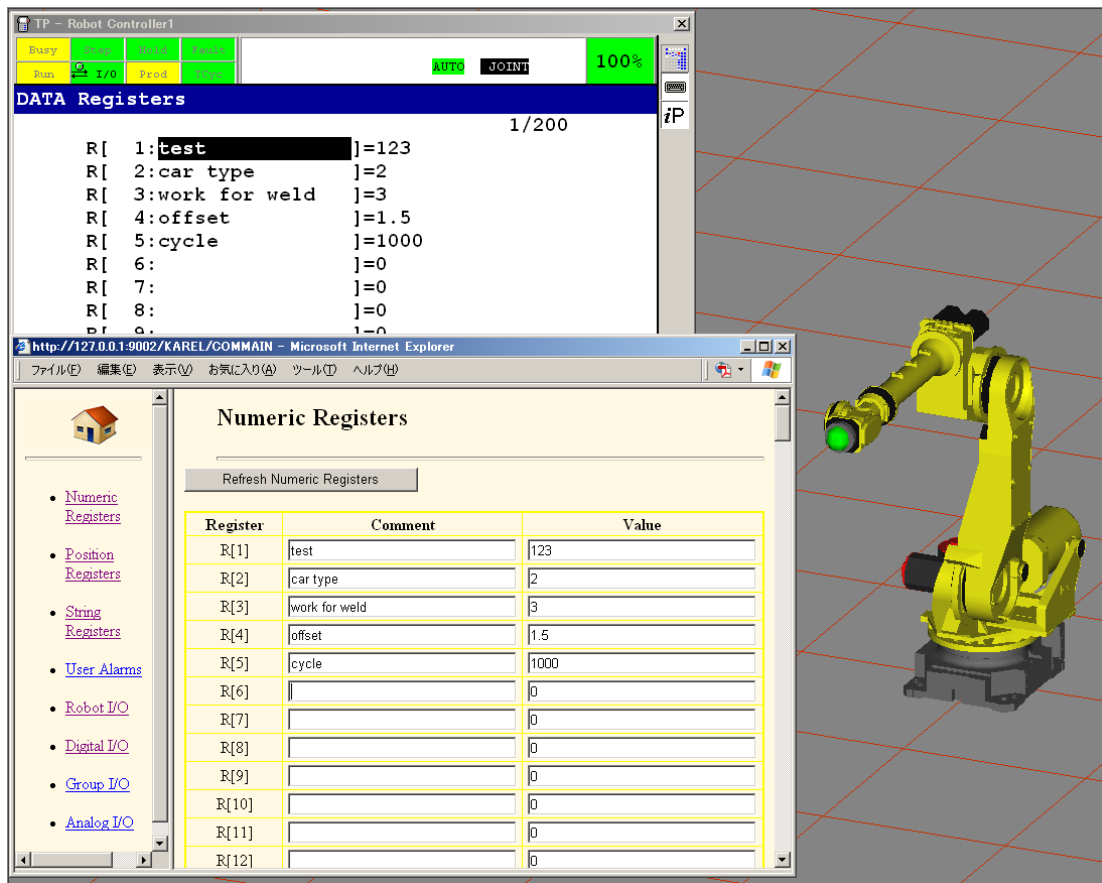
Browser Menu	Name	Address
[1]	Panel setup	ipnl/pnlgen.htm
[2]		
[3]		
[4]		
[5]		
[6]		
[7]		
[8]		
[9]		
[10]		

Browser Type Menu contains the menu entries within the BROWSER menu. Browser Favorites allow you to add links to the FAVORITE page.

To display ROBOT TOOLS of virtual robot controller by ROBOGUIDE, select “Robot” of menu bar. Then select “Default Browser” from displayed menu.



You can display pages of ROBOT TOOLS like following figure.



9.50 LINE REMARK INSTRUCTION

9.50.1 Overview

The Line Remark instruction allows you to disable the execution and editing of a line (comment out) in a program. Any instruction can be remarked.

Note

The Line Remark instruction can be used in the version 7DA7/05 or later.

9.50.2 Function

The Line Remark instruction has the following feature.

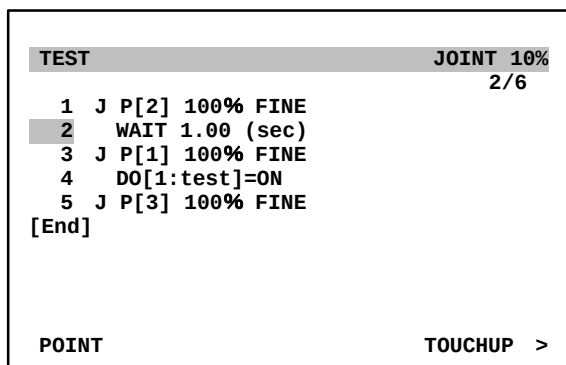
- The Line Remark instruction will prefix the line with “/” and then display the line as normal.
- You can remark or unremark lines by selecting “REMARK” in [EDCMD].
- When the language is changed, the Line Remark instruction is localized.
- The line is still valid even though it is commented out. In other words, a position still exists for the motion lines.
- You can copy or delete a remarked instruction by the command in [EDCMD].
- You can save/load TP and ASCII programs with Line Remark instruction. To load ASCII programs, ASCII UPLOAD option is necessary.
- The toolbars in the icon editor have a Line Remark toggle button. When pressed, the current line is toggled as a remark/unremark.

9.50.3 Teach LINE REMARK instruction

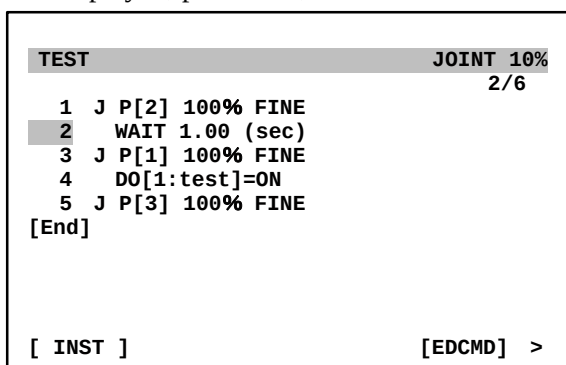
Procedure 9-63 Teach LINE REMARK instruction

Step

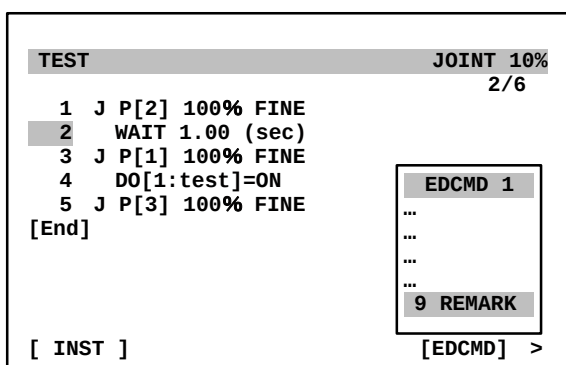
1. Move the cursor on the line number that you want to remark or unremark. If you want to remark or unremark more than one line, move the cursor on the top or end of the section that you want to change.



2. When F5, [EDCMD] is not displayed, press next, ">".



3. Press F5, [EDCMD].



4. Select "REMARK".

```

TEST                                JOINT 10%
                                2/6
1 J P[2] 100% FINE
2 WAIT 1.00 (sec)
3 J P[1] 100% FINE
4 DO[1:test]=ON
5 J P[3] 100% FINE
[End]

Select lines to remark or unremark
                                REMARK  UNREMARK

```

5. If you want to select more than one line, press UP or DOWN key. Then, selected line numbers are highlighted.

```

TEST                                JOINT 10%
                                2/6
1 J P[2] 100% FINE
2 WAIT 1.00 (sec)
3 J P[1] 100% FINE
4 DO[1:test]=ON
5 J P[3] 100% FINE
[End]

Select lines to remark or unremark
                                REMARK  UNREMARK

```

6. Select “REMARK” to remark or “UNREMARK” to unremark.

```

TEST                                JOINT 10%
                                2/6
1 J P[2] 100% FINE
2 //WAIT 1.00 (sec)
3 //J P[1] 100% FINE
4 //DO[1:test]=ON
5 J P[3] 100% FINE
[End]

[ INST ]                                [EDCMD] >

```

Note

When you remark lines, any lines already remarked are skipped. When you unremark lines, any lines not remarked are skipped.

9.50.4 Edit REMARKED instructions

You can edit a remarked instruction in the same way as a normal instruction except execution and edit by key input.

9.50.4.1 [EDCMD] with LINE REMARK

- **Copy**
You can copy or paste a remarked instruction. Then, the information that the instruction has is also copied. For example, a position still exists for the motion line, and when label is copied, “*” is attached to its label number.
- **Find**
You can find a remarked instruction in the same way as a normal instruction.
- **Replace**
You can replace a remarked instruction in the same way as a normal instruction.
- **Renumber**
The position number in a remarked motion line is also renumbered.
Renumber in the following example.

TEST		JOINT 10%
		2/6
1	J P[2] 100% FINE	
2	//WAIT 1.00 (sec)	
3	//J P[3] 100% FINE	
4	//DO[1:test]=ON	
5	J P[1] 100% FINE	
[End]		
Renumber OK?		
		YES NO

Select F4, “YES” and the position number in the motion lines including remarked ones are all renumbered.

TEST		JOINT 10%
		2/6
1	J P[1] 100% FINE	
2	//WAIT 1.00 (sec)	
3	//J P[2] 100% FINE	
4	//DO[1:test]=ON	
5	J P[3] 100% FINE	
[End]		
[INST]		[EDCMD] >

- **Comment**
You can select whether the comment for I/O is displayed or not in the same way as a normal instruction.

9.50.4.2 ASCII program with LINE REMARK

The programs with LINE REMARK can be saved as ASCII programs and also loaded. In the ASCII program, remarked line is described with “//” as well as in EDIT screen.

9.50.5 Consideration

- You cannot create a line with “//”. A line is created in the normal way. You have to press [EDCMD] Remark to convert a line to Line Remark.
- You cannot use “//” to comment out just a portion of a line.
- Right arrow in the TP Editor will not move the cursor off the remarked line number.
- TOUCHUP on a remarked motion line is ignored.
- When LABEL is remarked, “*” is attached to the LABEL number, for example, “//LBL[*1]”. When the line is unremarked, the label number can be changed back to LBL[1] automatically. In the case where user has already created an alternative LBL[1], then it remains LBL[*1].

9.51 FINISHING FUNCTION PACKAGE

Finishing Function Package (J558) is package of two optional software of Special Jog Sequence (J559) and Motion Instruction Enhanced Editing.

Special Jog Sequence and Motion Instruction Enhanced Editing are described below.

Note

Finishing Function Package can be used only in handling tool and LR handling tool.

9.51.1 Special Jog Sequence (J559)

This function enables Jog operation without pressing SHIFT key.

Pressing only Jog keys, the robot moves in low speed. So you can confirm the direction the robot moves in under safe condition.

Normal Jog operation is also possible by pressing SHIFT key additionally.

It is possible to order this function independently.

In that case, this function is available in all tools.

9.51.1.1 Usage

When this function is valid, Jog operation is possible in the following conditions.

Press only Jog keys:

The robot moves in the speed corresponds to override 5%. When the override is less than 5%, the robot moves in the speed corresponds to the override.

Press both Jog keys and SHIFT key:

The robot moves in the normal Jog speed. The Jog speed corresponds to the override specified in Teach Pendant.

When this function is invalid, Jog operation is possible in the following conditions.

Press only Jog keys:

The robot does not move and the following message is displayed.

“JOG-007 Press shift key to jog”

Press both Jog keys and SHIFT key:

The robot moves in the normal speed. The Jog speed corresponds to the override specified by Teach Pendant.

9.51.1.2 Setup

This function can be set up by the following system variable.

\$JOG_SP_MODE:

0: This function is invalid.

1-100: This function is valid.

When only Jog key is pressed, the robot moves in the speed corresponds to the override specified here.



WARNING

If large value is specified in this system variable, the robot possibly moves in high speed. For secure use, the value of this system variable should be less than 5.

9.51.1.3 Application

Automatic Override setup when changing frames:

Normally, override is automatically initialized to 10% when Jog frame is changed in Teach Pendant. With this function, as you can confirm the direction in low speed, you may want to make this automatic override setup invalid.

Setting the following system variable, the current override can be kept when frame is changed.

\$SCR.\$COORDOVRD = 100 (Standard value 10)

When the current override is larger than the value specified here, the override is changed into the value specified here.

9.51.2 Motion Instruction Enhanced Editing (J560)

This function includes two functions, “Motion instruction insert and instruction delete function” and “Data offset function”. These are described below.

It is possible to order this function independently.

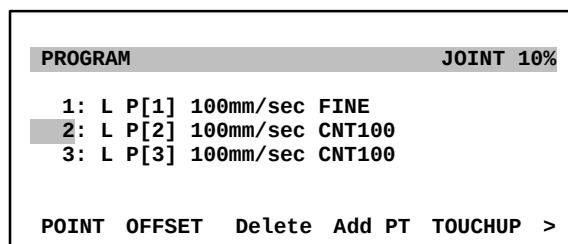
In that case, this function is available only in handling tool and LR handling tool.

9.51.2.1 Motion Instruction Insert and Instruction delete function

This function enables you to insert motion instructions or delete instructions in a program with function keys in the edit screen.

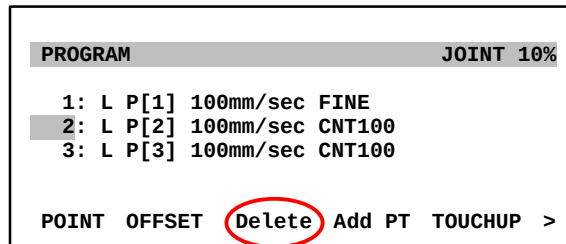
Usage

Function keys are displayed as follows in EDIT screen.

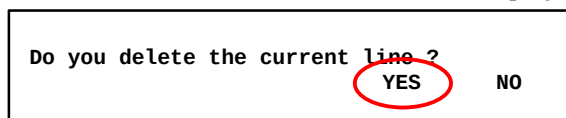


Procedure 9-64 Delete instructions

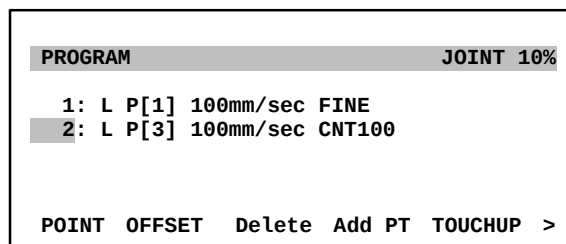
1. Move the cursor on the line you want to delete.
2. Enable Teach Pendant, and press SHIFT and F3, Delete.



3. F4, YES and F5, NO are displayed. Press YES to delete.
When NO is pressed, the deleted is cancelled and edit screen is displayed.

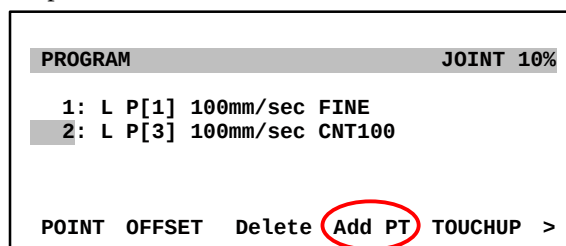


4. The line that the cursor was on is deleted.

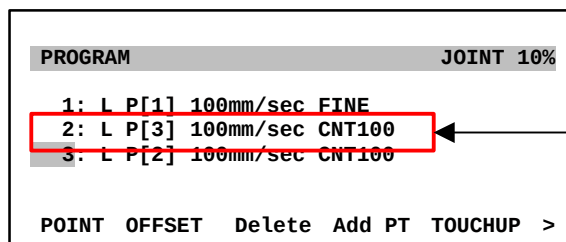


Procedure 9-65 Insert motion instructions

1. Move the cursor on the line you want to insert.
2. Enable Teach Pendant, and press SHIFT and F4, Add PT.



3. Motion line is inserted, and the cursor moves to the next line.



← This motion instruction is inserted.

Precautions

- Standard motion instruction that is selected in F1, POINT menu is inserted. This is the same motion instruction that is taught by pressing SHIFT key and POINT.
- When Circular motion is selected as standard motion instruction, Circular motion is inserted. In this case, current position is recorded in passing point and the target point is unspecified.
- After operations with this function, you cannot undo the operation by Undo in [EDCMD].
- This function cannot be used in quick menu.

9.51.2.2 Data Offset function

Data offset function enables you to change multiple position data or motion speed at once in a program that you are editing. This function includes 3 functions, “Feed rate conversion”, “Position data conversion (direct method)”, and “Position data conversion (2 point teach method)”.

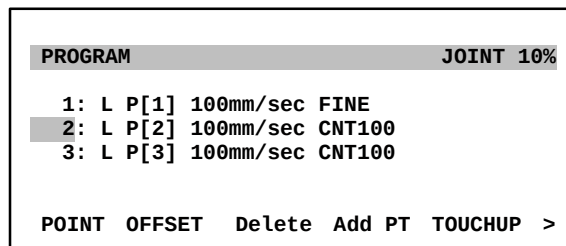
Feed Rate Conversion

- This function converts the feed rate of multiple motion instructions at once.
- You can convert a whole or part of the program you are editing.
- You can specify the offset to current feed rate.
- You can convert the feed rate of all motion instructions into specified value regardless of current feed rate.
- You can convert the motion instructions whose current feed rate is equal to specified value into the value you want to give.
- You can specify offset rate (%) to current feed rate.
- Except offset rate specification, only one motion type of Joint, Linear or Circular motion can be converted at once.

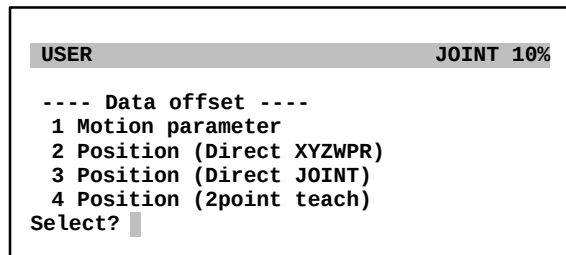
Procedure 9-66 Feed rate conversion

Step

1. Press F2, OFFSET in EDIT screen.



2. Input “1” to select “1 Motion parameter”.



3. To convert a whole program, input “1”. Then go to step 5.
To convert a part of a program, input “2”. Then go to step 4.

```

USER JOINT 10%
<< Motion parameter >>

---- Program range ----
1 Whole program
2 Part of the program
Select? █

```

4. Input start line number and last line number.

```

USER JOINT 10%
<< Motion parameter >>

---- Program range ----
1 Whole program
2 Part of the program
Select? 2

Start line number:

Last line number:

```

5. Select target motion type or All (Rate specify).
If you select All (Rate specify), go to step 11.

```

USER JOINT 10%
<< Motion parameter >>

---- Target motion type ----
1 Joint
2 Linear
3 Circular
4 All (Rate specify)
Select? █

```

6. Select the method for converting feed rate.
Input "1" to specify the offset to the current feed rate. Then go to step 7.
Input "2" to convert feed rate to specified value. Then go to step 8.
Input "3" to specify the rate to the current feed rate. Then go to step 11.

```

USER JOINT 10%
<< Motion parameter >>

---- Method ----
1 Offset
2 Absolute
3 Rate
Select? █

```

7. Input the offset. When you want to decrease the value, input negative value.
The unit is % when Joint motion or mm/sec when Linear motion or Circular motion.
Go to step 12.

```

USER JOINT 10%
<< Motion parameter >>

Offset(%): █

```

8. Input "1" to convert all the motion instructions. Then go to step 10.
Input "2" to convert only the motion instructions whose current feed rate is equal to the value specified by operation 9. Then go to step 9.

USER	JOINT 10%
<< Motion parameter >>	
Old data	
1 All	
2 Specify	
Select? █	

9. Input current feed rate of the motion instructions that you convert.
The unit is % when Joint motion or mm/sec when Linear or Circular motion.

USER	JOINT 10%
<< Motion parameter >>	
Old data(%): █	

10. Input new value of feed rate.
The unit is % when Joint motion or mm/sec when Linear or Circular motion.
Go to step 12.

USER	JOINT 10%
<< Motion parameter >>	
New data(%): █	

11. Specify the rate (-100 to 100).

USER	JOINT 10%
<< Motion parameter >>	
Rate(%): █	

12. Input "1" to convert the program.
Input "0" to cancel the conversion.

USER	JOINT 10%
<< Motion parameter >>	
Convert the program? (Yes=1/No=0) : █	

13. Edit screen will be displayed after conversion.

Position Data Conversion (direct method)

Position data conversion (Direct method) converts multiple position data in the program at once. Translation is specified by value.

- You can convert a whole or part of the program you are editing.
- You can specify the offset to current position data.

- You can convert the position data of all motion instructions into specified value regardless of current position data.
- You can convert the motion instructions whose current position data is equal to specified value into the value you want.
- You can specify the number of user frame to specify the value.
- You can select the type, Cartesian (X, Y, Z, W, P, R) or Joint (J1 to J9).
- Only one element of X, Y, Z, W, P, R or J1 to J9 can be converted at once.

Procedure 9-67 Position data conversion (direct method)

Step

1. Press F2, OFFSET in EDIT screen.

```

PROGRAM                                JOINT 10%

  1: L P[1] 100mm/sec FINE
  2: L P[2] 100mm/sec CNT100
  3: L P[3] 100mm/sec CNT100

POINT  OFFSET  Delete  Add PT  TOUCHUP  >

```

2. Input "1" to select "2 Position (Direct XYZWPR).
Input "2" to select "3 Position (Direct JOINT).

```

USER                                JOINT 10%

---- Data offset ----
  1 Motion parameter
  2 Position (Direct XYZWPR)
  3 Position (Direct JOINT)
  4 Position (2point teach)
Select? █

```

3. Input "1" to convert a whole program. Then go to step 5.
Input "2" to convert a part of a program. Then go to step 4.

```

USER                                JOINT 10%
<< Position (Direct XYZWPR) >>

---- Program range ----
  1 Whole program
  2 Part of the program
Select? █

```

4. Input start line number and last line number.

```

USER                               JOINT 10%
<< Position (Direct XYZWPR) >>

---- Program range ----
  1 Whole program
  2 Part of the program
Select? 2

Start line number:

Last line number:

```

5. Input the number of the element or axis number to convert.
When "Direct XYZWPR" is selected

```

USER                               JOINT 10%
<< Position (Direct XYZWPR) >>

---- Target element ----
  1 X    2 Y    3 Z
  4 W    5 P    6 R
Select? 

```

When "Direct JOINT" is selected

```

USER                               JOINT 10%
<< Position (Direct XYZWPR) >>

---- Program axis ----
Enter axis number? 

```

6. Input "1" to specify the offset to current position data. Then go to step 7.
Input "2" to specify the position data into specified value. Then go to step 8.

```

USER                               JOINT 10%
<< Position (Direct XYZWPR) >>

---- Method ----
  1 Offset
  2 Absolute
Select? 

```

7. Input the offset. When you want to decrease the value, input negative value.
The unit is mm when X, Y, Z or deg when W, P, R.
Go to step 11.

```

USER                               JOINT 10%
<< Position (Direct XYZWPR) >>

Offset(mm): 

```

8. Input "1" to convert all the position data. Then go to step 10.
Input "2" to convert only the motion instructions whose current position data is equal to the value specified by step 9. Then go to step 9.

```

USER                               JOINT 10%
<< Position (Direct XYZWPR) >>

Old data
 1 All
 2 Specify
Select? █

```

9. Input the current position data of the motion instruction that you convert.
The unit is mm when X, Y, Z or deg when W, P, R.

```

USER                               JOINT 10%
<< Position (Direct XYZWPR) >>

Old data(mm): █

```

10. Input new position data.
The unit is mm when X, Y, Z or deg when W, P, R.

```

USER                               JOINT 10%
<< Position (Direct XYZWPR) >>

New data(mm): █

```

11. Input the user frame number that is the base of conversion.
When world frame is used, input "0".

```

USER                               JOINT 10%
<< Position (Direct XYZWPR) >>

User frame number(World=0): █

```

12. Input "1" to convert the program.
Input "0" to cancel the conversion.

```

USER                               JOINT 10%
<< Position (Direct XYZWPR) >>

Convert the program? (Yes=1/No=0) : █

```

13. Edit screen will be displayed after conversion.

Position Data Conversion (2 point teach method)

Position data conversion (2 point teach method) converts multiple position data in the program at once. Translation is specified by teaching 2 point, the original position and the destination position.

- You can convert a whole or part of the program you are editing.
- Parallel element (X, Y, Z) is converted on world frame and rotation element (W, P, R) is converted on tool frame.

Procedure 9-68 Position data conversion (2 point teach method)

Step

1. Press F2, OFFSET in EDIT screen.

```

PROGRAM                                JOINT 10%

  1: L P[1] 100mm/sec FINE
  2: L P[2] 100mm/sec CNT100
  3: L P[3] 100mm/sec CNT100

POINT  OFFSET  Delete  Add PT  TOUCHUP  >

```

2. Input "4" to select "4 Position (2point teach)".

```

USER                                JOINT 10%

---- Data offset ----
  1 Motion parameter
  2 Position (Direct XYZWPR)
  3 Position (Direct JOINT)
  4 Position (2point teach)
Select? █

```

3. Input "1" to convert a whole program. Then go to step 5.
Input "2" to convert a part of a program. Then go to step 4.

```

USER                                JOINT 10%
<< Position (2point teach) >>

---- Program range ----
  1 Whole program
  2 Part of the program
Select? █

```

4. Input start line number and last line number.

```

USER                                JOINT 10%
<< Position (2point teach) >>

---- Program range ----
  1 Whole program
  2 Part of the program
Select? 2

Start line number:
Last line number:

```

5. Input "1" after moving the robot to the original position.


```

USER                               JOINT 10%
<< Position (2point teach) >>

Move robot to the original position
then enter 1: █

```

6. Input "1" after moving the robot to the destination position.

```

USER                               JOINT 10%
<< Position (2point teach) >>

Move robot to the destination position
then enter 1: █

```

7. Each translation is displayed.
 The value X, Y, Z is the translation on world frame.
 The value W, P, R is the translation on tool frame.
 Input "1" to convert the program.
 Input "0" to cancel the conversion.

```

USER                               JOINT 10%
<< Position (2point teach) >>
Translation is the following
(XYZ:World.WPR:Tool)
X:    32.167
Y:    -2.409
Z:   105.409
W:    -0.034
P:     1.045
R:     2.340

Convert the program? (Yse=1/No=0) : █

```

8. Edit screen will be displayed after conversion.

Precautions

- The original program is copied to YFDTOFBK. When you want to use the original program, copy YFDTOFBK.
- A program whose name is YFDTOFBK is overwritten after conversion. Please do not create YFDTOFBK.
- In the multi group system, only the position data of first group can be converted.
- When the position data is converted in a system with extended axes, only the position data of robot axes are converted. The position data of extended axes are not converted.
- When another screen is displayed (ex current position display) during the operation of data offset function, you can return to the operation of data offset function by pressing EDIT key, or selecting "9 USER" in MENU.
- Press F1, "ABORT" when you want to abort data offset function and return to EDIT screen. When you press F2, "OFFSET" again after this operation, you can restart data offset function from the beginning.
- When teach pendant is disabled, data offset function does not work by pressing F2, "OFFSET". When teach pendant is turned disabled during the operation of data offset function, the operation is aborted and EDIT screen is displayed.
- When the position data is singular point or out of stroke limit after conversion, the position data will be unspecified data. In this case, the number of the unspecified data will be displayed and displayed

as follows “Error occurred(1:OK 0:Cancel)”. When you select 1, the position data that have no error will be normally converted and the position data that have errors will be unspecified data. When you select 0, the conversion will be canceled.

- In conversion of position data by “Position (Direct XYZWPR)” or “Position (2point teach)”, the amount of the change in all axes is automatically adjusted not to exceed plus and minus 180 degree.

9.52 FOR/ENDFOR STATEMENT

FOR/ENDFOR statement is a function that repeats a loop within FOR and ENDFOR statements specified times.

Note

FOR/ENDFOR statement can be used in the version 7DA7/14 or later.

9.52.1 Overview

FOR/ENDFOR statement has two instructions, FOR statement and ENDFOR statement.

- FOR statement start of FOR/ENDFOR loop
- ENDFOR statement end of FOR/ENDFOR loop

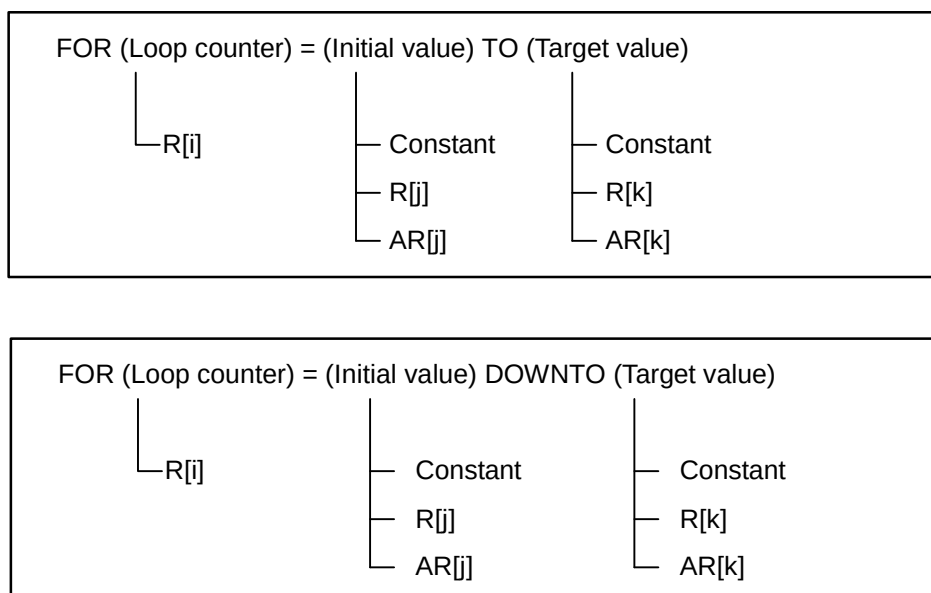
Instructions within FOR and ENDFOR statements are repeated.

The number of times to repeat is determined by the specified values in FOR statement.

9.52.2 FOR/ENDFOR statement specification

9.52.2.1 Form of FOR statement

The form of FOR statement is as follows.



Register is used for Loop counter.

Constant, register, argument register is used for Initial value. For constant, integer whose range is –32767 to 32767 can be specified.

Constant, register, argument register is used for Target value. For constant, integer whose range is –32767 to 32767 can be specified.

9.52.2.2 FOR statement execution

When FOR statement is executed, Initial value is substituted into Loop counter.

The following condition should be satisfied to execute FOR/ENDFOR loop.

- When TO is specified, Initial value is equal to or smaller than Target value.
- When DOWNTTO is specified, Initial value is equal to or larger than Target value.

When this condition is not satisfied, the cursor moves to the next line of combined ENDFOR statement and FOR/ENDFOR loop is not executed.

FOR statement is executed only once in the FOR/ENDFOR loop.

9.52.2.3 Form of ENDFOR statement

The form of ENDFOR statement is as follows.

ENDFOR

9.52.2.4 ENDFOR statement execution

FOR/ENDFOR loop is repeated as long as the following condition is satisfied.

- When TO is specified, the value of Loop counter is smaller than Target value.
- When DOWNTTO is specified, the value of Loop counter is larger than Target value.

When this condition is satisfied and TO is specified, the value of Loop counter is incremented. When this condition is satisfied and DOWNTTO is specified, the value of Loop counter is decremented. And after that, the cursor moves to the next line of combined FOR statement.

When this condition is not satisfied, the cursor moves to next line and FOR/ENDFOR loop is not repeated.

Note

An internal delay is used when FOR/ENDFOR loop is executed repeatedly. Therefore, WAIT statement is not necessary in FOR/ENDFOR loop to repeat the loop.

9.52.3 Teach FOR/ENDFOR statement

9.52.3.1 How to teach FOR/ENDFOR statement

How to teach FOR/ENDFOR statement is as follows.

Procedure 9-69 Teach FOR/ENDFOR statement

Step

1. In this example, insert blank lines in line 1 and 5.

```

Program                                JOINT 10%
1:
2: L P[2] 100mm/sec CNT100
3: L P[3] 100mm/sec CNT100
4: L P[1] 100mm/sec FINE
5:
6: L P[3] 100mm/sec CNT100

[ INST ]                                [EDCMD] >

```

2. Move the cursor to line 1 and press F1, [INST].
3. Select FOR/ENDFOR.

```

Program                                JOINT 10%
1:
2: L P[2] 100mm/sec CNT100
3: L P[3] 100mm/sec CNT100
4: L P[1] 100mm/sec FINE
5:
6: L P[3] 100mm/sec CNT100

[ INST ]                                [EDCMD] >

```

Instruction
FOR/ENDFOR

4. Select FOR. FOR statement is taught.

```

Program                                JOINT 10%
1: FOR R[...]=... TO ...
2: L P[2] 100mm/sec CNT100
3: L P[3] 100mm/sec CNT100
4: L P[1] 100mm/sec FINE
5:
6: L P[3] 100mm/sec CNT100

[ INST ]                                [EDCMD] >

```

FOR statement
FOR
ENDFOR

5. The cursor automatically moves to the index of the register. Input the value of the index. In this example, input 1.

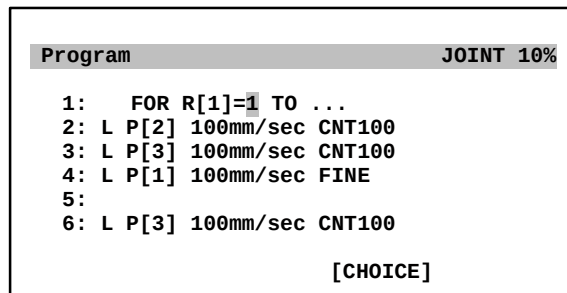
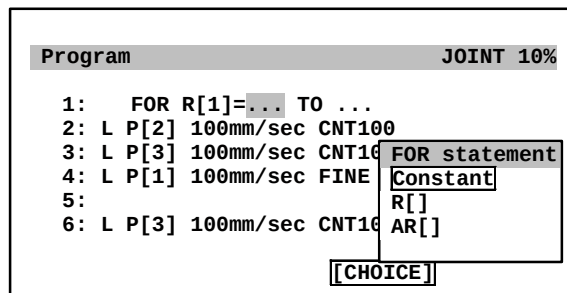
```

Program                                JOINT 10%
1: FOR R[...]=... TO ...
2: L P[2] 100mm/sec CNT100
3: L P[3] 100mm/sec CNT100
4: L P[1] 100mm/sec FINE
5:
6: L P[3] 100mm/sec CNT100

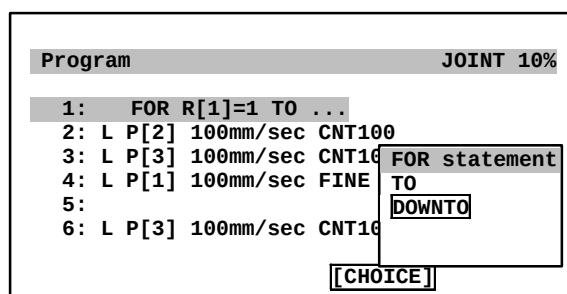
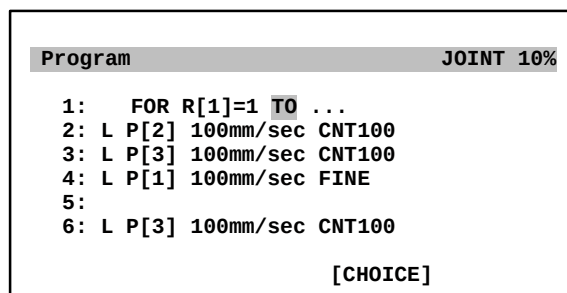
DIRECT INDIRECT [CHOICE]

```

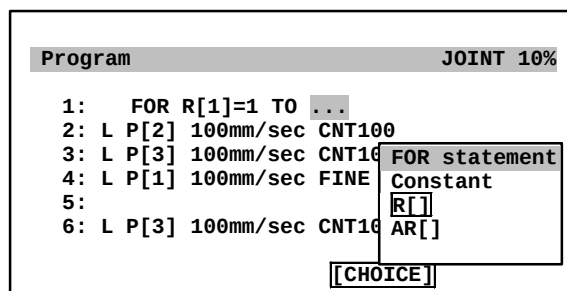
6. The cursor automatically moves to Initial value part. In this example, Constant is selected and the value is 1. To choose register or argument register, press F4, [CHOICE] and select one of them.



7. The cursor automatically moves to TO/DOWNTTO part. In this example, press ENTER. To choose DOWNTTO, press F4, [CHOICE] and select DOWNTTO.



8. The cursor automatically moves to Target value part. In this example, select R[]. Input 2 for the index of the register.



```

Program                                JOINT 10%

1:  FOR R[1]=1 TO R[2]
2:  L P[2] 100mm/sec CNT100
3:  L P[3] 100mm/sec CNT100
4:  L P[1] 100mm/sec FINE
5:
6:  L P[3] 100mm/sec CNT100

DIRECT INDIRECT [CHOICE]

```

9. Move the cursor to line 5 and press F1, [INST].
10. Select FOR/ENDFOR.

```

Program                                JOINT 10%

1:  FOR R[1]=1 TO R[2]
2:  L P[2] 100mm/sec CNT100
3:  L P[3] 100mm/sec CNT100
4:  L P[1] 100mm/sec FINE
5:
6:  L P[3] 100mm/sec CNT100

[ INST ]                                [EDCMD] >

```

Instruction
FOR/ENDFOR

11. Select ENDFOR. ENDFOR statement is taught.

```

Program                                JOINT 10%

1:  FOR R[1]=1 TO R[2]
2:  L P[2] 100mm/sec CNT100
3:  L P[3] 100mm/sec CNT100
4:  L P[1] 100mm/sec FINE
5:  ENDFOR
6:  L P[3] 100mm/sec CNT100

[ INST ]                                [EDCMD] >

```

FOR statement
FOR
ENDFOR

9.52.3.2 FOR/ENDFOR statement Combination

FOR and ENDFOR statements are automatically combined after teaching. The closest FOR statement and ENDFOR statement is orderly combined.

By teaching FOR/ENDFOR statements additionally in FOR/ENDFOR loop, nested loops can be formed. Up to 10 nested loops can be formed. However, teaching more than 10 nested loops causes an alarm in execution.

The number of FOR statement and ENDFOR statement in a program should be the same. When the number is not the same, an alarm occurs in execution. Alarms related to FOR/ENDFOR function is described in section 9.52.5.

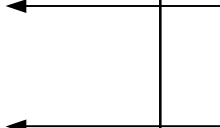
How to combine FOR and ENDFOR statements is described in the following example.

Teach FOR statement in line 1. In this case, the number of FOR statement and ENDFOR statement is not the same. Therefore, if this program is executed, the alarm "INTP-670 Need ENDFOR for FOR in line 1" occurs.

Program	JOINT 10%
1: FOR R[1]=1 TO 5	
2: L P[1] 100mm/sec CNT100	
3:	
4: L P[2] 100mm/sec FINE	
5: L P[3] 100mm/sec CNT100	
6:	
7: L P[4] 100mm/sec CNT100	
8:	
[INST]	[EDCMD] >

Teach ENDFOR statement in line 6. FOR/ENDFOR loop is formed by FOR statement in line 1 and ENDFOR statement in line 6.

Program	JOINT 10%
1: FOR R[1]=1 TO 5	
2: L P[1] 100mm/sec CNT100	
3:	
4: L P[2] 100mm/sec FINE	
5: L P[3] 100mm/sec CNT100	
6: ENDFOR	
7: L P[4] 100mm/sec CNT100	
8:	
[INST]	[EDCMD] >

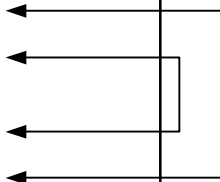


Teach FOR statement in line 3. In this case, the number of FOR statement and ENDFOR statement is not the same. Therefore, if this program is executed, the alarm “INTP-670 Need ENDFOR for FOR in line 1” occurs.

Program	JOINT 10%
1: FOR R[1]=1 TO 5	
2: L P[1] 100mm/sec CNT100	
3: FOR R[2]=5 DOWNT0 1	
4: L P[2] 100mm/sec FINE	
5: L P[3] 100mm/sec CNT100	
6: ENDFOR	
7: L P[4] 100mm/sec CNT100	
8:	
[INST]	[EDCMD] >

Teach ENDFOR statement in line 8. The closest FOR statement and ENDFOR statement is orderly combined. Therefore, the first FOR/ENDFOR loop is formed by FOR statement in line 3 and ENDFOR statement in line 6, and the second FOR/ENDFOR loop is formed by FOR statement in line 1 and ENDFOR statement in line 8. In this example, 2 nested loops are formed.

Program	JOINT 10%
1: FOR R[1]=1 TO 5	
2: L P[1] 100mm/sec CNT100	
3: FOR R[2]=5 DOWNT0 1	
4: L P[2] 100mm/sec FINE	
5: L P[3] 100mm/sec CNT100	
6: ENDFOR	
7: L P[4] 100mm/sec CNT100	
8: ENDFOR	
[INST]	[EDCMD] >



Note

Please be careful not to use the same register number as Loop counters in the same nested loop.
It may cause abnormal behavior.

9.52.4 Execution examples

9.52.4.1 Backward execution

Backward execution is prohibited on FOR/ENDFOR statement.

However, backward execution on instructions within FOR/ENDFOR loop is allowed.

In the following example, when backward execution is started from line 1 or 5, the alarm “INTP-238 BWD execution completed” occurs. On the other hand, backward execution can be done when started from line 2, 3, 4 or 6.

```

Program                                JOINT 10%

1:   FOR R[1]=1 TO R[2]
2: L P[1] 100mm/sec CNT100
3: L P[2] 100mm/sec CNT100
4: L P[3] 100mm/sec FINE
5:   ENDFOR
6: L P[4] 100mm/sec CNT100

[ INST ]                                [EDCMD] >

```

9.52.4.2 Examples

The examples of FOR/ENDFOR statement are described with the following program.

```

Program                                JOINT 10%

1:   FOR R[1]=1 TO R[2]
2: L P[1] 100mm/sec CNT100
3: L P[2] 100mm/sec CNT100
4: L P[3] 100mm/sec FINE
5:   ENDFOR
6: L P[4] 100mm/sec CNT100

[ INST ]                                [EDCMD] >

```

Case 1: R[2]=3

TO is specified and Initial value is smaller than Target value. Therefore, the condition of FOR statement is satisfied.

As the value of Loop counter changes from 1 to 3, the condition of ENDFOR statement is satisfied and FOR/ENDFOR loop is repeated three times.

```

Program                                JOINT 10%

1:   FOR R[1]=1 TO R[2]
2: L P[1] 100mm/sec CNT100
3: L P[2] 100mm/sec CNT100
4: L P[3] 100mm/sec FINE
5:   ENDFOR
6: L P[4] 100mm/sec CNT100

[ INST ]                                [EDCMD] >

```

R[1] changes as follows:

Loop	1 st	2 nd	3 rd
1:	1	-	-
2:	1	2	3
3:	1	2	3
4:	1	2	3
5:	1->2	2->3	3
6:	-	-	3

(NOTE) “-” means the line is not executed.

Case 2: R[2]=1

TO is specified and Initial value is equal to Target value. Therefore, the condition of FOR statement is satisfied.

However, as the value of Loop counter is equal to Target value, the condition of ENDFOR statement is not satisfied.

Therefore, FOR/ENDFOR loop is executed only once.

Program	JOINT 10%
1: FOR R[1]=1 TO R[2]	
2: L P[1] 100mm/sec CNT100	
3: L P[2] 100mm/sec CNT100	
4: L P[3] 100mm/sec FINE	
5: ENDFOR	
6: L P[4] 100mm/sec CNT100	
[INST]	[EDCMD] >

R[1] changes as follows:

Loop	1 st
1:	1
2:	1
3:	1
4:	1
5:	1
6:	1

(NOTE) "-" means the line is not executed.

Case 3: R[2]=0

TO is specified and Initial value is larger than Target value. Therefore, the condition of FOR statement is not satisfied.

The cursor moves to the next line of combined ENDFOR statement, that is, line 6 and FOR/ENDFOR loop is not executed.

Program	JOINT 10%
1: FOR R[1]=1 TO R[2]	
2: L P[1] 100mm/sec CNT100	
3: L P[2] 100mm/sec CNT100	
4: L P[3] 100mm/sec FINE	
5: ENDFOR	
6: L P[4] 100mm/sec CNT100	
[INST]	[EDCMD] >

R[1] changes as follows:

Loop	1 st
1:	1
2:	-
3:	-
4:	-
5:	-
6:	1

(NOTE) "-" means the line is not executed.

Other examples**Start within FOR/ENDFOR loop: R[1]=0**

When the following program is executed from line 3 and R[1]=0, the condition of ENDFOR statement is satisfied. Therefore, FOR/ENDFOR loop is repeated 4 times (0 to 3).

Program	JOINT 10%
1: FOR R[1]=1 TO 3	
2: L P[1] 100mm/sec CNT100	
3: L P[2] 100mm/sec CNT100	
4: L P[3] 100mm/sec FINE	
5: ENDFOR	
6: L P[4] 100mm/sec CNT100	
Start from line 3	[EDCMD] >

R[1] changes as follows:

Loop	1 st	2 nd	3 rd	4 th
1:	-	-	-	-
2:	-	1	2	3
3:	0	1	2	3
4:	0	1	2	3
5:	0->1	1->2	2->3	3
6:	-	-	-	3

(NOTE) "-" means the line is not executed.

Start within FOR /ENDFOR loop: R[1]=5

When the following program is executed from line 3 and R[1]=5, the condition of ENDFOR statement is not satisfied. Therefore, FOR/ENDFOR loop is not repeated.

Program		JOINT 10%
1:	FOR R[1]=1 TO 3	
2:	L P[1] 100mm/sec CNT100	
3:	L P[2] 100mm/sec CNT100	
4:	L P[3] 100mm/sec FINE	
5:	ENDFOR	
6:	L P[4] 100mm/sec CNT100	
Start from line 3		[EDCMD] >

R[1] changes as follows:

Loop	1 st
1:	-
2:	-
3:	5
4:	5
5:	5
6:	5

(NOTE) "-" means the line is not executed.

JMP/LBL statements exist within FOR/ENDFOR loop:

In the following program, JMP/LBL statements exist within the FOR/ENDFOR loop. The cursor moves from line 3 to line 7 by JMP statement. FOR/ENDFOR loop between line 1 and 4 is not executed. As line 7 is also within the FOR/ENDFOR loop and the condition of ENDFOR statement in line 8 is satisfied, FOR/ENDFOR loop between lines 5 and 8 is repeated.

Finally, the value of R[1] is equal to 1 and R[2] is equal to 2.

Program		JOINT 10%
1:	FOR R[1]=1 TO 10	
2:	R[2]=0	
3:	JMP LBL[1]	
4:	ENDFOR	
5:	FOR R[2]=1 TO 2	
6:	L P[2] 100mm/sec CNT100	
7:	LBL[1]	
8:	ENDFOR	
9:	L P[3] 100mm/sec CNT100	
[INST]		[EDCMD] >

R[1] changes as follows:

Loop	1 st	2 nd	3 rd
1:	*	-	-
2:	0	-	-
3:	0	-	-
4:	-	-	-
5:	-	-	-
6:	-	1	2
7:	0	1	2
8:	0->1	1->2	2
9:	-	-	2

(NOTE) "-" means the line is not executed.

Note

Please be careful when you use JMP/LBL within FOR/ENDFOR loop.
Before you use them, please consider how the loop works. Otherwise, it may cause fatal errors.

Loop counter value is modified within FOR/ENDFOR loop:

Loop counter can be modified within the FOR/ENDFOR loops.

In the following example, when DI[1] = ON, R[1] is changed to 11. As this value is larger than Target value, this FOR/ENDFOR loop is finished.

Program		JOINT 10%
1:	FOR R[1]=6 TO 10	
2:	L P[1] 100mm/sec CNT100	
3:	L P[2] 100mm/sec CNT100	
4:	L P[3] 100mm/sec CNT100	
5:	IF DI[1]=OFF, JMP LBL[1]	
6:	R[1]=11	
7:	LBL[1]	
8:	ENDFOR	
9:	L P[4] 100mm/sec CNT100	
[INST]		[EDCMD] >

R[1] changes as follows:

Loop	1 st
1:	6
2:	6
3:	6
4:	6
5:	6
6:	6->11
7:	11
8:	11
9:	11

(NOTE) "-" means the line is not executed.

Note

Please be careful when you modify Loop counter value within FOR/ENDFOR loop.

Changing Loop counter value within FOR/ENDFOR loop may cause an abnormal behavior or infinite loops.

Please be careful when you modify Target value within FOR/ENDFOR loop when register or argument register is specified for them. Changing the value within FOR/ENDFOR loop may cause an abnormal behavior.

Loop counter value is calculated within FOR/ENDFOR loop:

In the following example, FOR/ENDFOR loop counter is calculated. After execution, Loop counter value will be 11. Loop counter value is different from Target value. However, this result is correct.

Program		JOINT 10%		
1:	FOR R[1]=6 TO 10			
2:	L P[1] 100mm/sec CNT100			
3:	L P[2] 100mm/sec CNT100			
4:	L P[3] 100mm/sec CNT100			
5:	R[1]=R[1]+1			
6:	ENDFOR			
7:	L P[4] 100mm/sec CNT100			
[INST]		[EDCMD] >		

R[1] changes as follows:			
Loop	1 st	2 nd	3 rd
1:	6	-	-
2:	6	8	10
3:	6	8	10
4:	6	8	10
5:	6->7	8->9	10->11
6:	7->8	9->10	11
7:	-	-	11

(NOTE) "-" means the line is not executed.

This is explained as follows.

Line 1, R[1] is 6 (Initial value).

Line 5, R[1] is 7.

Line 6, R[1] is 7 and this satisfies the ENDFOR condition. R[1] is incremented to be 8.

Line 5, R[1] is 9.

Line 6, R[1] is 9 and this satisfies the ENDFOR condition. R[1] is incremented to be 10.

Line 5, R[1] is 11.

Line 6, R[1] is 11 and this does not satisfy the ENDFOR condition. The cursor goes to next line.

This means the user can calculate Loop counter, but need to be careful that the Loop counter value is different from Target value.

Note

In some cases, calculating Loop counter causes an alarm. This problem can be avoided by using DIV just before FOR/ENDFOR statement or input the same value in DATA screen. Especially when you calculate with real value use division, please make sure to use DIV.

R[1]=R[1] DIV 1

9.52.5 Alarms

Alarms occur in the following conditions with FOR/ENDFOR function.

- Execute when the number of FOR statement is smaller than ENDFOR statement
- Execute when the number of ENDFOR statement is smaller than FOR statement
- Execute when there is over 10 nested loops
- Other than integer is used for Initial value or Target value in FOR statement
- Other than integer is used for the value of Loop counter or Target value in ENDFOR statement
- Execute when the number of FOR statement is smaller than ENDFOR statement

Execute the following program. As FOR statement for ENDFOR statement in line 5 does not exist, the alarm “INTP-669 Need FOR for ENDFOR in line 5” occurs in execution.

Program	JOINT 10%
1:	
2: L P[1] 100mm/sec CNT100	
3: L P[2] 100mm/sec CNT100	
4: L P[3] 100mm/sec CNT100	
5: ENDFOR	
6: L P[4] 100mm/sec CNT100	
[INST]	[EDCMD] >

- Execute when the number of ENDFOR statement is smaller than FOR statement

Execute the following program. As ENDFOR statement for FOR statement in line 1 does not exist, the alarm “INTP-670 Need ENDFOR for FOR in line 1” occurs in execution.

Program	JOINT 10%
1: FOR R[1]=1 TO R[2]	
2: L P[1] 100mm/sec CNT100	
3: L P[2] 100mm/sec CNT100	
4: L P[3] 100mm/sec CNT100	
5:	
6: L P[4] 100mm/sec CNT100	
[INST]	[EDCMD] >

- Execute when there is over 10 nested loops

The alarm “INTP-671 Too many FOR” occurs in execution.

- Other than integer is used for Initial value or Target value in FOR/ENDFOR statement
- Other than integer is used for the value of Loop counter or Target value in ENDFOR statement

Execute the following program. As the value of the register for Target value in FOR statement in line 2 is not integer, the alarm “INTP-672 (program, 2) Value type is not integer” occurs when line 2 is executed. In other cases, calculating the Loop counter, Initial value or Target value may cause the same alarm.

Program	JOINT 10%
1: R[2]=1.234	
2: FOR R[1]=1 TO R[2]	
3: L P[2] 100mm/sec CNT100	
4: L P[3] 100mm/sec CNT100	
5: ENDFOR	
6: L P[4] 100mm/sec CNT100	
[INST]	[EDCMD] >

Note

In some cases calculating a register used for Loop counter or Target value causes the alarm above even though the result and the value displayed in DATA screen is integer. This problem can be avoided by using DIV as follows just before FOR/ENDFOR statement or input the same value in DATA screen. Especially when you calculate with real value use division, please make sure to use DIV.

R[1]=R[1] DIV 1

9.53 MATH FUNCTION INSTRUCTION

- Math function instruction enables to calculate math function (ex: SIN COS and so on) in the TP program.
- Math function instruction can be used in assignment statement, conditional statement and wait command.

Note

Math function is a optional function.
You can order math function after 7DA7 14.

9.53.1 Type of Math functions

Usable instructions of math functions are as follows.

Table 9.53.1 Usable instruction of math function

Function	Explanation	Restriction of argument
SQRT[x]	Square root	$0 \leq x$
SIN[x]	Trigonometric function Units are degrees.	None
COS[x]		None
TAN[x]		Except $90, 270 \pm 360n$
ASIN[x]		$-1 \leq x \leq 1$
ACOS[x]		$-1 \leq x \leq 1$
ATAN[x]		None
ATAN2[x,y]		Except $x=0, y=0$
LN[x]	Natural Logarithm	$0 < x$
EXP[x]	Exponent	None
ABS[x]	Absolute	None
TRUNC[x]	Truncate	$-2.1 \times 10^9 \leq x \leq 2.1 \times 10^9$
ROUND[x]	Round off	$-2.1 \times 10^9 \leq x \leq 2.1 \times 10^9$

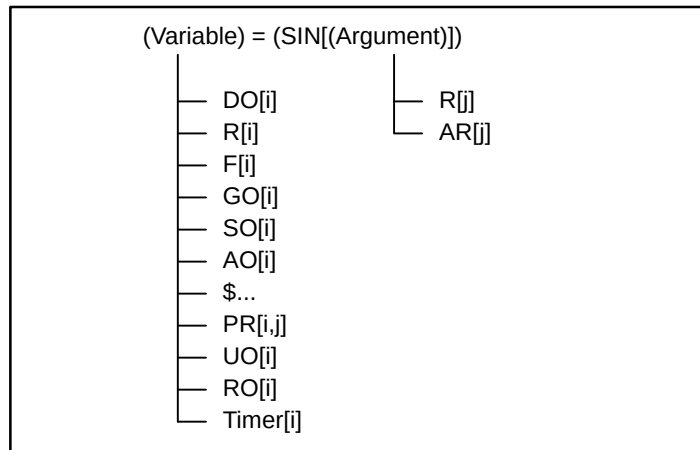
- Units of aforementioned functions are degrees.
- Conversion functions are not supported because the user can achieve them by multiplying a constant value(57.29579 or 0.017453 based on requirement).

9.53.2 Instruction format of math function

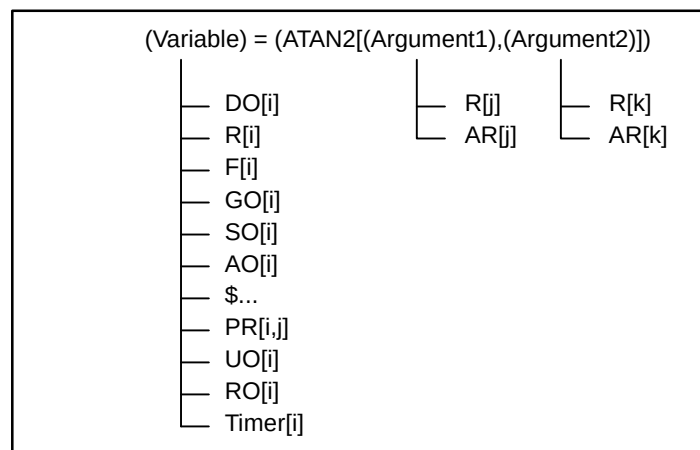
Instruction type of math functions is explained in the following example.

9.53.2.1 Instruction format of assignment statements

- The function which only has one argument (Except of ATAN2[]))

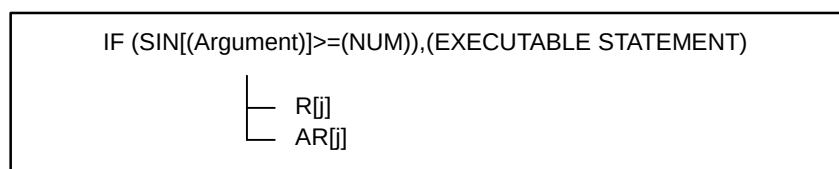


- The function which has two arguments (ATAN2[]))



- The argument of math functions is “Register “ and “Argument Register”.
- Constant is not used directly.

9.53.2.2 Instruction format of relational statements



- A mixed logic instruction expression can be used in the conditional statement of a conditional branch command.
- If the result of conditional statement is ON, the executable statement of the conditional branch command is executed.
- The argument of math functions is “Register “ and “Argument Register”.
- Constant is not used directly.

9.53.2.3 Instruction format of wait command statements

WAIT (SIN[(Argument)]>=(NUM))

└ R[]
└ AR[]

- A mixed logic expression can be specified in the conditional statement of a wait command.
- Wait until the specified condition is satisfied.
- The argument of math functions is “Register “ and “Argument Register”.
- Constant is not used directly.

9.53.3 Function specification of Math functions

9.53.3.1 Square root (SQRT)

Function name: SQRT[X]

Argument: A positive integer or a positive real value to calculate square root.

This function calculates the positive square root of the argument.

Example:

```
1 : R[1] = 4
2 : R[2] = 2
3 : R[3] = (SQRT[R[1]])
4 : R[4] = (SQRT[R[2]])
```

Execute this TP program then,

```
R[3] = 2
R[4] = 1.41421
```

If the argument is minus value, then the following error occurs.

```
INTP-674      Invalid argument
```

9.53.3.2 Trigonometric function (SIN)

Function name: SIN[X]

Argument: An angle to calculate SIN (in degree). An integer or a real value.

This function calculates sine of the argument.

Example:

```
1 : R[1] = 90
2 : R[2] = 45
3 : R[3] = (SIN[R[1]])
4 : R[4] = (SIN[R[2]])
```

Execute this TP program then,

```
R[3] = 1
R[4] = 0.70710
```

9.53.3.3 Trigonometric function (COS)

Function name: COS[X]

Argument: An angle to calculate COS (in degree). An integer or a real value.

This function calculates cosine of the argument.

Example:

```
1 : R[1] = 90
2 : R[2] = 45
3 : R[3] = (COS[R[1]])
4 : R[4] = (COS[R[2]])
```

Execute this TP program then,

```
R[3] = 0
R[4] = 0.70710
```

9.53.3.4 Trigonometric function (TAN)

Function name: TAN[X]

Argument: An angle to calculate TAN (in degree). An integer or a real value.

This function calculates tangent of the arguments.

Example:

```
1 : R[1] = 0
2 : R[2] = 45
3 : R[3] = (TAN[R[1]])
4 : R[4] = (TAN[R[2]])
```

Execute this TP program then,

```
R[3] = 0
R[4] = 1
```

If the argument equals to $180 \cdot n + 90$ and execute, then the following Error will be posted.

```
INTP-674      Invalid argument
```

9.53.3.5 Inverse trigonometric function (ASIN)

Function name: ASIN[x]

Argument: An integer or a real value to calculate ASIN. (Return value is degree.)

This function calculates arc sine of the argument.

Example:

```
1 : R[1] = 1
2 : R[2] = 0.5
3 : R[3] = (ASIN[R[1]])
4 : R[4] = (ASIN[R[2]])
```

Execute this TP program then,

R[3] = 90 R[4] = 30

If the argument ranges are $x > 1$ or $x < -1$ and execute program, then the following Error will be posted.

INTP-674	Invalid argument

9.53.3.6 Inverse trigonometric function (ACOS)

Function name: ACOS[x]

Argument: An integer or a real value to calculate ACOS. (Return value is degree.)

This function calculates arc cosine of the argument.

Example:

1 : R[1] = 1 2 : R[2] = 0.5 3 : R[3] = (ACOS[R[1]]) 4 : R[4] = (ACOS[R[2]])

Execute this TP program then,

R[3] = 0 R[4] = 60

If the argument ranges are $x > 1$ or $x < -1$ and execute program, then the following Error will be posted.

INTP-674	Invalid argument

9.53.3.7 Inverse trigonometric function (ATAN2)

Function name: ATAN2[x,y]

1st argument: x coordinates of the point to calculate ATAN2. An integer or a real value.

2nd argument: y coordinates of the point to calculate ATAN2. An integer or a real value.

This function calculates arc tangent of the 1st and 2nd arguments. (Return value is degree.)

Example:

1 : R[1] = -1 2 : R[2] = 0.5 3 : R[3] = (ATAN2[R[1],R[2]]) 4 : R[4] = (ATAN2[R[2],R[1]])

Execute this TP program then,

R[3] = 153.4349 R[4] = -63.43

9.53.3.8 Inverse trigonometric function (ATAN)

Function name: ATAN [x]

Argument: An integer or a real value to calculate ATAN.

Example:

```

1 : R[1] = 1
2 : R[2] = 0.5
3 : R[3] = (ATAN[R[1]])
4 : R[4] = (ATAN[R[2]])

```

Execute this TP program then,

```

R[3] = 45
R[4] = 26.56

```

9.53.3.9 Exponent

Function name: EXP[x]

Argument: An integer or a real value to calculate EXP.

This function calculates a value equal to e (approximately 2.71828) raised to the power specified by the argument.

Example:

```

1 : R[1] = 0
2 : R[2] = 1
3 : R[3] = (EXP[R[1]])
4 : R[4] = (EXP[R[2]])

```

Execute this TP program then,

```

R[3] = 1
R[4] = 2.71828

```

9.53.3.10 Natural Logarithm

Function name: LN[x]

Argument: An integer or a real value to calculate LN.

This function calculates the natural logarithm of the argument.

Example:

```

1 : R[1] = 1
2 : R[2] = 2.718281828
3 : R[3] = (LN[R[1]])
4 : R[4] = (LN[R[2]])

```

Execute this TP program then,

```

R[3] = 0
R[4] = 1

```

If the argument is $x \leq 0$ and the program execute, then the following Error will be posted.

```

INTP-674      Invalid argument

```

9.53.3.11 Absolute (ABS)

Function name: ABS[x]

Argument: An integer or a real value to calculate ABS.

This function calculates the absolute value of the argument.

Example:

```
1 : R[1] = 5.5
2 : R[2] = -5.2
3 : R[3] = (ABS[R[1]])
4 : R[4] = (ABS[R[2]])
```

Execute this TP program then,

```
R[3] = 5.5
R[4] = 5.2
```

9.53.3.12 Truncate (TRUNC)

Function name: TRUNC[x]

Argument: A real value to calculate TRUNC.

This function converts the real argument to an integer by removing fractional part of the real value.

Example:

```
1 : R[1] = 5.5
2 : R[2] = -5.2
3 : R[3] = (TRUNC[R[1]])
4 : R[4] = (TRUNC[R[2]])
```

Execute this TP program then,

```
R[3] = 5
R[4] = -5
```

9.53.3.13 Round off (ROUND)

Function name: ROUND[x]

Argument: A real value to calculate ROUND.

This function calculates integer value closest to the real argument.

Example:

```
1 : R[1] = 5.5
2 : R[2] = -5.2
3 : R[3] = (ROUND[R[1]])
4 : R[4] = (ROUND[R[2]])
```

Execute this TP program then,

```
R[3] = 6
R[4] = -5
```

9.53.4 Background operation of Math function

Math function instructions are part of the mixed logic instruction.
Then Math function instruction can be used in background operation.

Table 9.53.4 Background operation of math function

Mode	Maximum allowable number of items	Scanning time	Available data	Available operators
Standard mode	No restriction	(Number of items in standard mode / 300-Number of items in high-level mode)*ITP The number of items as used here refers to the total number of items in each mode background operation programs. One ITP is usually eight milliseconds.	SIN, COS, TAN, ASIN, ACOS, ATAN, ATAN2, SQRT, LN, EXP, ABS, TRUNC, ROUND	(,), =, <>, <, <=, >, >=, +, -, *, /, DIV, MOD
High-level mode	270	Eight milliseconds.		

- The scanning time of Math function is 2 or 3 times of normal items, because Math function instruction needs more time to calculate.
- The scanning time of ATAN2 instruction is 3 times of normal items.
- The other math function instructions of scanning time are 2 times of normal items.

9.53.5 Teach Math function instruction

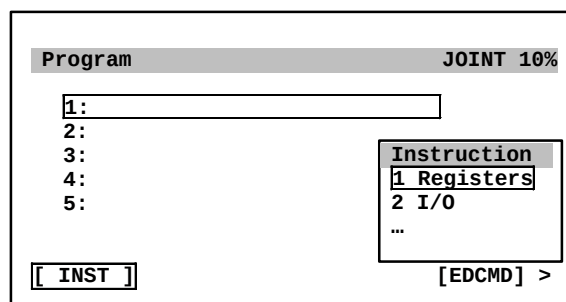
Math function can be used in mixed logic instruction.
As an example, the teaching process of following statement is as follows.

$$1 : R[1] = (\text{SIN}[R[2]])$$

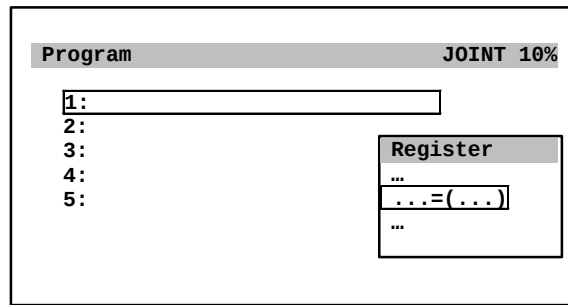
Procedure 9-70 Examples of teaching math function

Step

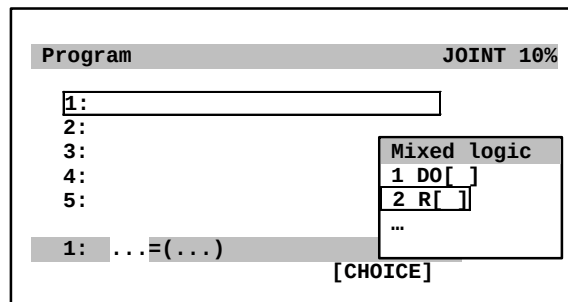
- 1 Press F1 key(INST).
- 2 Select Register.



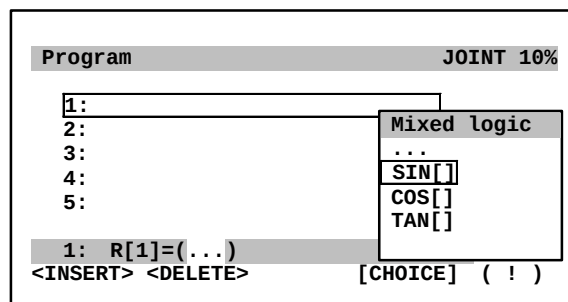
- 3 Select mixed logic instructions "...=(...)",



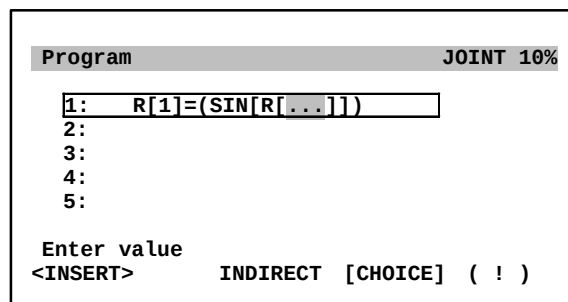
- 4 Select register "R[]". Then the cursor moves in the square bracket.



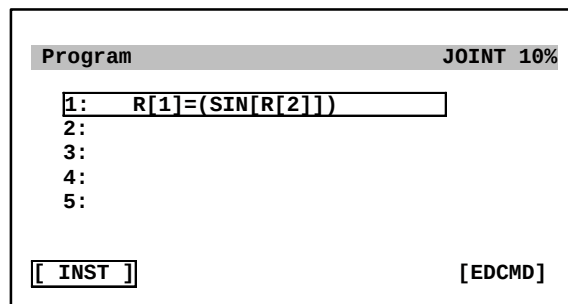
- 5 Input index 1.
6 Then the cursor moves to the right side of equal sign.
7 Select "SIN[]".



- 8 Then the cursor moves in the square bracket. And the default value R[...] is selected.
Arguments are "Register " and "Argument Register". (Not Constant)
The "Register" is default value of argument.
So to use "Argument Register", press F3(CHOICE) then select "Argument Register".
9 For example Register[2] is used here. Input value 2 then the cursor moves to the right parentheses.



- 10 Move cursor to finish teaching.
11 Teaching of SIN[R[2]] function completed.



9.53.6 Restriction of teaching math function

- The argument of math functions is “Register “ and “Argument Register”.
- Constant is not used directly.
- If the user wants to use constant, input constant to a register, and use the register.
- You can put multiple math functions in a single line.
- You cannot put math functions in a math function.

Following statements are supported.

```
1 : R[1] = (SIN[R[2]])
2 : R[2] = (COS[R[3]])
3 : R[3] = (SIN[R[4]] + COS[R[4]])
```

Following statements are not supported.

```
R[1] = (SIN[90])
* Input constant directly.
R[3] = (SQRT[SIN[R[4]] * SIN[R[4]] + COS[R[4]]*COS[R[4]]])
* SIN and COS are in SQRT.
```

- Math function might have a negligible calculation error of 10^{-7} due to internal calculation error because math function follows the same specification of KAREL Built-Ins.
- For example the result of COS[R[1]] (when R[1]=90), may be 10^{-8} that is not exactly zero. This behavior can be avoided by rounding off the result using ROUND function.

9.53.7 Exceptions and Restriction

Math function instruction posts the alarm in the following conditions.

- When the argument exceeds the domain of function definition.

Example :LN[R[1]](R[1] = -1)
 :ASIN[R[2]] (R[2] = 10)
 :SQRT[R[3]] (R[3] = -4)
 :TAN[R[4]] (R[4] = 90)

INTP-674 Invalid argument

- The value overflows when executed.

Example :EXP[R[1]] (R[1] = 100)

INTP-323 Value overflow

- If the variables are divided by zero then, the following error will be occurred.

Example :R[1] / SIN[R[2]] (R[2] = 0)
:10/COS[R[3]] (R[3] = 90)

INTP-208 Divide by 0

9.54 GROUP MASK EXCHANGE

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.

```

Group Mask Exchange          JOINT 10%
Program Selection            1/10
Orig Prog(*, *, *, *, *, *, *)
1
New Prog (*, *, *, *, *, *, *)
2
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]

[ TYPE ] EXEC                >

```

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 9.54 Setting items in group masck exchange screen

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.

Procedure 9-71 Copying a Program Using Group Mask Exchange

Condition

- The Program on which the exchange is to be performed exists.

```

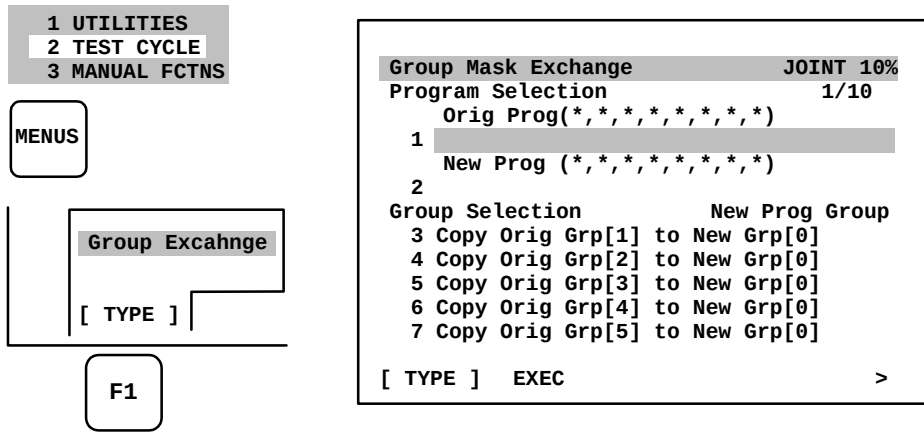
TEST1                      JOINT 10%
                             1/6
1: J P[1] 100% FINE
2: J P[2] 70% CNT50
3: L P[3] 1000cm/min CNT30
4: L P[4] 500mm/sec FINE
5: J P[1] 100% FINE
[End]

POINT                      TOUCHUP>

```

Step

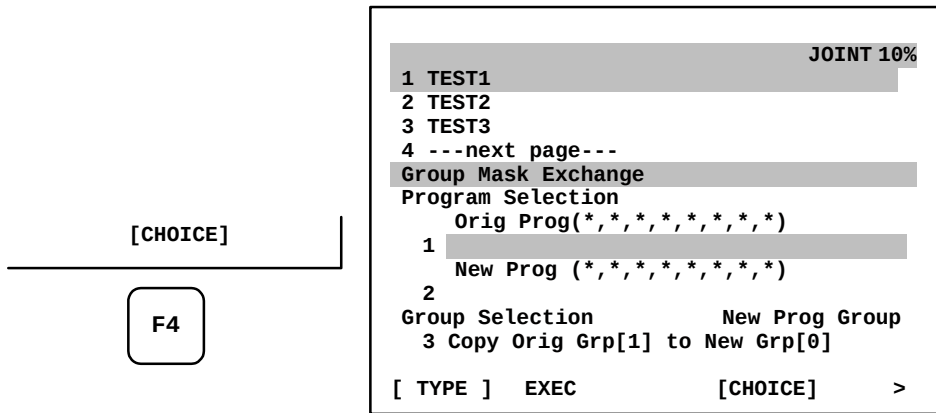
- 1 Press MENUS.
- 2 Select UTILITIES.
- 3 Press F1, [TYPE].
- 4 Select Group Exchg and press ENTER. You will see a screen similar to the following.



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 F4, [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.


```

                                JOINT 10%
1 Words
2 Upper Case
3 Lower Case
4 Options                      --Insert--
Group Mask Exchange
Program Selection
  Orig Prog(1,1,*,*,*,*,*)
1 TEST1
  New Prog (*,*,*,*,*,*,*)
2
Group Selection                New Prog Group
3 Copy Orig Grp[1] to New Grp[0]

PRG      MAIN      SUB      TEST

```

- 7 To select a motion group in the original program to copy to the new program motion groups:
- Move the cursor to the first New group [#] to copy.
 - Type a group number, and press ENTER.

```

Group Mask Exchange                JOINT 10%
Program Selection                  3/10
  Orig Prog(1,1,*,*,*,*,*)
1 TEST1
  New Prog (*,*,*,*,*,*,*)
2 TEST2
Group Selection                New Prog Group
3 Copy Orig Grp[1] to New Grp[2]
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]

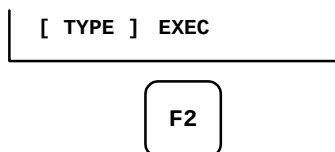
[ TYPE ] EXEC                      >

```

NOTE

If any of the entries for New Group [#] are left at 0, this group will be deleted from the new program.

- 8 When you have finished identifying the groups to be copied, press F2, EXEC.

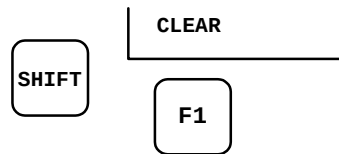
**NOTE**

If the new program exists when you press F2, EXEC, you must confirm the overwrite operation before the execute takes place.

- 9 To clear a single entry, press NEXT, > and then press F1, CLEAR.



- 10 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 PALLETIZING FUNCTION

This chapter explains the palletizing function.

Contents of this chapter

- 10.1 PALLETIZING FUNCTION
- 10.2 PALLETIZING INSTRUCTIONS
- 10.3 TEACHING THE PALLETIZING FUNCTION
- 10.4 EXECUTING THE PALLETIZING FUNCTION
- 10.5 MODIFYING THE PALLETIZING FUNCTION
- 10.6 PALLETIZING FUNCTION WITH EXTENDED AXES
- 10.7 PALLETIZING ALL-POINT TEACHING

10.1 PALLETIZING FUNCTION

Palletizing function

Palletizing is a function for the orderly stacking of workpieces by only teaching several representative points.

- A stacking pattern can be created easily by teaching representative stack points.
- A path pattern can be created by teaching path points (approach points and retraction points).
- Multiple path patterns can be set to perform palletizing in a wide variety of patterns.

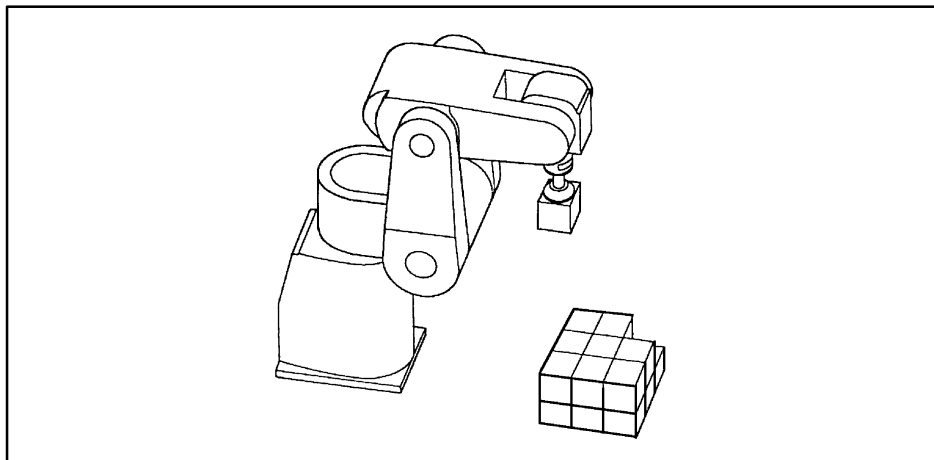


Fig. 10.1 (a) Palletizing

Structure of the palletizing function

The palletizing function consists of the following two patterns:

- Stacking pattern : Determines the method of stacking workpieces.
- Path pattern : Determines the path along which the robot hand moves to stack workpieces.

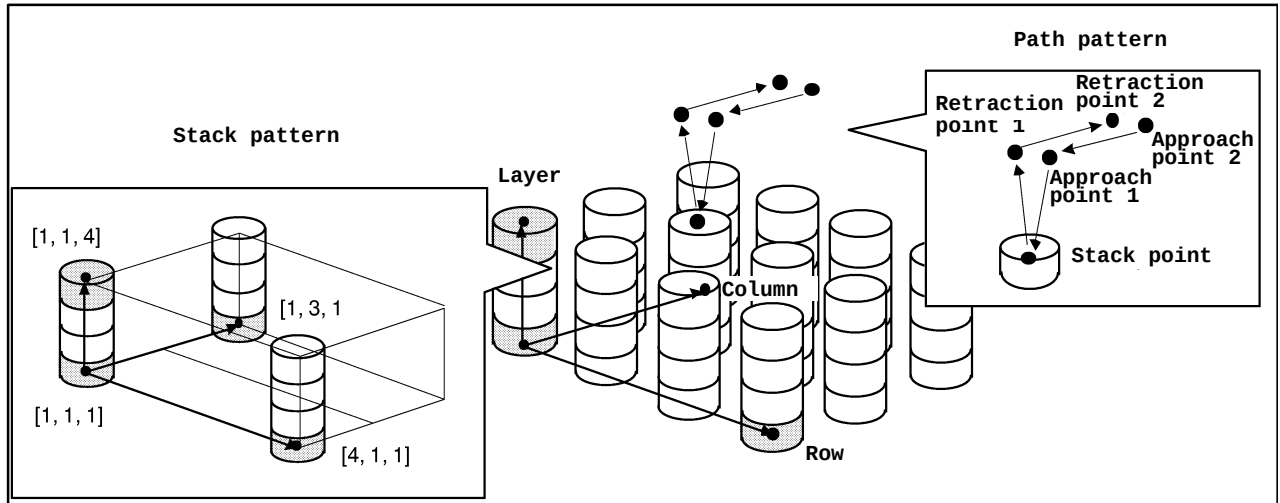


Fig. 10.1 (b) Palletizing Pattern

Types of palletizing

There are the following four types of palletizing according to the methods for setting stack and path patterns (See Section 10.3).

- Palletizing B and palletizing BX
- Palletizing E and palletizing EX

- Palletizing B

Palletizing B can achieve a comparatively easy stacking pattern by one kind of path pattern.

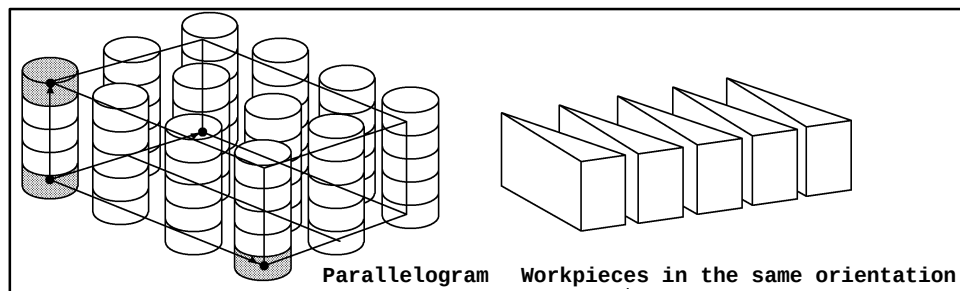


Fig. 10.1 (c) Palletizing-B Function

- Palletizing E

Palletizing E can be used for more complex stack patterns (such as when the postures of workpieces are to be changed or when the shape made by the stacked workpieces, as viewed from below, is not a rectangle).

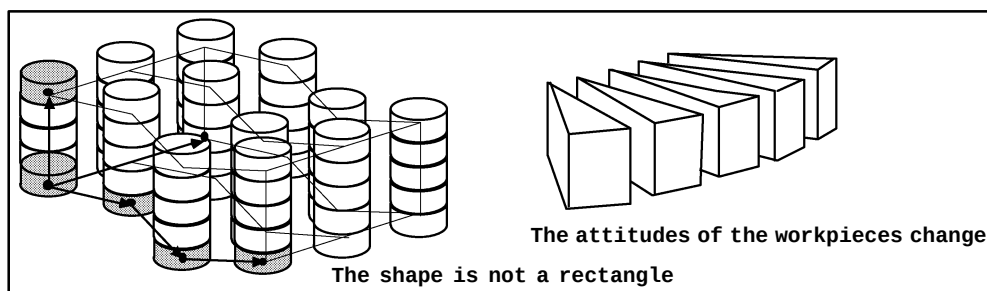


Fig. 10.1 (d) Palletizing E

- Palletizing BX and EX

For palletizing BX and EX, multiple path patterns can be set. For palletizing B and E, only one path pattern can be set.

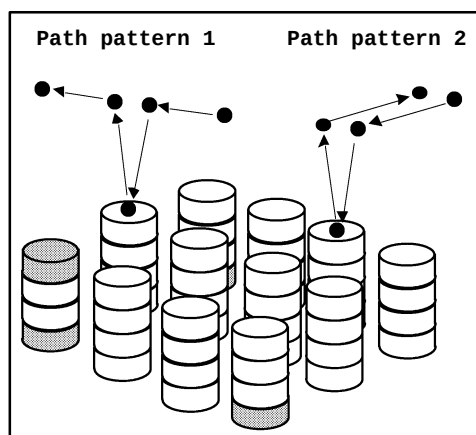


Fig. 10.1 (e) Palletizing BX and EX

10.2 PALLETIZING INSTRUCTIONS

The following palletizing instructions are available:

Table 10.2 Palletizing instructions

Instruction	Function
Palletizing instruction	Calculates the current path based on a stacking pattern, path pattern, and the value held in the palletizing register, and rewrites the position data of a palletizing motion instruction.
Palletizing motion instruction	A motion instruction dedicated to palletizing. It has position data of an approach point, stack point, or retraction point.
Palletizing end instruction	Increments (or decrements) the value of a palletizing register.

Palletizing instruction

Based on the value held in the palletizing register, the palletizing instruction calculates the position of the current stack point from a stack pattern, and also calculates the current path from a path pattern. It then rewrites the position data of a palletizing motion instruction.

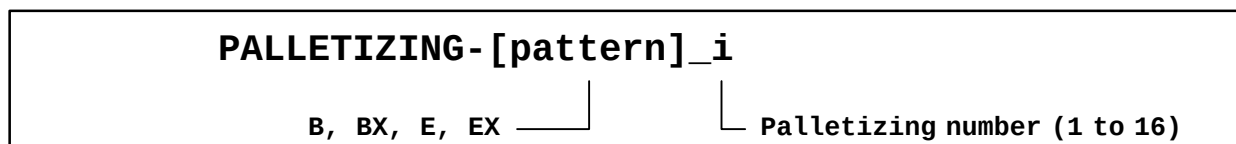


Fig. 10.2 (a) Format of the Palletizing Instruction

Palletizing motion instruction

The palletizing motion instruction is a motion instruction that uses three path points — an approach point, stack point, and retraction point — as position data. This instruction is dedicated to palletizing. Each palletizing instruction rewrites such position data.

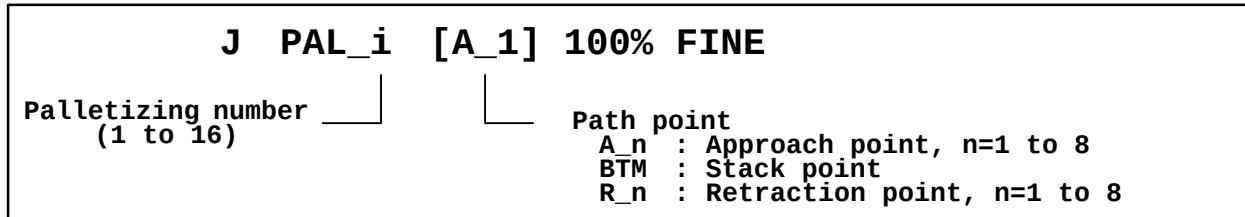


Fig. 10.2 (b) Format of the Palletizing Motion Instruction

Palletizing end instruction

The palletizing end instruction calculates the next stack point and increments (or decrements) the palletizing register value.

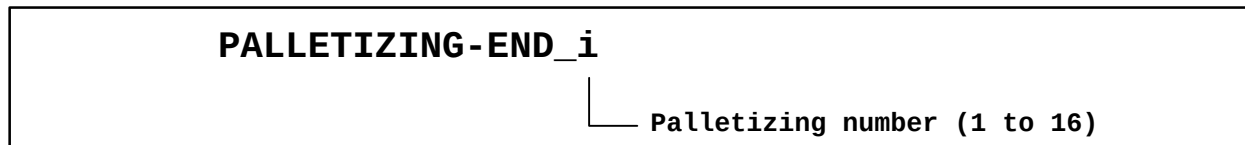


Fig. 10.2 (c) Format of the Palletizing End Instruction

Example

```

1:   PALLETIZING-B_3
2: J PAL_3[ A_2 ] 50% CNT50
3: L PAL_3[ A_1 ] 100mm/sec CNT10
4: L PAL_3[ BTM ] 50mm/sec FINE
5:   hand1 open
6: L PAL_3[ R_1 ] 100mm/sec CNT10
7: J PAL_3[ R_2 ] 50% CNT50
8:   PALLETIZING-END_3
  
```

- Palletizing number

Upon completion of the teaching of palletizing data, palletizing numbers are written automatically together with the instructions (palletizing instruction, palletizing motion instructions, and palletizing end instruction).

When a new palletizing operation is taught, a palletizing number is assigned automatically.

Palletizing register instruction

The palletizing register instruction is used to control palletizing. It performs stack point specification, comparison, and branch. (See Subsection 10.4.1)

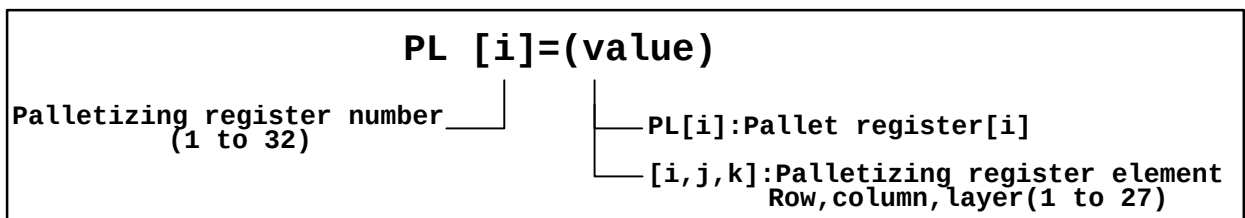


Fig. 10.2 (d) Palletizing Register

10.3 TEACHING THE PALLETIZING FUNCTION

The palletizing function is taught using the following procedure:

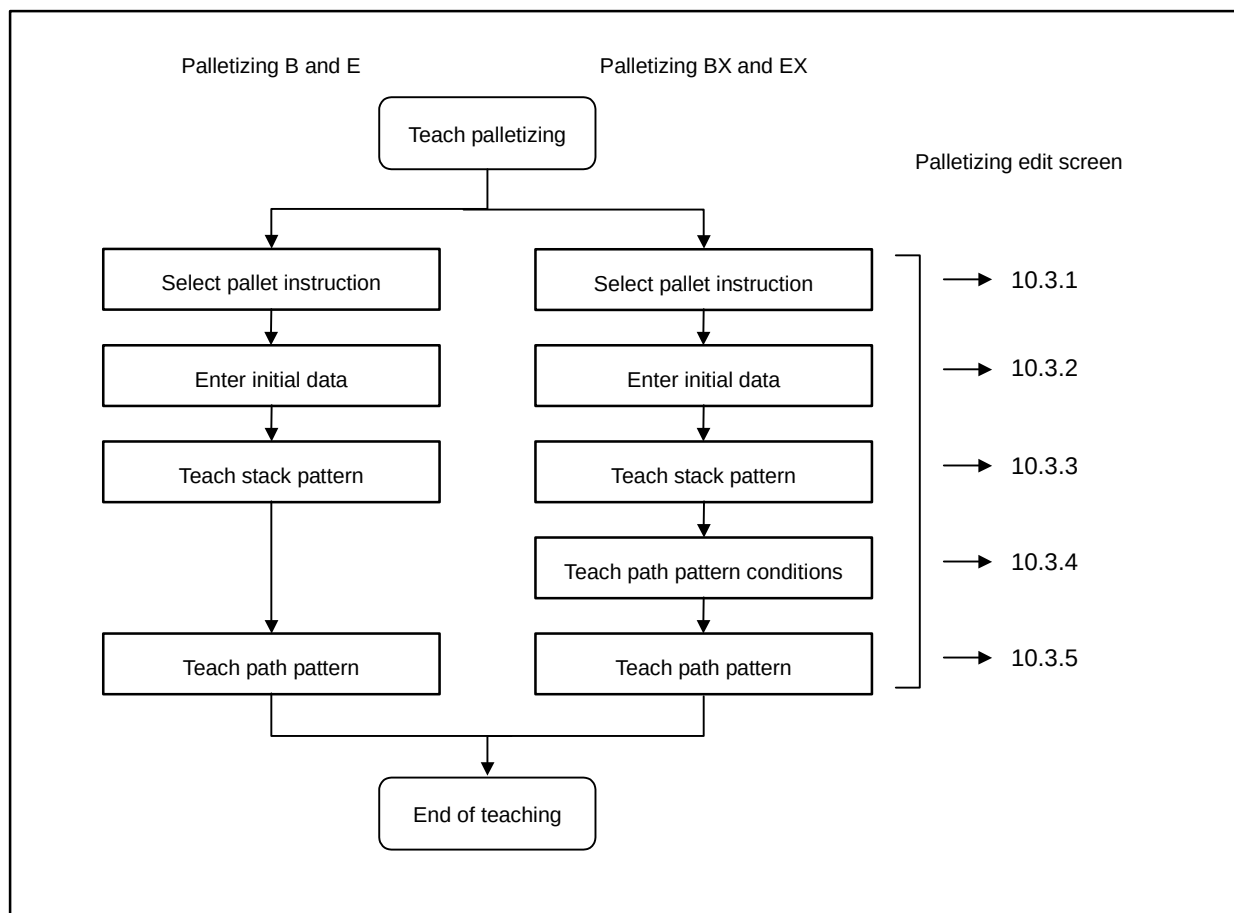


Fig. 10.3 Procedure for Teaching the Palletizing Function

The palletizing function is taught on the palletizing edit screens. One of the palletizing edit screens appears automatically when a palletizing instruction is selected.

When the palletizing function is taught, necessary palletizing instructions such as a palletizing instruction, palletizing motion instruction, and palletizing end instruction are inserted automatically.

The following sections explain the teaching of palletizing EX. For palletizing B, BX, or E, assume that some functions of palletizing EX are restricted.

NOTE

To improve the motion accuracy of palletizing, Tool Center Point should be accurately set. (See Subsection 3.9.1, "Setting a Tool Coordinate System")

10.3.1 Selecting a Palletizing Instruction

To select a pallet instruction, select the type of palletizing to be taught (palletizing B, BX, E, or EX).

Procedure 10-1 Selecting a palletizing instruction

Condition

- Make sure that the teach pendant is enabled.
- Make sure that the palletizing instruction is selected on the program edit screen.

PROGRAM1	JOINT 30%
5: L P[2] 300mm/sec CNT50	6/6
[End]	
[POINT]	TOUCHUP >

Step

- 1 Press NEXT ">" to display the next page. Press F1 "[INST]" to display a submenu.

[INST]	
F1	

Instruction	JOINT 30%
1 Registers	5 JMP/LBL
2 I/O	6 CALL
3 IF/SELECT	7 Palletizing
4 WAIT	8 ---next page---
PROGRAM1	6/6
5: L P[2] 300mm/sec CNT50	

- 2 Select "Palletizing."

JOINT 30%
5 JMP/LBL
6 CALL
7 Palletizing
8 ---next page---
ENTER

PALLETIZING system
1 PALLETIZING-B 5 PALLETIZING-END
2 PALLETIZING-BX 6
3 PALLETIZING-E 7
4 PALLETIZING-EX 8
PROGRAM1
5: L P[2] 300mm/sec CNT50
6/6

- 3 Select "PALLETIZING-EX." The screen then changes to the initial data input screen automatically, one of the palletizing edit screen.

PALLETIZING system
1 PALLETIZING-B 5
2 PALLETIZING-BX 6
3 PALLETIZING-E 7
4 PALLETIZING-EX 8
PROGRAM1
ENTER

PROGRAM1	JOINT 30%
PALLETIZING Configuration	
PALETIZING_1 []	
TYPE = [PALLET] INCR = [1]	
PAL REG = [1] ORDER = [RCL]	
ROWS = [1 LINE FIX]	
COLUMNS = [1 LINE FIX]	
LAYERS = [1 LINE FIX 1]	
AUXILIARY ROS = [NO]	
APPR = [1] RTRT = [1] PATTERN = [1]	
Press ENTER	
PROG	DONE

10.3.2 Inputting Initial Data

On the initial data input screen, specify how palletizing is performed. The data set on the initial data input screen is used for subsequent teach screens. The initial data screen contains the following items:

For palletizing B

PROGRAM1		JOINT	30%
PALLETIZING Configuration			
PALETIZING_1 []			
TYPE = [PALLET] INCR = [1]			
PAL REG = [1] ORDER = [RCL]			
ROWS = [5]			
COLUMNS = [4]			
LAYERS = [3]			
AUXILIARY ROS = [NO]			
APPR = [2] RTRT = [2]			
Press ENTER			
PROG		DONE	

For palletizing BX

PROGRAM1		JOINT	30%
PALLETIZING Configuration			
PALETIZING_1 []			
TYPE = [PALLET] INCR = [1]			
PAL REG = [1] ORDER = [RCL]			
ROWS = [5]			
COLUMNS = [4]			
LAYERS = [3]			
AUXILIARY ROS = [NO]			
APPR = [2] RTRT = [2] PATTERN = [2]			
Press ENTER			
PROG		DONE	

For palletizing E

PROGRAM1		JOINT	30%
PALLETIZING Configuration			
PALETIZING_1 []			
TYPE = [PALLET] INCR = [1]			
PAL REG = [1] ORDER = [RCL]			
ROWS = [5 LINE FIX]			
COLUMNS = [4 LINE FIX]			
LAYERS = [3 LINE FIX 1]			
AUXILIARY ROS = [NO]			
APPR = [2] RTRT = [2]			
Press ENTER			
PROG		DONE	

For palletizing EX

PROGRAM1		JOINT	30%
PALLETIZING Configuration			
PALETIZING_1 []			
TYPE = [PALLET] INCR = [1]			
PAL REG = [1] ORDER = [RCL]			
ROWS = [5 LINE FIX]			
COLUMNS = [4 LINE FIX]			
LAYERS = [3 LINE FIX 1]			
AUXILIARY ROS = [NO]			
APPR = [2] RTRT = [2] PATTERN = [2]			
Press ENTER			
PROG		DONE	

Table 10.3.2 (a) Types of Palletizing

	Arrangement mode	Layer pattern	Posture control	Path pattern count
B	2-point teaching only	Not set	Always fixed	1
BX	2-point teaching only	Not set	Always fixed	1 to 16
E	2-point teaching, all-point teaching, or interval specification	Set	Fixed or split	1
EX	2-point teaching, all-point teaching, or interval specification	Set	Fixed or split	1 to 16

When a pallet instruction is selected, the initial data input screen corresponding to the selected type of palletizing appears. For palletizing EX, all palletizing functions can be specified. For palletizing B, BX, and E, restrictions are imposed on the specification of the functions.

This section explains how to enter initial data for palletizing EX. For palletizing B, BX, or E, assume that some functions of palletizing EX are restricted.

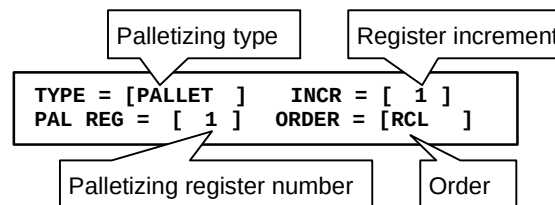
Table 10.3.2 (b) Initial Palletizing Data

Palletizing number	A number is assigned automatically when a palletizing statement is taught. PALLETIZING_N: 1 to 16
Palletizing type	Specify whether the palletizing register is to be incremented or decremented by the palletizing end instruction. (See Subsection 10.4.1.) Select stacking (PALLET) or unstacking (DEPALLET).
Register increment	Specify the value by which the value held in the palletizing register is to be incremented or decremented by the palletizing end instruction. (See Subsection 10.4.1.)
Palletizing register	Specify the palletizing register to be used by the palletizing instruction and palletizing end instruction.
Order	Specify the stacking (unstacking) order of row, column, and layer. R: Row, C: Column, L: Layer
Numbers of rows, columns, and layers	Numbers of rows, columns, and layers for a stacking pattern. (See Subsection 10.3.3.) 1 to 127

Arrangement mode	How rows, columns, and layers are arranged for a stack pattern. The 2-point or all-point teaching, or interval specification can be specified (only for palletizing E or EX).
Posture control	Control the posture at rows, columns, and layers for a stacking pattern. (only for palletizing E or EX).
Layer pattern count	How workpieces are stacked can be specified for each layer (only for palletizing E or EX). 1 to 16
Number of approach points	Number of approach points in a path pattern. (See Subsection 10.3.5.) 0 to 8
Number of retraction points	Number of retraction points in a path pattern. (See Subsection 10.3.5.) 0 to 8
Path pattern count	Number of path patterns (Subsection 10.3.4) (only for palletizing BX or EX). 1 to 16

Initial data related to the stacking method

In the palletizing function, the stack point is controlled using a palletizing register. (See Subsection 10.4.1, “Palletizing Register.”) How the palletizing register is controlled can be specified as initial data. According to this data, the way of stacking is determined.



- For the pallet type (TYPE), specify either PALLET or DEPALLET (standard setting: PALLET). (See Subsection 10.4.1, “Palletizing Register.”)
- For the register increment (INCR), specify by which amount the stack (unstack) position advances or retracts. That is, specify a value by which the palletizing register is incremented or decremented by the palletizing end instruction. The standard setting is 1. (See Subsection 10.4.1, “Palletizing Register.”)
- As the palletizing register, specify the register number of a palletizing register used for stack control.



CAUTION

Make sure that the specified palletizing register number is not used by another palletizing function.

- For the order (ORDER), specify the stacking/unstacking order of row, column, and layer.

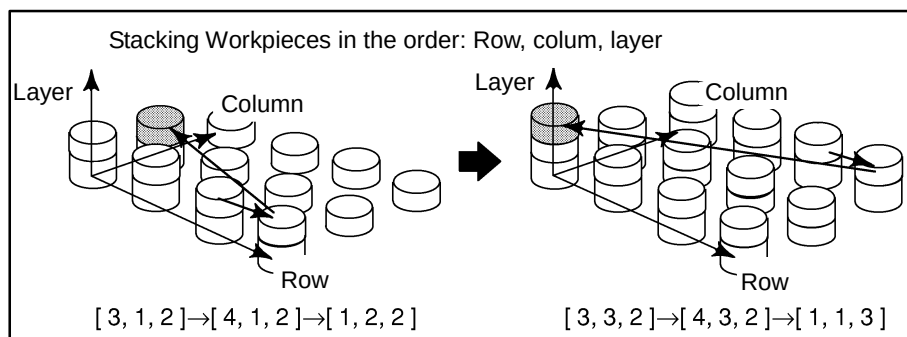
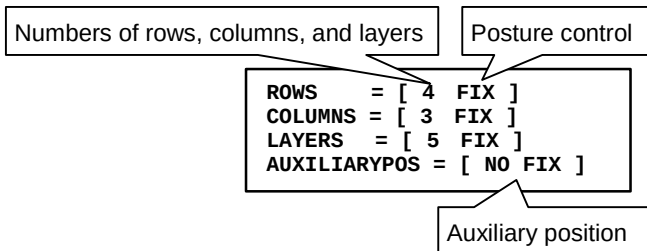


Fig. 10.3.2 Palletizing Order

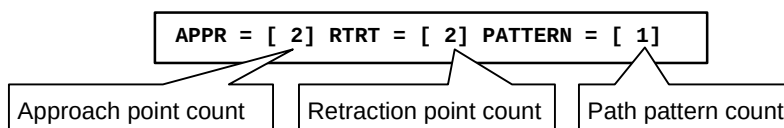
Initial data related to a stacking pattern

As the stacking pattern data, specify the numbers of rows, columns, and layers, posture control type and also specify whether to provide an auxiliary position. (See Subsection 10.3.3, “Teaching a Stacking Pattern.”)



Initial data related to a path pattern

As the initial path pattern data, specify the number of approach points and the number of retraction points. (See Subsection 10.3.5, “Teaching a Path Pattern.”)



Procedure 10-2 Inputting initial palletizing data

Step

- 1 As the palletizing instruction, select palletizing-EX. The initial data input screen then appears. (See Subsection 10.3.1, “Selecting a Palletizing Instruction.”)

PALLETIZING system 1 PALLETIZING-B 5 2 PALLETIZING-BX 6 3 PALLETIZING-E 7 4 PALLETIZING-EX 8 PROGRAM1	PROGRAM1 JOINT 30% PALLETIZING Configuration PALETIZING_1 [] TYPE = [PALLET] INCR = [1] PAL REG = [1] ORDER = [RCL] ROWS = [1 LINE FIX] COLUMNS = [1 LINE FIX] LAYERS = [1 LINE FIX 1] AUXILIARY ROS = [NO] APPR = [1] RTRT = [1] PATTERN = [1] Press ENTER PROG DONE

ENTER

NOTE

PALLETIZING_1 displayed on the initial data input screen indicates the first palletizing instruction in the program.

- 2 To enter a comment, follow the procedure below.
 - a Place the cursor on the comment line, then press the ENTER key. The character input submenu appears.

JOINT 30% []] INCR = [1] ENTER	JOINT 30% 1 Upper Case 2 Lower Case 3 Punctuation 4 Options ---Insert--- PROGRAM1

- b Select the type of character input to be used — upper case, lower case, punctuation or options — with the ↑ and ↓ keys.
 - c Press an appropriate function key, then enter characters.

- d After a comment has been completely entered, press the ENTER key.

PROGRAM1		JOINT	30%
PALLETIZING Configuration			
PALETIZING_1	[WORK4	]	
PROG	DONE		

- 3 To select a palletizing type, move the cursor to the TYPE field, then select a function key.

PROG	PALLET	DEPALL
F2		

PROGRAM1		JOINT	30%
PALLETIZING Configuration			
TYPE =	[PALLET	]	INCR = [1]
PROG	PALLET	DEPALL	DONE

- 4 To enter a numeric value, press a numeric key, then press the ENTER key.

1	ENTER

PROGRAM1		JOINT	30%
PALLETIZING Configuration			
TYPE =	[PALLET	]	INCR = [1]
PROG	DONE		

- 5 Specify a palletizing order by selecting the function keys in the target order.

PROG	R	C
F2		

PROGRAM1		JOINT	30%
PALLETIZING Configuration			
PAL REG =	[1]	ORDER =	[R]
PROG	R	C	L
DONE			

After the second item has been selected, the third item is determined automatically.

PROG	R	C
F3		

PROGRAM1		JOINT	30%
PALLETIZING Configuration			
PAL REG =	[1]	ORDER =	[RCL]
PROG	R	C	L
DONE			

- 6 To specify the number of rows, columns, and layers, input numeric value by the numeric keys and press enter key. To specify the arrangement mode, move the cursor to the setting field and select the function key.

PROG	LINE	FREE	
F2			

PROGRAM1		JOINT	30%
PALLETIZING Configuration			
LAYERS	= [1	LINE	FIX 1]
AUXILIARY POS	= [NO		]
PROG	LINE	FREE	DONE

- 7 To specify the arrangement mode by the interval specification, move the cursor to the setting field and enter numeric value. (unit : mm)

PROG	LINE	FREE	
2	0	0	ENTER

PROGRAM1		JOINT	30%
PALLETIZING Configuration			
LAYERS	= [1	200	FIX 1]
AUXILIARY POS	= [NO		]
PROG	LINE	FREE	DONE

- 8 To specify whether there are an auxiliary points or not, move the cursor to the setting field and select the desired function key menu.

PROG	YES	NO	
F3			

PROGRAM1		JOINT	30%
PALLETIZING Configuration			
LAYERS	= [1	200	FIX 1]
AUXILIARY POS	= [NO		]
PROG	YES	NO	DONE

NOTE

When specifying the setting of auxiliary points, also select either of FIX/INTER.

- 9 Enter the approach point count and retraction point count.
 10 To stop the initial data setting, press F1 "PROG."

PROG	FIX	INTER
F1		

**CAUTION**

When the initial data setting is stopped before it is completed, the values set up to that time are invalidated.

```

PROGRAM1                                JOINT 30%
PALLETIZING Configuration

PALETIZING_1 [WORK PALLET]
TYPE = [PALLET] INCR = [ 1 ]
PAL REG = [ 1 ] ORDER = [RCL ]
ROWS    = [4 LINE FIX]
COLUMNS = [3 LINE FIX]
LAYERS   = [5 LINE FIX 1]
AUXILIARY ROS = [ NO ]
APPR = [ 2] RTRT = [ 2] PATTERN = [ 1]
Press ENTER

PROG                                     DONE

```

- 11 When all data items have been entered, press F5 “DONE.” The initial data input screen then disappears and the palletizing stacking pattern teach screen appears.

DONE

F5

```

PROGRAM1                                JOINT 30%
PALLETIZING Bottom Points

1: *P [ 1, 1, 1 ]
2: *P [ 4, 1, 1 ]
3: *P [ 1, 3, 1 ]
4: *P [ 1, 1, 5 ]

PROG                                     RECORD  DONE  >

```

When the palletizing stacking pattern teach screen is displayed after the setting or changing of initial palletizing data is completed with F5 “DONE,” the palletizing register is initialized automatically. (See Subsection 10.4.1, “Palletizing Register.”)

10.3.3 Teaching a Stacking Pattern

On the palletizing stacking pattern teach screen, teach representative stack points of a stacking pattern. From these representative points, a target stack point is calculated automatically at the time of palletizing.

Stacking Pattern Teach Screen

```

PROGRAM1                                JOINT 30%
PALLETIZING Bottom Points

1: P [ 1, 1, 1 ]
2: P [ 4, 1, 1 ]
3: P [ 1, 3, 1 ]
4: P [ 1, 1, 5 ]

BACK                                     RECORD  DONE  >

```

With or without an auxiliary position

A list of the positions to be taught is displayed based on the initial palletizing data. Following this list, teach the positions of the representative stack points.

For the stacking pattern without an auxiliary position, individually teach four tops of the stacking pattern.

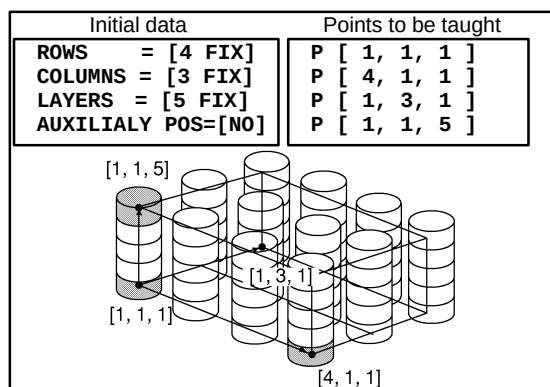


Fig. 10.3.3 (a) Stacking Pattern with No Auxiliary Position

For a stacking pattern with an auxiliary position, when the shape of the first layer is a trapezoid, also teach the fifth position using the function provided.

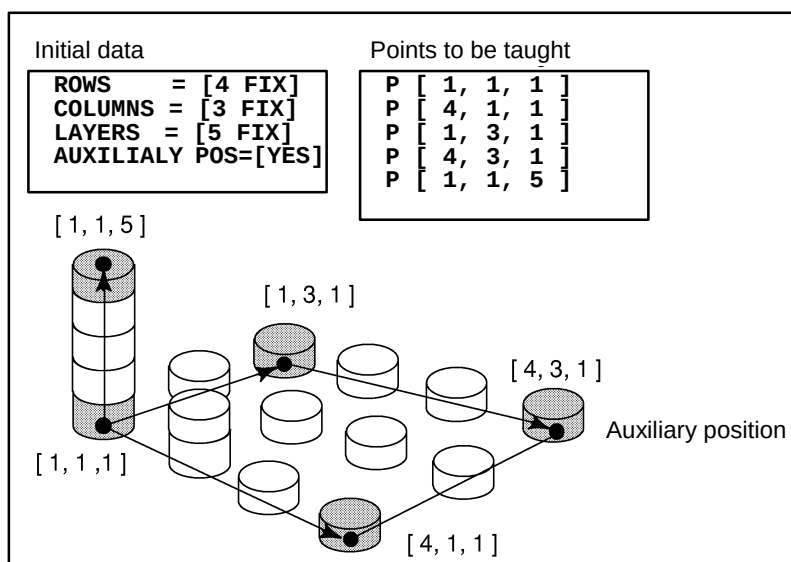


Fig. 10.3.3 (b) Stacking Pattern with an Auxiliary Position

Types of arrangement modes/2-point teaching

When 2-point teaching is selected, teach the representative two points at both ends to set all the points in the row, column, and layer directions (standard).

NOTE

The following explanation is not relevant to palletizing B and BX. See Subsection 10.3.4.

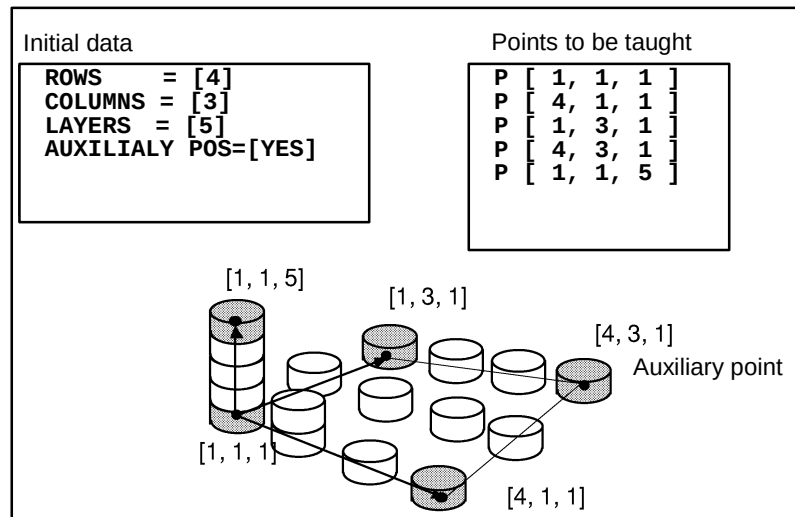


Fig. 10.3.3 (c) Teaching Method by 2-Point Teaching

All-point teaching

When selecting all-point teaching, directly teach all the points in the row, column, and layer directions.

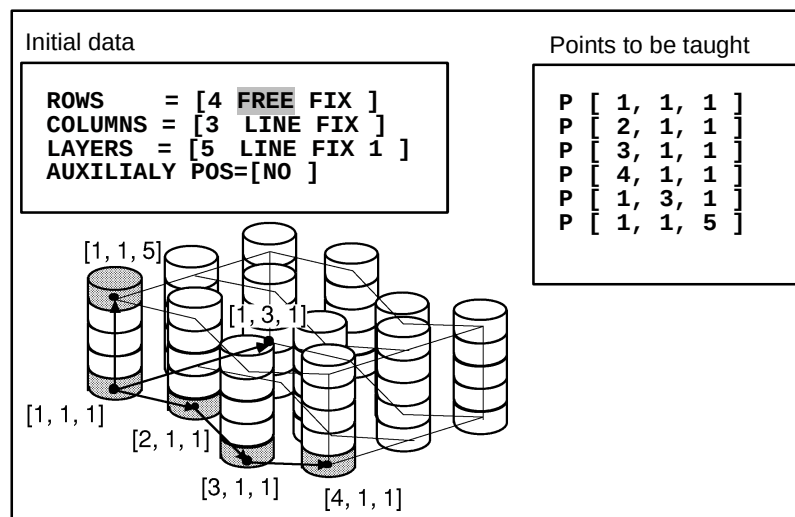


Fig. 10.3.3 (d) Teaching Method by All-Point Teaching

Interval specification

When selecting the interval specification, specify the two points at both ends in each of the row, column, and layer directions, as well as the distance between workpieces, to set all points.

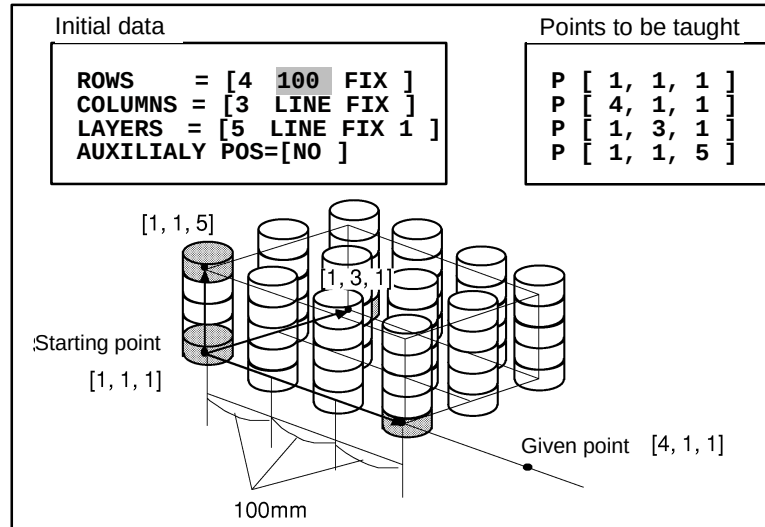


Fig. 10.3.3 (e) Teaching Method by Interval Specification

Types of posture control

When the fixed posture is specified, workpieces at all the stack points always take the posture taught at point [1,1,1] (standard).

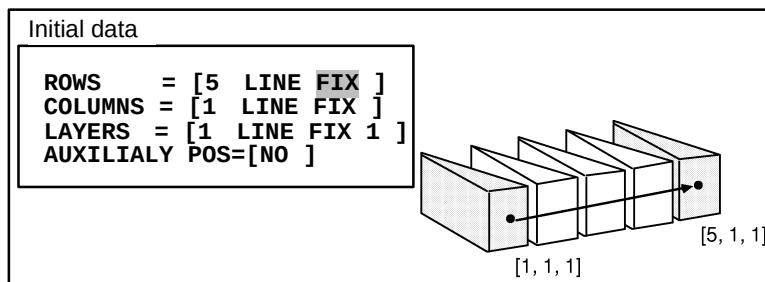


Fig. 10.3.3 (f) Postures of Workpieces at Stack Points when the Fixed Posture Is Specified

For the split posture, when 2-point teaching is specified, workpieces take the postures obtained by splitting the postures taught at the two end points. When all-point teaching is specified, workpieces take the postures at the taught points.

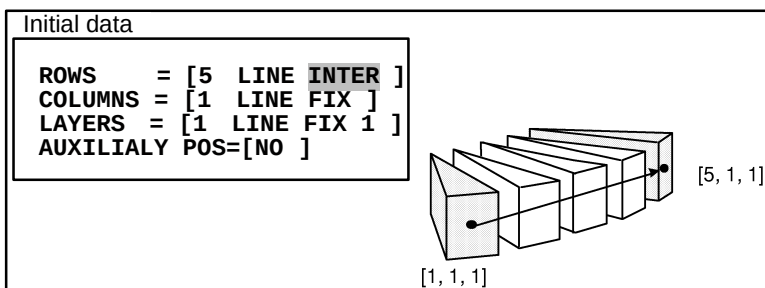


Fig. 10.3.3 (g) Postures of Workpieces at Stack Points when the Split Posture Is Specified

Layer pattern count

To change the stack pattern every few layers, enter the number of layer patterns.

The layer pattern count is valid only when 2-point teaching is specified for the layer arrangement (for other cases, the layer pattern count is always 1).

For the first layer, the workpieces are always stacked at the stack points in layer pattern 1.

When the layer pattern count is N, the numbers of layers and layer patterns are the same until layer N. For layer (N+1) and beyond, layer patterns starting from layer pattern 1 are repeated.

Teach the position for the layer direction only in the teaching for layer pattern 1.

The position for the layer direction in each layer are calculated by the taught position for layer pattern1.

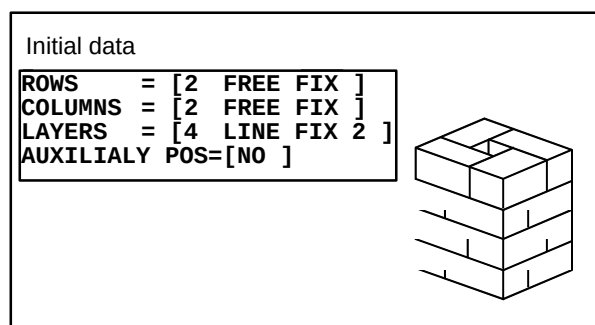


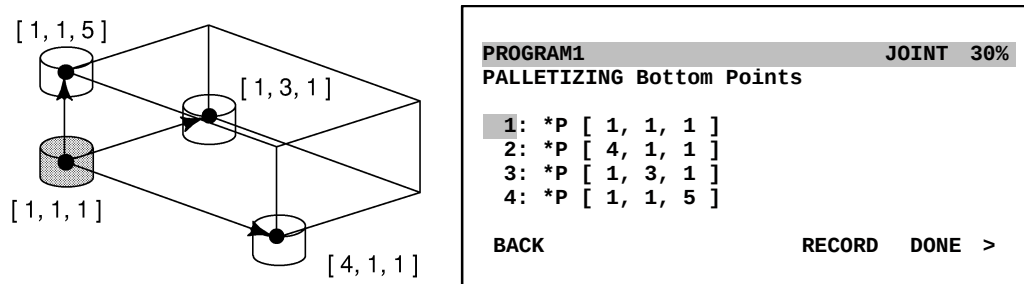
Fig. 10.3.3 (h) Posture at the stacking point in division posture

When the total number of layers is less than 16, a number not greater than the total number can be set for the layer pattern count. If a number less than the layer pattern count is subsequently specified for the number of layers, the layer pattern count is automatically changed to the number of layers.

Procedure 10-3 Teaching a palletizing stacking pattern

Step

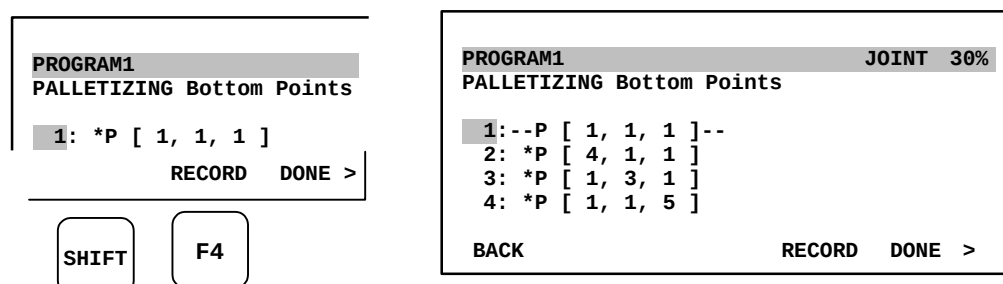
- 1 Based on the initial data setting, a list of stack points to be taught is displayed.



NOTE

The number of representative stack points to be recorded depends on the numbers of rows, columns, and of layers set on the initial data input screen. In the above sample screen, four rows, three columns, and five layers are set. For each point, row, column, and then layer numbers are specified.

- 2 Move the robot by jog feed to a position which is to be taught as a representative stack point.
- 3 Place the cursor on the appropriate line, and press F4 "RECORD" while holding down the SHIFT key. The current robot position is then recorded.



The positions not yet taught are marked with an asterisk. The positions already taught are marked with "--".

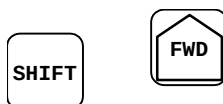
- 4 To display detailed position data for a taught representative stack point, move the cursor to the target stack point number, then press F5 "POSITION." The detailed position data is then indicated.

PROGRAM1 PALLETIZING Bottom Points 1:--P [1, 1, 1]-- POSITION>		Position Detail JOINT 30% PAL_1[BTM] UT:0 UT:1 CONF: NT 0 X 516.129 mm W 180.000 deg Y 111.347 mm P 0.000 deg Z 1010.224 mm R 0.000 deg PROGRAM1 PALLETIZING Bottom Points 1:--P [1, 1, 1]-- 2: *P [4, 1, 1] 3: *P [1, 3, 1] 4: *P [1, 1, 5] CONFIG DONE	

Numeric values can also be entered directly to specify position data. To return to the previous stacking pattern teach screen, press F4 "DONE."

PROGRAM1 JOINT 30% PALLETIZING Bottom Points 1:--P [1, 1, 1]-- 2:--P [4, 1, 1]-- 3:--P [1, 3, 1]-- 4:--P [1, 1, 5]-- BACK RECORD DONE >	

- 5 Pressing the FWD key while holding down the SHIFT key causes the robot to move to the representative stack point indicated by the cursor. This operation can be performed to confirm the taught point.



- 6 To return to the previous initial data teach screen, press F1 "BACK."

BACK
F1

- 7 Press F5 DONE to display the path pattern condition setting screen (BX or EX) or path pattern teaching screen (B or E) (Subsection 10.3.4 or 10.3.5).

NOTE

When layer patterns are used (E or EX) and F5 DONE is pressed, the screen for specifying the stack pattern for the next layer appears.

RECORD DONE > F5	PROGRAM1 JOINT 30% PALLETIZING Bottom Points 1:--P [1, 1, 2]-- 2/2 2:--P [4, 1, 2]-- 3:--P [1, 3, 2]-- BACK RECORD DONE >

10.3.4 Setting Path Pattern Conditions

The screen explained in this section is not displayed for palletizing B or E. See Subsection 10.3.5.

The palletizing path pattern condition setting screen is used to set conditions indicating which path pattern is to be used for each stack point in advance when multiple path patterns are to be set on the path pattern teaching screen (See Subsection 10.3.5).

For palletizing BX and EX, multiple path patterns can be set independently for stack points. For palletizing B and E, this screen is not displayed because only one path pattern can be set.

Path pattern condition setting screen

PROGRAM1		JOINT 30%	
PALLETIZING Route Patterns			
PTN[1]	=	[* , 3-1, *]	
PTN[2]	=	[* , 3-2, *]	
PTN[3]	=	[* , 3-0, *]	
BACK		DIRECT	MODULO DONE >

To specify a path for each stack point, as many path patterns as required must be specified when initial data is set. For each path pattern, set a path pattern condition.

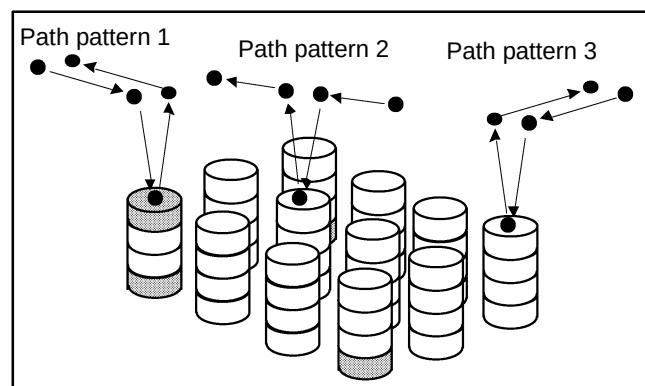


Fig. 10.3.4 (a) Palletizing Using Three Path Patterns

How to use path pattern conditions

- The execution of palletizing uses the path pattern with the condition number for which the row, column, and layer numbers at the stack point match the row, column, and layer (element) values of a path pattern condition.
- In direct specification mode, specify numbers between 1 to 127 for a stack point. An asterisk indicates an arbitrary stack point.
- In the remainder specification mode, specify path pattern condition element "m-n" using a remainder system for a stack point.
Layer element "3-1" indicates a layer corresponding to a stack point value for which a remainder of 1 is obtained by dividing the value by 3.
- If the current stack point corresponds to no path pattern condition, an alarm occurs. If the current stack point corresponds to two or more path pattern conditions, a path pattern condition is used according to the following conditions:
 - a. A path pattern condition specified in direct specification mode is used.
 - b. When two or more path pattern conditions are specified in direct specification mode, a path pattern condition specified in the remainder specification mode is used. When two or more path pattern conditions are specified in remainder specification mode, a path pattern condition in which the greatest value is specified for m is used.

- c. When two or more path pattern conditions satisfy conditions a and b above, the path pattern condition having the smallest path pattern condition number is used.

The following shows the priority among the sample path pattern conditions:

Example PTN	[1] = [*	, 1	, 2	]
PTN	[2] = [*	, *	, 2	]
PTN	[3] = [*	, 3-2	, 4-1	]
PTN	[4] = [*	, *	, 4-1	]
PTN	[5] = [*	, *	, 2-1	]
PTN	[6] = [*	, *	, *	]

For the example shown on the previous page, pattern 1 is used for the stack points in column 1, pattern 2 is used for the stack points on column 2, and pattern 3 is used for the stack points in column 3.

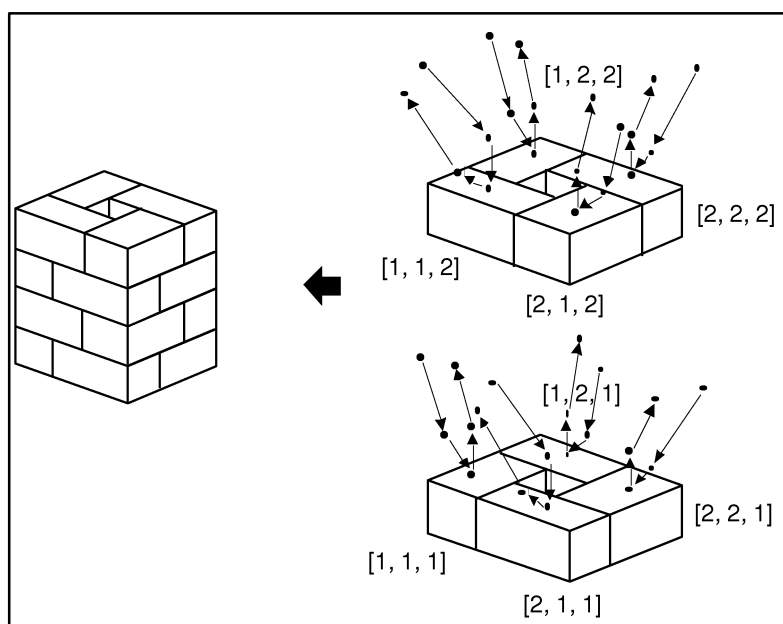


Fig. 10.3.4 (b) Box Palletizing Using Eight Path Patterns

Example PTN	[1] = [1	, 1	, 2-1	]
PTN	[2] = [2	, 1	, 2-1	]
PTN	[3] = [1	, 2	, 2-1	]
PTN	[4] = [2	, 2	, 2-1	]
PTN	[5] = [1	, 1	, 2-0	]
PTN	[6] = [2	, 1	, 2-0	]
PTN	[7] = [1	, 2	, 2-0	]
PTN	[8] = [2	, 2	, 2-0	]

In the this example, eight path patterns are defined and repeated for every two layers because different paths must be set according to the box position.

Procedure 10-4 Setting palletizing path pattern conditions

Step

- 1 Condition items to be entered are displayed according to the value set for the pattern count as initial data.

PROGRAM1		JOINT 30%	
PALLETIZING Route Patterns			
PTN[1] = [	*	,	*
PTN[2] = [	*	,	*
PTN[3] = [	*	,	*
Enter value			
BACK	DIRECT	MODULO	DONE >

- 2 In direct specification mode, position the cursor to the point to be changed and enter a numeric value. To specify an asterisk (*), enter zero.

JOINT 30%	
Patterns	
[* , *	1
[* , *	
[* , *	
INPUT	

PROGRAM1		JOINT 30%	
PALLETIZING Route Patterns			
PTN[1] = [	*	,	1
	*		
BACK	DIRECT	MODULO	DONE >

- 3 In remainder specification mode, press F4 MODULO. The target item is divided into two sub-items. Enter a value for each sub-item.

DIRECT	MODULO	DONE >
F4		

PROGRAM1		JOINT 30%	
PALLETIZING Route Patterns			
PTN[1] = [	*	,	1-0
	*		
BACK	DIRECT	MODULO	DONE >

- 4 To specify values in direct specification mode, press F3 DIRECT.

DIRECT	MODULO	DONE >
F3		

- 5 Press F1 BACK to display the previous stack point teaching screen again.

BACK	DIRECT
F1	

- 6 Press F5 DONE to display the next path pattern teaching screen.

DIRECT	MODULO	DONE >
F5		

PROGRAM1		JOINT 30%	
PALLETIZING Route Patterns			
IF PL[1] = [	*	,	*
	*		
1:Joint *P[A_2]	30%	FINE	1/1
2:Joint *P[A_1]	30%	FINE	
3:Joint *P[BTM]	30%	FINE	
4:Joint *P[R_1]	30%	FINE	
5:Joint *P[R_2]	30%	FINE	
Teach Route Point			
BACK	POINT	RECORD	DONE >

10.3.5 Teaching a Path Pattern

On the palletizing path pattern teach screen, set several path points which are passed before and after a workpiece is stacked (or unstacked) at a stack point. The path points change depending on the position of the stack point.

Path Pattern Teach Screen

PROGRAM1		JOINT	30%
PALLETIZING Route Points			
IF PL[1] = [*, *, *]		1/3	
1:Joint	P[A_2]	30%	FINE
2:Joint	*P[A_1]	30%	FINE
3:Joint	*P[BTM]	30%	FINE
4:Joint	*P[R_1]	30%	FINE
5:Joint	*P[R_2]	30%	FINE
Teach Route Points			
BACK	POINT	RECORD	DONE >

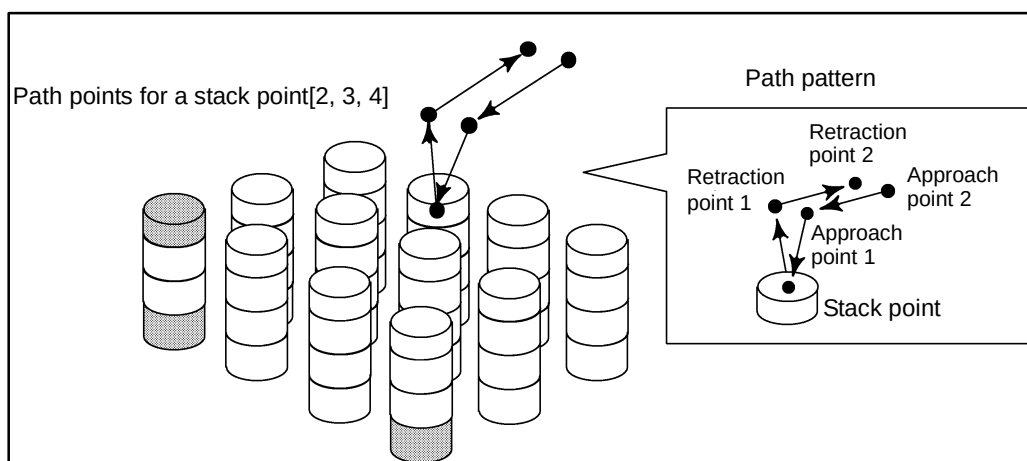


Fig. 10.3.5 Palletizing Path

Procedure 10-5 Teaching a palletizing path pattern

Step

- 1 Based on the initial data setting, a list of the path points to be taught is displayed.

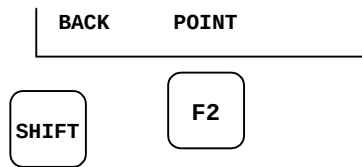
PROGRAM1		JOINT	30%
PALLETIZING Route Points			
IF PL[1] = [*, *, *]		1/3	
1:Joint	*P[A_2]	30%	FINE
2:Joint	*P[A_1]	30%	FINE
3:Joint	*P[BTM]	30%	FINE
4:Joint	*P[R_1]	30%	FINE
5:Joint	*P[R_2]	30%	FINE
Teach Route Points			
BACK	POINT	RECORD	DONE >

NOTE

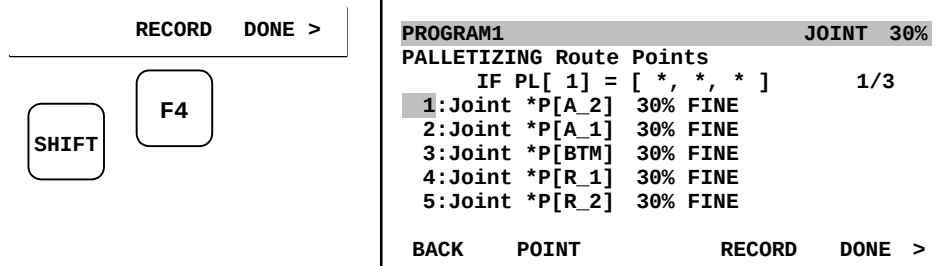
The number of path points to be recorded depends on the number of approach points and retraction points set on the initial data input screen. In this sample screen, the number of approach points is 2 and the number of retraction points is 2.

- 2 Move the robot by jog feed to a position which is to be taught as a path point.

- 3 Move the cursor to the setting field to be taught and teach the position using one of the following operation.
- a Press F2 “POINT” while holding down the SHIFT key. When pressing F2 “POINT”, without pressing and holding the SHIFT key, default logical motion menu is displayed and then you can set the motion type or feedrate, etc. (This key is displayed only at teaching route pattern points.)

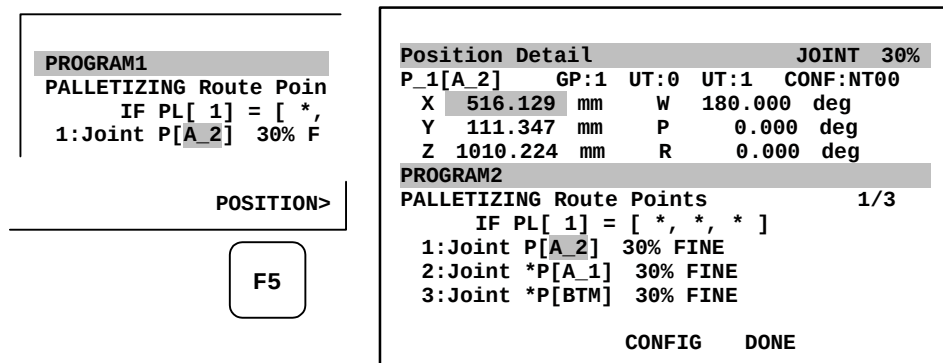


- b Press and hold the SHIFT key and press F4 “RECORD”.

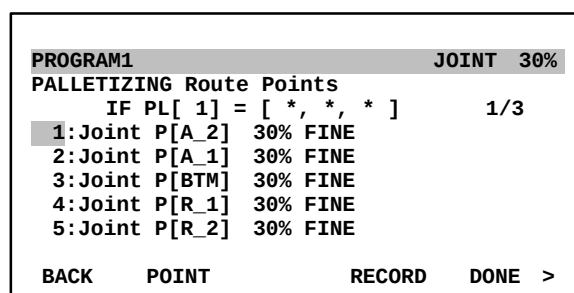


The positions not yet taught are marked with an asterisk *.

- 4 To display detailed position data for a taught path point, move the cursor to the target path point number, and press F5 “POSITION.” The detailed position data is then indicated.



Numeric values can also be entered directly to specify position data. To return to the previous path pattern teach screen, press F4 “DONE.”



- 5 Pressing the FWD key while holding down the SHIFT key causes the robot to move to the path point indicated by the cursor. This operation can be performed to confirm the taught point.



- 6 To return to the stacking pattern teach screen, press F1 “BACK.”

BACK

F1

- 7 Press F5 “DONE” to display the next path pattern teaching screen. If the path pattern is only one pattern, go to the step 9.

RECORD DONE >

F5

```

PROGRAM1                                JOINT 30%
PALLETIZING Bottom Points
IF PL[ 1] = [ *, *, * ]                2/3
1: *P[A_2]
2: *P[A_1]
3: *P[BTM]
4: *P[R_1]
5: *P[R_2]

BACK                                RECORD    DONE >

```

- 8 Press F1 BACK to specify the previous path pattern. Press F5 DONE to specify the next path pattern.
- 9 After teaching of all the path patterns is complete, press F5 DONE to exit from the palletizing edit screen and display the program screen again. The palletizing instruction is automatically written in the program.

```

PRG1                                JOINT 30%
                                13/13
5: L P[2] 300mm/sec CNT50
6: PALLETIZING-EX_4
7: P PAL_4[A_2] 30% FINE
8: P PAL_4[A_1] 30% FINE
9: P PAL_4[BTM] 30% FINE
10: P PAL_4[R_1] 30% FINE
11: P PAL_4[R_2] 30% FINE
12: PALLETIZING-END_4
[End]

POINT                                TOUCHUP >

```

- 10 Editing, such as modifying a hand instruction executed at a stack position or the motion format at a path point can be performed on this program screen, in the same way as for normal programs.

PALLET1		JOINT 30%
		14/14
5:	L P[2] 300mm/sec CNT50	
6:	PALLETIZING-EX_4	
7:	P PAL_4[A_2] 30% CNT30	
8:	P PAL_4[A_1] 30% CNT30	
9:	P PAL_4[BTM] 30% FINE	
10:	hand open	
11:	P PAL_4[R_1] 30% CNT30	
12:	L PAL_4[R_2] 300mm/sec CNT30	
13:	PALLETIZING-END_4	
[End]		
POINT		TOUCHUP >

For details of palletizing programs, see Section 10.4, “Executing the Palletizing Function.”

10.3.6 Notes on Teaching the Palletizing Function

- The palletizing function is enabled only when a program contains these three instructions: A palletizing instruction, palletizing motion instruction, and palletizing end instruction. When just one of the three instructions is taught into a subprogram by another operation such as copying, normal operation cannot be performed.
- When all palletizing data has been taught, palletizing numbers are automatically written together with the instructions (a palletizing instruction, palletizing motion instruction, and palletizing end instruction). The user need not be concerned about the duplication of these numbers in other programs. (Each program has its own data for palletizing numbers.)
- In the palletizing motion instruction, C (circular motion) cannot be specified as the motion format.
- When palletizing, with a system with extended axes, there are some special conditions. For a system with extended axes, refer to Section 10.6 “Palletizing Function with extended axes”
- When the position data in the stack points and the path points are taught, user frame is not used. World frame is always used in teaching these points, and user frame number is the position data of these points is always 0.

10.4 EXECUTING THE PALLETIZING FUNCTION

The execution of the palletizing function is shown below.

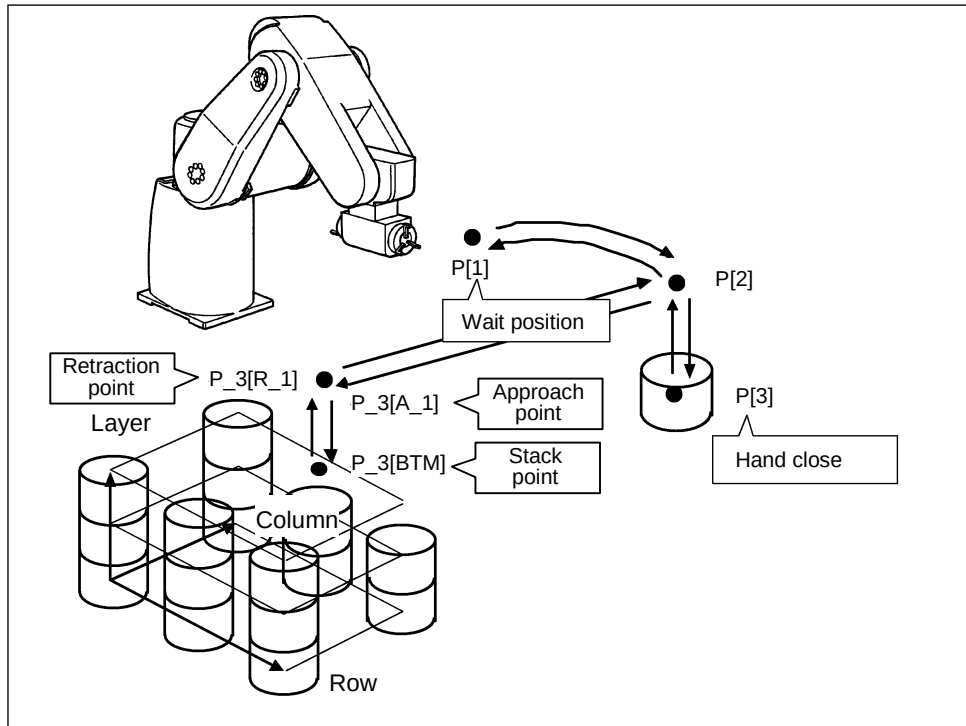


Fig. 10.4 (a) Example of Palletizing

Example

```

5: J P[1] 100% FINE
6: J P[2] 70% CNT50
7: L P[3] 50mm/sec FINE
8: hand close
9: L P[2] 100mm/sec CNT50
10: PALLETIZING-B_3
11: L PAL_3[ A_1 ] 100mm/sec CNT10
12: L PAL_3[ BTM ] 50mm/sec FINE
13: hand open
14: L PAL_3[ R_1 ] 100mm/sec CNT10
15: PALLETIZING-END_3
16: J P[2] 70% CNT50
17: J P[1] 100% FINE

```

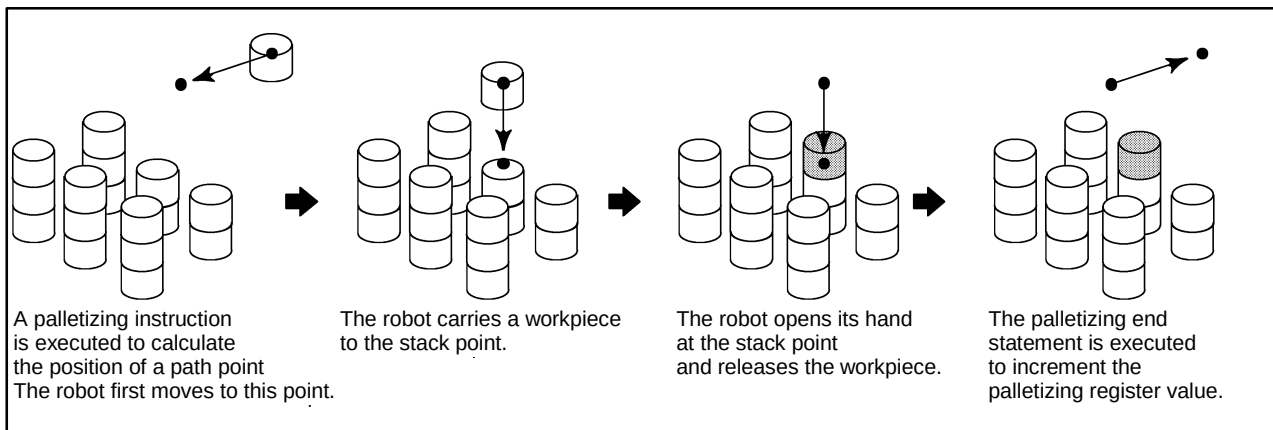


Fig. 10.4 (b) Workpiece Stacking Process

10.4.1 Palletizing Register

The palletizing register manages the position of the current stack point. When a palletizing instruction is executed, the value held in the palletizing register is referenced, and the actual stack point and path points are calculated. (See Section 7.5, “PALLETIZING REGISTER.”)

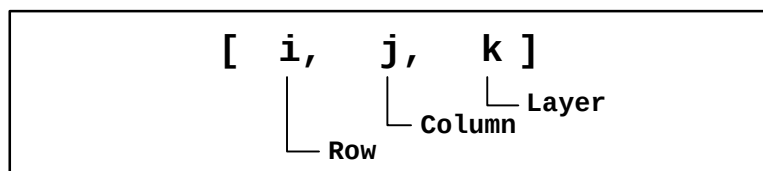


Fig. 10.4.1 (a) Palletizing Register

The palletizing register indicates the row, column, and layer which are used for calculating the position of the stack point when the palletizing instruction is executed.

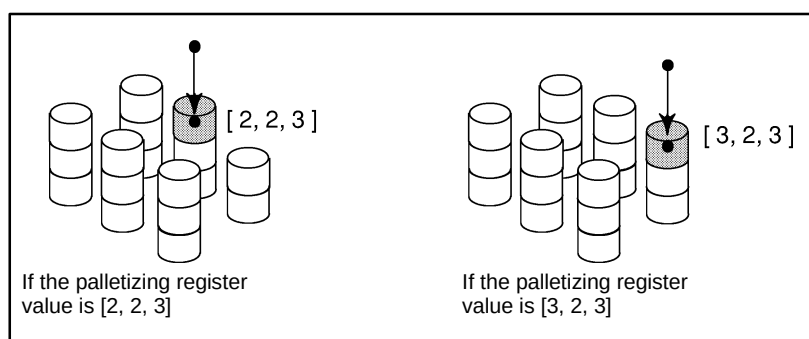


Fig. 10.4.1 (b) Relationship between the Palletizing Register and Stack Point

Updating the Palletizing Register

The palletizing register is incremented (or decremented) by executing the palletizing end instruction. The increment (decrement) method is determined depending on the initial data setting.

For 2-row, 2-column, and 2-layer palletizing with ORDER = [RCL] specified, executing the palletizing end instruction changes the palletizing register as follows:

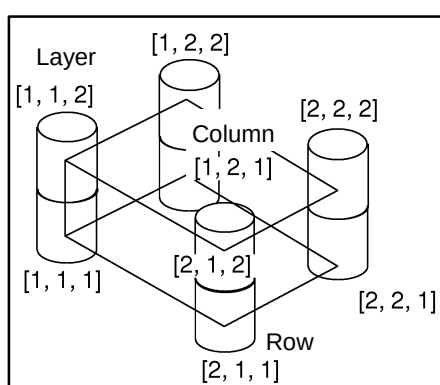


Fig. 10.4.1 (c) Example of 2-Row, 2-Column, and 2-Layer Palletizing

Table 10.4.1 (a) Incrementing (Decrementing) Order of the Palletizing Register

	TYPE = [PALLET]		TYPE = [DEPALLET]	
	INCR = [1]	INCR = [-1]	INCR = [1]	INCR = [-1]
Initial value	[1, 1, 1]	[2, 2, 1]	[2, 2, 2]	[1, 1, 2]
↓	[2, 1, 1]	[1, 2, 1]	[1, 2, 2]	[2, 1, 2]
↓	[1, 2, 1]	[2, 1, 1]	[2, 1, 2]	[1, 2, 2]
↓	[2, 2, 1]	[1, 1, 1]	[1, 1, 2]	[2, 2, 2]

	TYPE = [PALLET]		TYPE = [DEPALLET]	
	INCR = [1]	INCR = [-1]	INCR = [1]	INCR = [-1]
Initial value	[1, 1, 2]	[2, 2, 2]	[2, 2, 1]	[1, 1, 1]
↓	[2, 1, 2]	[1, 2, 2]	[1, 2, 1]	[2, 1, 1]
↓	[1, 2, 2]	[2, 1, 2]	[2, 1, 1]	[1, 2, 1]
↓	[2, 2, 2]	[1, 1, 2]	[1, 1, 1]	[2, 2, 1]
↓	[1, 1, 1]	[2, 2, 1]	[2, 2, 2]	[1, 1, 2]

Initializing the palletizing register

When F5 “DONE” is pressed upon completion of the setting or changing of initial palletizing data, the palletizing stacking pattern teach mode is set. At this time, the palletizing register is initialized automatically. (See Subsection 10.3.2, “Inputting Initial Data.”)

Table 10.4.1 (b) Initial Value of the Palletizing Register

Initial data		Initial value		
TYPE	INCR	ROWS	COLUMNS	LAYERS
PALLET	Positive value	1	1	1
	Negative value	Total number of rows	Total number of columns	1
DEPALLET	Positive value	Total number of rows	Total number of columns	Total number of layers
	Negative value	1	1	Total number of layers

10.4.2 Controlling the Palletizing Function by a Palletizing Register

In 5-row, 1-column, and 5-layer palletizing, suppress stack operation for the fifth workpiece in each even-numbered layer. (Stack five workpieces in odd-numbered layers, and stack four workpieces in even-numbered layers.)

PROGRAM2 JOINT 30% 1/17	
1: PL[1] = [1,1,1]	Palletizing register [1] is loaded with [1,1,1].
2: LBL[1]	
3: IF PL[1] = [5,*,2-0], JMP LBL[2]	If the row is 5 and the layer is an even number, a jump to label [2] is performed.
4: L P[1] 100mm/sec FINE	
5: hand close	
6: PALLETIZING-B_1	
7: L PAL_1[A_2] 100mm/sec CNT30	
8: P PAL_1[BTM] 50mm/sec FINE	
9: hand open	
10: L PAL_1[R_1] 300mm/sec CNT30	
11: LBL[2]	
12: IF PL[1] = [5,1,5], JMP LBL[3]	If the row, column, and layer value are [5,1,5], a jump to label [3] is performed.
13: PALLETIZING-END_1	
14: JMP LBL[1]	
15: LBL[3]	
16: END	Set the values of the row, column and layer for the next stack operation in the palletizing register [1].
[End]	
POINT	TOUCHUP >

Procedure 10-6 Displaying the palletizing status

```

PROGRAM1                                JOINT 30%
                                           6/82
5: L P[2] 300mm/sec CNT50
6:  PALLETIZING-B_1
7: L PAL_1[A_2] 1000cm/min CNT30

[MODIFY]                                [CHOICE] LIST >

```

Step

- 1 To display the palletizing status, place the cursor on the palletizing instruction, then press F5 “[LIST].” The current stack point and the value of the palletizing register are displayed.

```

LIST >
F5
Pallet List                                JOINT 30%
PALLETIZING-B_1
CURRENT BOTTOM POINT [ 3, 1, 1 ]
PALLET REGISTER [ 1]=[ 4, 1, 1 ]
ROUTE PATTERN[ 1]=[ *, *, * ]
PROGRAM1                                6/82

```

10.5 MODIFYING THE PALLETIZING FUNCTION

The palletizing data and palletizing instructions which were taught can be modified later.

Procedure 10-7 Modifying palletizing data

```

PROGRAM1                                JOINT 30%
                                           6/82
5: L P[2] 300mm/sec CNT50
6:  PALLETIZING-B_1
7: L PAL_1[A_2] 1000cm/min CNT30

Select item
[MODIFY]                                [CHOICE] LIST >

```

Step

- 1 Place the cursor on the palletizing instruction which is to be modified, then press F1 “[MODIFY]” to display the modification menu.
- 2 Select a target palletizing edit screen in the modification menu.

```

1 CONFIG
2 BOTTOM
3 PATTERN
4 ROUTE
MODIFY
F1 ENTER
PROGRAM1                                JOINT 30%
PALLETIZING Bottom Points
1:--P [ 1, 1, 1 ]--
2:--P [ 4, 1, 1 ]--
3:--P [ 1, 3, 1 ]--
4:--P [ 1, 1, 5 ]--
[End]
BACK                                RECORD DONE >

```

To return to the previous palletizing edit screen, press F1 “BACK.”

To proceed to the next palletizing edit screen, press F5 “DONE.”

NOTE

After palletizing data has been modified, the normal edit screen can be called from any palletizing screen. In this case, the new data after modification is kept valid.

- When the modification is completed, press NEXT ">" to display the next page. Then press F1 "PROG."

PROG

F1

Procedure 10-8 Changing the palletizing number

```

PROGRAM1                                JOINT 30%
                                         6/52
6:  PALLETIZING-B_1
7:  L PAL_1[A_1] 300mm/sec CNT30
8:  L PAL_1[BTM] 50mm/sec FINE
9:    hand open
10: L PAL_1[R_1] 300mm/sec CNT30
11: PALLETIZING-END_1

```

Step

- Place the cursor on the palletizing instruction having the palletizing number which is to be changed, then enter a new number.

PROGRAM1

6: PALLETIZING-B_1

2

ENTER

```

PROGRAM1                                JOINT 30%
                                         6/52
6:  PALLETIZING-B_2
7:  L PAL_2[A_1] 300mm/sec CNT30
8:  L PAL_2[BTM] 50mm/sec FINE
9:    hand open
10: L PAL_2[R_1] 300mm/sec CNT30
11: PALLETIZING-END_2

```

At the same time the palletizing number of the palletizing instruction is changed, the palletizing numbers of the palletizing motion and palletizing end instructions are also changed.

**CAUTION**

When changing palletizing numbers, make sure that the new numbers are not used by other palletizing instructions.

10.6 PALLETIZING FUNCTION WITH EXTENDED AXES

This section explains the special items when using palletizing on a system with extended axes. When teaching the bottom point or route points for palletizing, which are different from the usual teaching of motion instruction, the position which is removed the position of extended axes is recorded.

Execution

When palletizing is executed, palletizing will be done at the position of the extended axes at that time. (The robot does not automatically return to the position of the extended axes at teaching bottom points or route points.) For example, on a system with a linear extended axis, when the palletizing instruction is executed at a point 1000 mm away from the position at teaching palletizing, the robot will perform the same motion as taught at the position which is 1000 mm away from the teaching position.

In case that the skip instruction is taught in the motion instruction immediately before the palletizing instruction, when the skip condition is satisfied during the execution of that motion instruction, the palletizing is executed at the position of the extended axis when the skip condition is satisfied.

Position teaching/modifying

When teaching the bottom/route points for palletizing on a system with extended axes or modifying the position, the following attention is necessary.

- When teaching bottom/route points for palletizing, it must be done at the fixed point.
 - When the position of extended axes shifts while teaching the bottom/route points for the same palletizing, the robot will follow a different path than the taught path when the program is executed.
- When modifying the position, move the robot to the position of the extended axes at teaching bottom/route points before modifying.

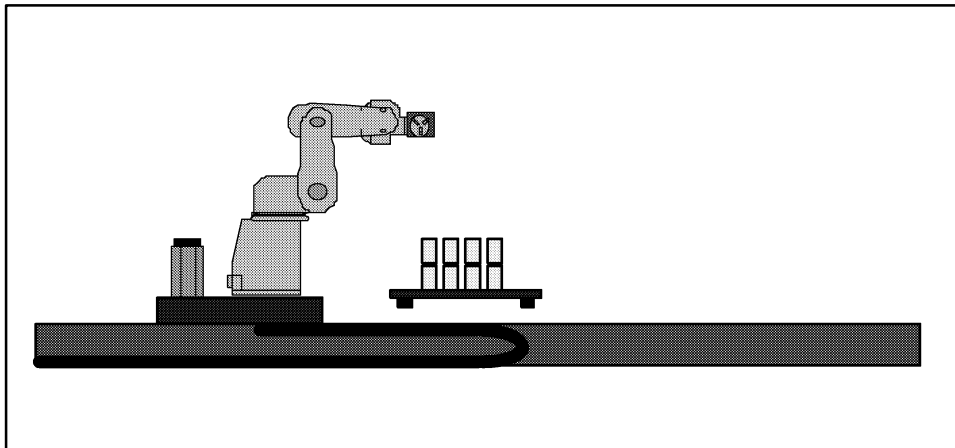


Fig. 10.6 Palletizing function with extended axes

10.7 PALLETIZING ALL-POINT TEACHING

Palletizing all-point teaching allows the stacking (or unstacking) of workpieces without changing the taught configuration by changing a system variable.

Operation

Set the following values to stack (or unstack) workpieces without changing the taught posture and configuration:

- 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 INTER for posture control in the row, column, or layer direction for which FREE is specified as the arrangement mode.

For each taught workpiece in the specified direction, all workpieces corresponding to the taught workpiece are stacked (or unloaded) with the same posture and configuration as that for the taught workpiece.

The following shows an example of palletizing for an irregular arrangement of four rows, two columns, and five layers.

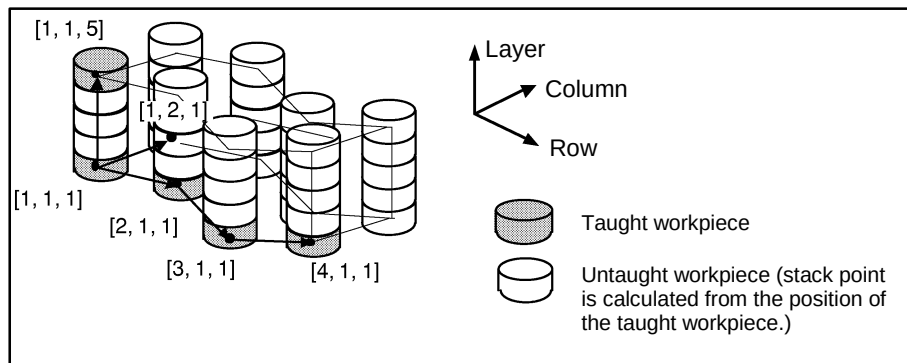


Fig. 10.7 Palletizing All-Point Teaching

The following palletizing initial data is set:

```
ROWS=      [4   FREE  INTER]
COLUMNS =  [2   LINE  FIX  ]
LAYERS =    [5   LINE  FIX  1]
```

In this example, FREE and INTER are set in the row direction. When system variable \$PALCFG.\$FREE_CFG_EN is set to TRUE under this condition, the following configurations are used:

- Configuration for stacking (unloading) workpieces on row 1: Configuration in P[1,1,1]
- Configuration for stacking (unloading) workpieces on row 2: Configuration in P[2,1,1]
- Configuration for stacking (unloading) workpieces on row 3: Configuration in P[3,1,1]
- Configuration for stacking (unloading) workpieces on row 4: Configuration in P[4,1,1]

Notes

Note the following points when using this function:

- 1 FREE and INTER can be set at the same time in only one of the row, column, and layer directions (when this function is not to be used, set system variable \$PALCFG.\$FREE_CFG_EN to FALSE).

This is because if FREE and INTER are set at the same time in two or more of the row, column, and layer directions, two or more configurations to be taken at the position of an untaught workpiece (workpiece for which the stack point is calculated from the position of a taught workpiece) are made.

If such a setting is made in a program, the program causes the error indicated by PALT-024 Calculation error occurred and cannot be executed.

- 2 Carefully teach a program so that the program is not stopped due to a configuration mismatch alarm.

If the configuration in the current position differs from the configuration data for the destination position, the robot cannot move in Linear operation mode (a configuration mismatch alarm occurs and execution of the program is stopped).

The configuration at a stack point is used for the configuration at an approach or retraction point during palletizing. Therefore, if the pallet operation instruction to be executed first is in Linear operation mode, a configuration mismatch may occur depending on the configuration of the robot when an attempt is made to execute the line.

To avoid such a problem, specify Joint for the operation mode of the first pallet operation instruction.

For example, to avoid a configuration mismatch alarm, the following programming can be used for palletizing with three approach points and two retraction points.

```
:
10:  PLLETIZING-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 hand 1
16:L PAL_1[R_1] 300mm/sec CNT10
17:L PAL_1[R_2] 500mm/sec CNT50
18: PALLETIZING-END_1
:
```

11 FANUC *i*Pendant

Contents of this chapter

- 11.1 OVERVIEW
- 11.2 APPEARANCE AND OPERATIONS
- 11.3 SETTING UP *i*Pendant
- 11.4 SOFT PANEL

11.1 OVERVIEW

The *i*Pendant is an Internet-type robot teach pendant having a large color liquid-crystal display panel. This teach pendant allows you to reference multiple data items simultaneously and its visibility has been remarkably increased. The current user interface is also available with this teach pendant, so those who are familiar with operations of conventional teach pendants can use this teach pendant easily.

This section mainly describes differences between the conventional teach pendant and *i*Pendant. Operations which are not described in this manual are common to the conventional teach pendant and *i*Pendant.

The following items are different from those of the conventional teach pendant:

- Software LED display
While the conventional teach pendant uses 11 LEDs to display the status, the *i*Pendant displays the status with icons in the status window on the screen.
- Screen split function
The *i*Pendant can display two or three split screens as well as one screen to enable multiple data items to be checked at a time.
- How to change the operation target screen when multiple screens are displayed
When multiple screens are displayed simultaneously, the operation target screen can be changed in turn.
- Function of displaying one screen and status sub-window
This function displays information such as the current position and safety signals with icons in the status sub-window (left screen).
- Screen menus displayed by pressing the MENU key and those displayed on the edit screen.
The screen menus, which are all displayed at a time allow you to quickly move to a desired screen and quickly insert a required command.
- Internet browser screen
You can enter a URL to access data on the network.
- Color display according to the alarm severity
Each message on the alarm list screen is displayed in the color specified according to its severity.

The latest model of *i*Pendant has USB function. The followings are the improvement points of the latest model.

- The case design is changed for easy grip and good weight balancing. The location of the enable switch and the emergency stop button is changed to avoid miss-operation.
- Key location change
The location of the left shift key and the PREV key are swapped, and the location of the right shift key and the NEXT key are swapped for easy grip and easy operation.
- LED lamp on key sheet

A green LED lamp, that is the indicator for cycle power status of a robot controller, and a red LED lamp, that is the indicator for alarm status of a robot controller, are prepared on the key sheet.

- Group key
A new key is added on the key sheet for easy change of the current motion group.
- USB port
A USB port to support USB memory is prepared on the right side of the case.

It is possible that some or all of the above functions are not available on your robot because either of that the functions are not supported yet in your controller software or the firmware of your *i*Pendant, or some of the functions are optional and your controller does not have the optional.

The *i*Pendant is an optional function.

NOTE

You can confirm the *i*Pendant firmware version by the following method.

- 1 Press MENU key, select NEXT, select STATUS. Press F1:TYPE, select Version ID, then version identification screen is displayed. The item of the TP core firmware shows the *i*Pendant firmware version. It might be showed as Vx.xxxx (x is a numeric number, for example V7.5063)

The following method 2 also can be used to confirm the *i*Pendant firmware version. The lower two figures of the version got by the method 2 are different from the version got by the method 1 because of the difference of the numeration of the version but the both mean same firmware version.

- 2 At the upper side of the first *i*Pendant screen after cycle power on, the firmware version is displayed as the message "iPendant Vx.xxP/xx" (x is a numeric number, for example V7.50P/01).

11.2 APPEARANCE AND OPERATIONS

This section describes the appearance of the *i*Pendant and operations specific to the *i*Pendant.

11.2.1 Appearance and Switches

Fig. 11.2.1(a) and Fig. 11.2.1(b) show the locations of the emergency stop button, teach pendant enable switch, and deadman switches.

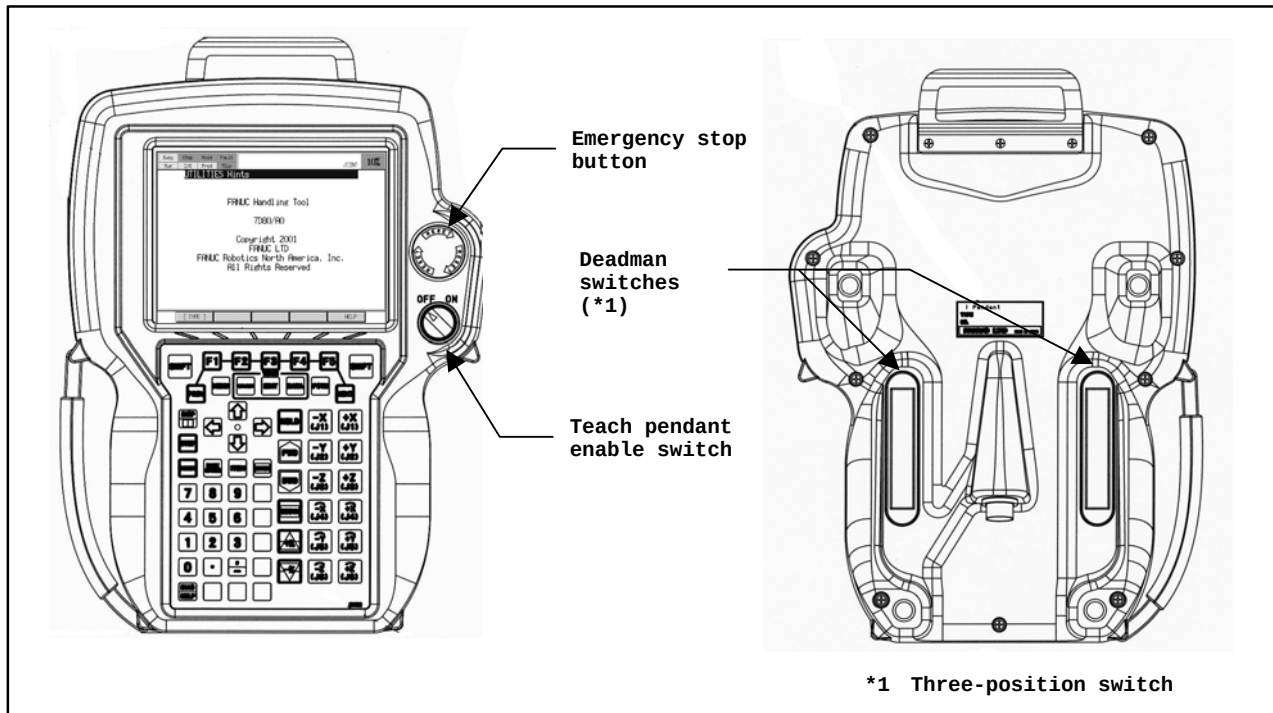
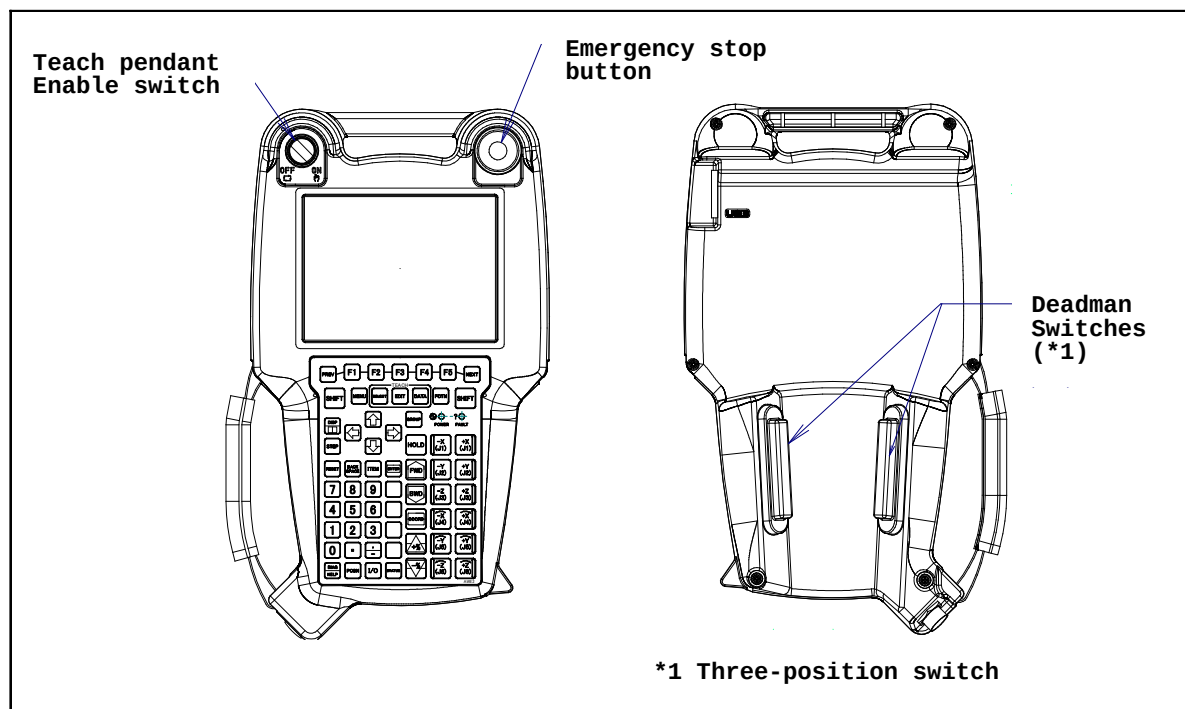


Fig. 11.2.1(a) Teach pendant switches

Fig. 11.2.1 (b) Teach pendant switches (New *i*Pendant)

11.2.2 Key Switches

Fig. 11.2.2 (a) and Fig. 11.2.2 (b) shows the *i*Pendant key switch layout. This subsection describes the screen focus change/screen split key, Diagnose/Help key and group key specific to the *i*Pendant.

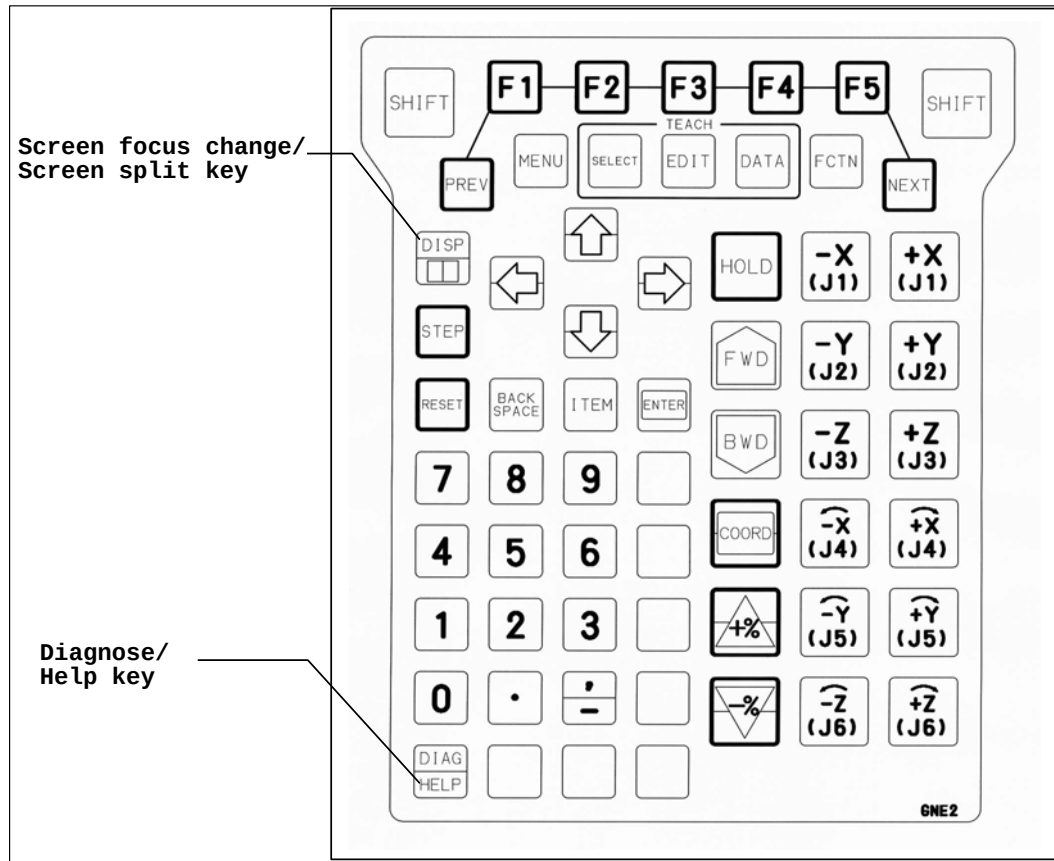


Fig. 11.2.2 (a) Teach pendant key switches

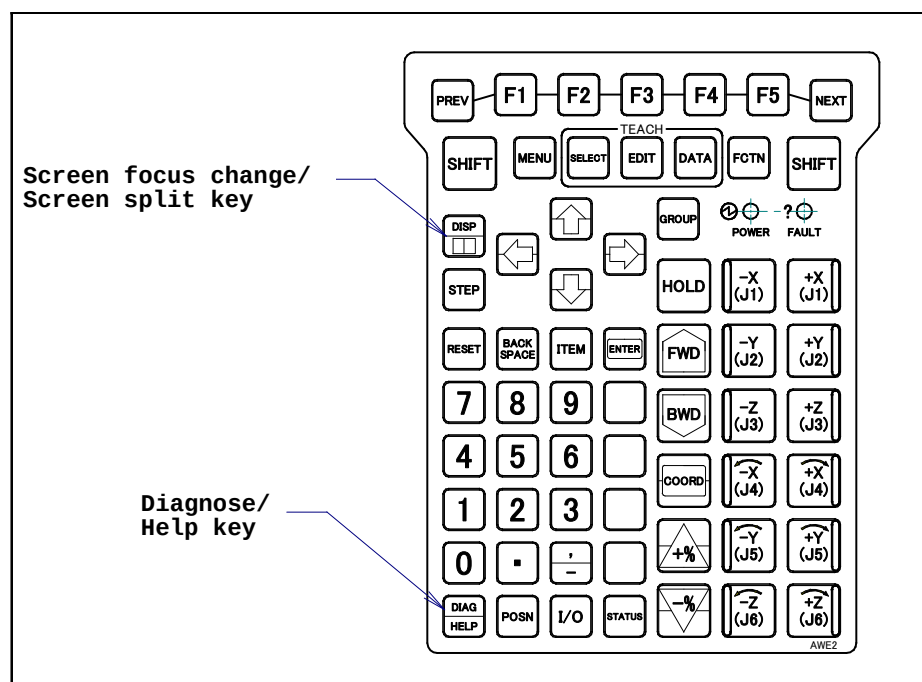


Fig. 11.2.2 (b) Teach pendant key switches (New *i*Pendant)

Table 11.2.2 Menu-related key switches

Key	Function
	Pressing this key singly changes the operation target screen. Pressing this key together with SHIFT key splits the screen (single Screen, double screens, triple screens, or status/single screen).
	Pressing this key singly moves to the hint screen. Pressing this key together with the SHIFT key moves to the alarm screen.
	Pressing this key singly changes the current motion group and sub-group selection step by step like this: G1, G1S, G2, G2S, G3, ... G1. Pressing a numeric key, that is identical with the group number you wish to select, with a GROUP key selects the wished motion group directly. Pressing a 0 key with a GROUP key increases the subgroup number in the selected group. Only the latest <i>i</i> Pendant has this key.

NOTE

The group key is available only in the condition that J601 Multi motion group option or J518 Extended axes control option is ordered and a extended axis or a independent axis has been added and set up. Also the group key is available on the controller software 7DA3/18 or later, 7DA4 series, 7DA5 series or the later series from 7DA5.

11.2.3 Touch Panel

*i*Pendant provides a touch panel as an option.

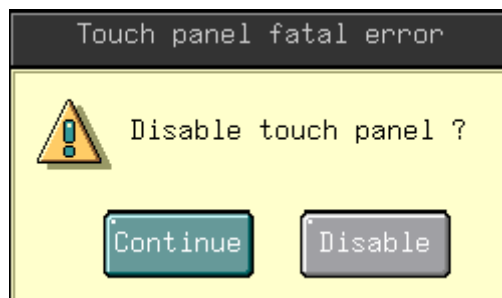
The screens on which operations can be performed using the touch panel are as follows.

Note that not all operations can be performed using the touch panel.

- Operation Panel screen / BROWSER screen (Web browser screen) / Status sub-window screen
- Software keyboard
- Screen switching (When multiple screens are displayed, moving to the desired screen is accomplished by touching the screen.)
- Software button from F1 to F5 placed below the screen.

NOTE

- If you press 2 or more places on touch panel, touch panel may recognize wrong place, which is different from touched place. Please touch only one place on touch panel.
- When touch panel is broken, the panel can recognize wrong place that are different from place operator touched. Don't use touch panel to set operations that can affect safety of system.
- Use touch panel by finger or dedicated pen for touch panel operation. If you use sharp object like usual pens, touch panel may get broken.
- When following dialog box is displayed on *i*Pendant, touch panel may be broken. Turn down robot controller and exchange *i*Pendants.



11.2.4 Status Window

The window at the top of the *i*Pendant screen is called the status window. In this window, eight software LEDs, alarm indication, and override value are displayed.

Each software LED is "on" when displayed together with an icon or "off" when displayed with no icon.

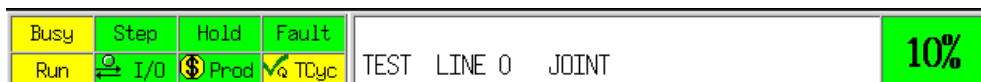


Table 11.2.4 Description of software LEDs

LEDs (Upper: On, Lower: Off)	Description
Busy  	Indicates that the robot is working. This LED is on during execution of a program.
Step  	Indicates that the robot is in the step operation mode.
Hold  	Indicates that the HOLD button is being held or the HOLD signal is input.
Fault  	Indicates that an alarm occurs.
Run  	Indicates that a program is being executed.
I/O  	Application-specific LED. This is a sample LED for a handling tool.
Prod  	Application-specific LED. This is a sample LED for a handling tool.
TCyc  	Application-specific LED. This is a sample LED for a handling tool.

11.2.5 Splitting the Screen

Pressing  key together with the SHIFT key displays the following screen menu:

In case that the software series of the controller is 7DA5 or later, the wide screen is added. For detail of wide screen, please refer to 11.2.14.8.

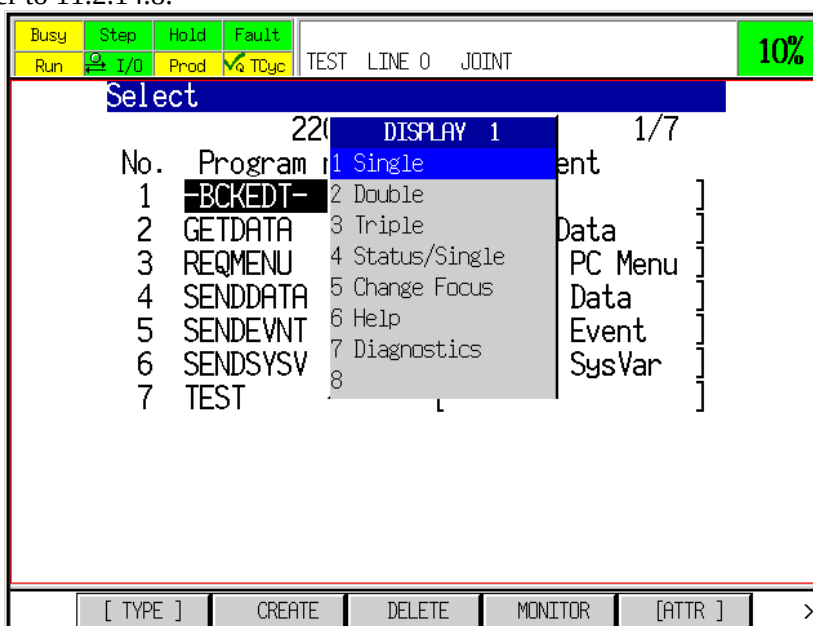


Fig. 11.2.5 (a) Screen split menu (one screen display)

Table 11.2.5 Description of the screen split menu

Item	Description
Single	Displays only one data item on the screen. The screen is not split.
Double	Splits the screen into right and left screens.
Triple	Splits the right screen into top and down screens and displays a total of three screens.
Status/Single	Splits the screen into right and left screens. The right screen is slightly larger than the left screen and the status sub-window with icons is displayed on the left screen.
Change focus	Changes focus of the operation target screen when multiple screens are displayed.

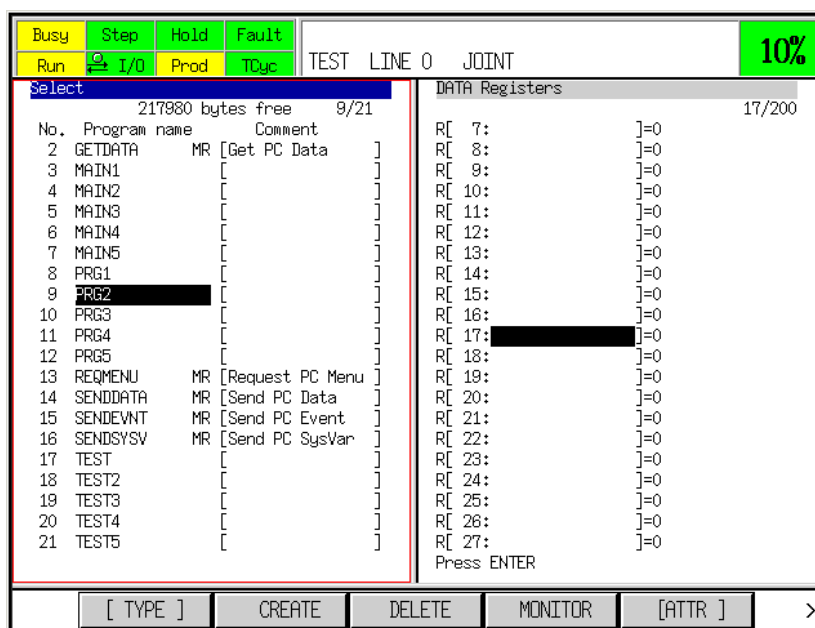


Fig. 11.2.5 (b) Example of displaying double screens

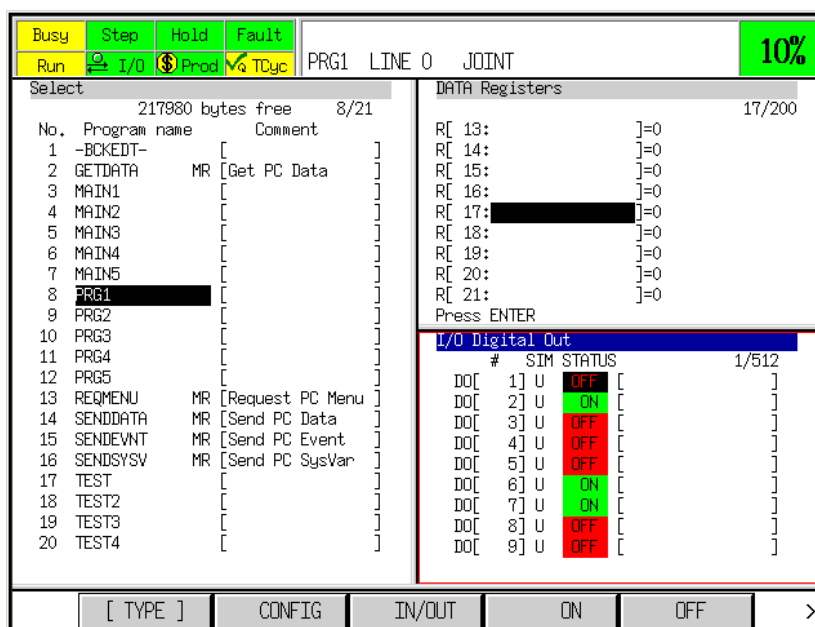


Fig. 11.2.5 (c) Example of displaying triple screens

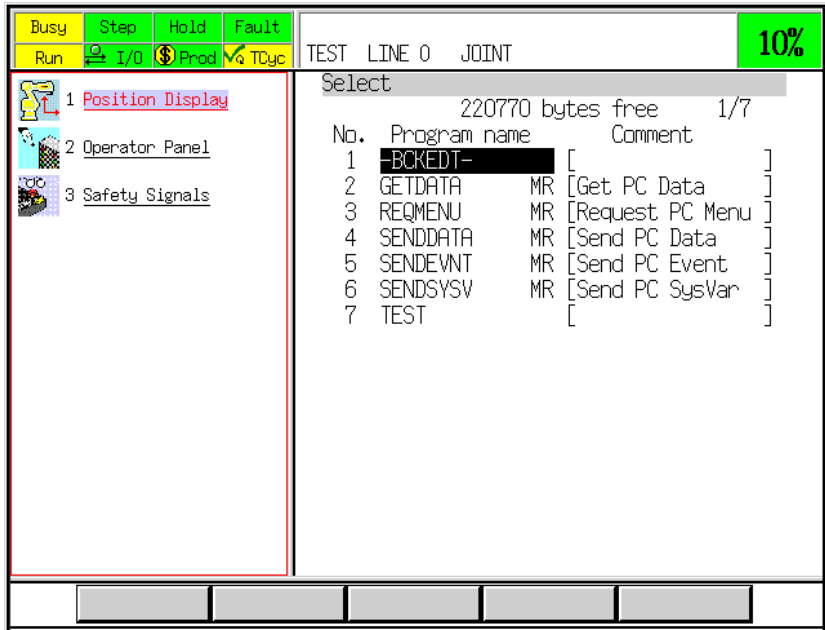


Fig. 11.2.5 (d) Example of displaying the status/single screen

NOTE
When the amount of free memory in the DRAM memory is not enough, the screen is not split and the alarm TPIF-142 Menu limit exceeded occurs. When the alarm TPIF-142 Menu limit exceeded occurs, please check the rest of temporary memory in the memory status display screen.

11.2.6 Changing the Operation Target Screen

Pressing the  key changes the operation target screen in turn. The title line of the screen which can be operated is displayed in blue and the frame of the screen is displayed in red. Press the above key together with the SHIFT key. The following screen menu appears. By selecting “5. Change focus” from this menu, you can also change the operation target screen.

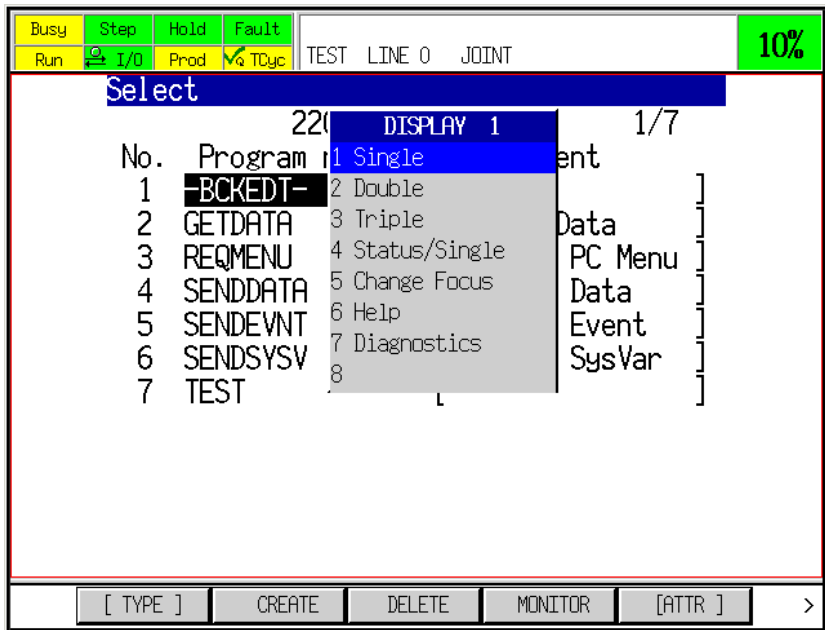


Fig. 11.2.6 Screen switch menu

11.2.7 Internet Browser Screen

To display the Internet browser screen, press the MENU key. The following screen menu appears:

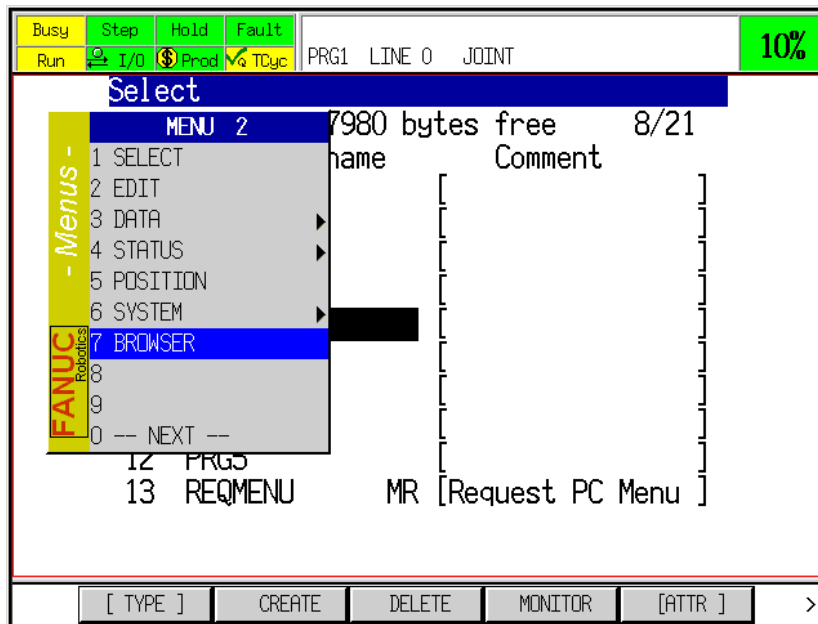


Fig. 11.2.7 (a) Screen menu

Select "BROWSER" from the screen menu. The following screen appears:

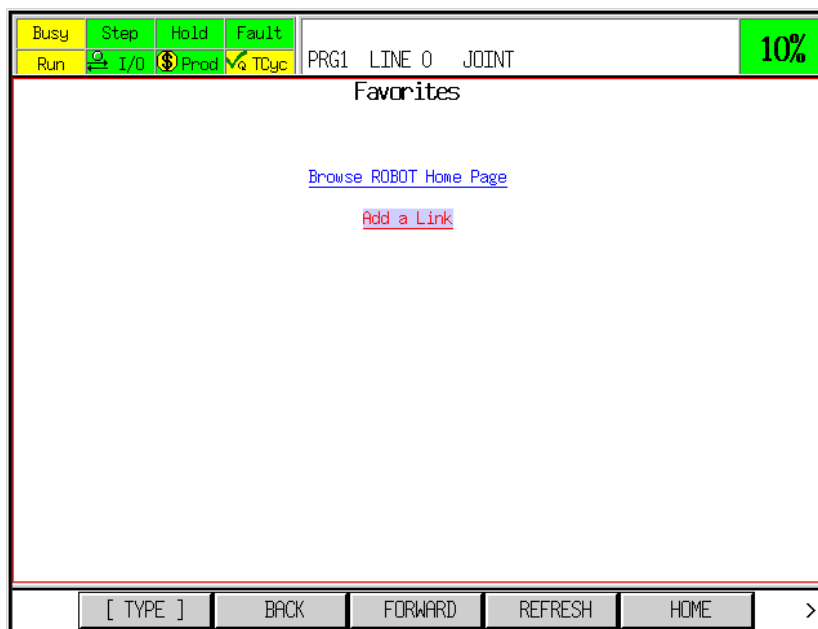


Fig. 11.2.7 (b) Internet browser screen

Select "Add a Link" and press the ENTER key. The following URL input screen appears:

Fig. 11.2.7 (c) URL input screen

Position the cursor on "Enter a Name" or "Enter an Address" in the above screen and press the ENTER key. The following software keyboard appears. Enter alphabetic and other characters. After confirming your entry, press the exit key at the lower right to exit the software keyboard.

In the following sample screen, the name and address have been entered. After you have entered the name and address, position the cursor on the continue button and press the ENTER key.

Enter a Name:

Enter an Address:

[TYPE] BACK FORWARD REFRESH HOME >

After the continue button is pressed, the registered link information is displayed as follows:

Favorites

[Browse ROBOT Home Page](#)

[Add a Link](#)

Browse Link	Modify Link Definition	Delete Link
FANUC LTD (http://www.fanuc.co.jp/eindex.htm)	Modify FANUC LTD	Delete FANUC LTD

[TYPE] BACK FORWARD REFRESH HOME >

Fig. 11.2.7 (d) Link screen

There are the following attentions to remove link at internet browser screen.

NOTE

Links definition is saved to CUSTLIST.DT when you backup using "All of above" from file screen. Zero size CUSTLIST.DT will be saved when you backup if you remove all links. And FILE-062 alarm can be occurred. Add any link if FILE-062 alarm is occurred.

11.2.8 Screen Selection Menu and Screen Menus on the Edit Screen

Pressing the MENU key displays the following screen menu. Positioning the cursor on a menu item with **▶** displays its submenus at a time.

Use the right arrow key to select the item corresponding to a screen to be displayed.

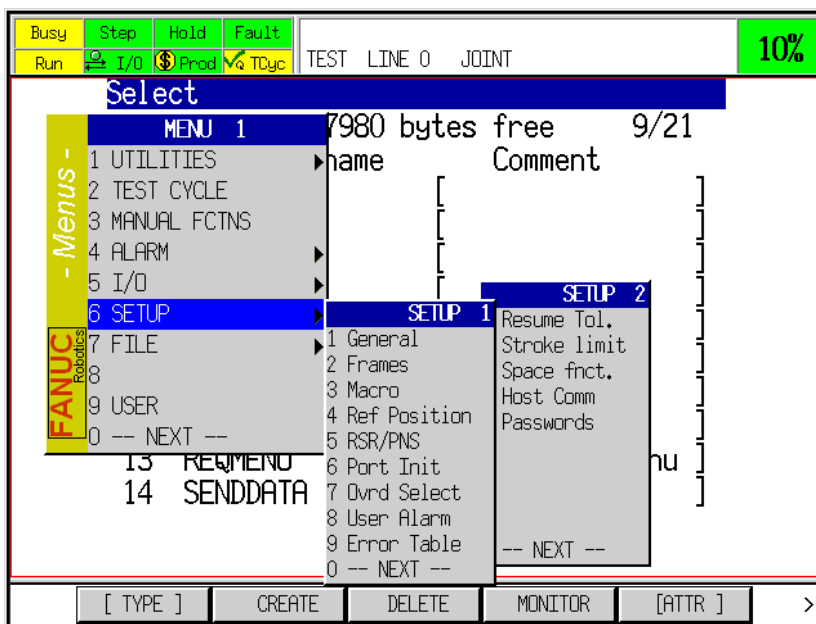


Fig. 11.2.8 (a) Screen selection menu

Pressing F1 "INST" on the edit screen displays the following screen menus. You can reference all commands at a time. Position the cursor on a desired command to insert it.

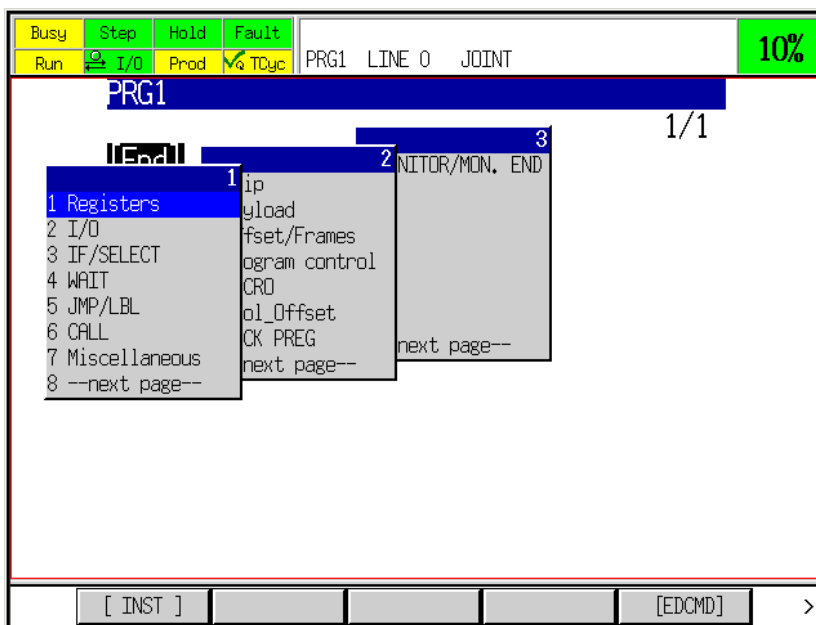
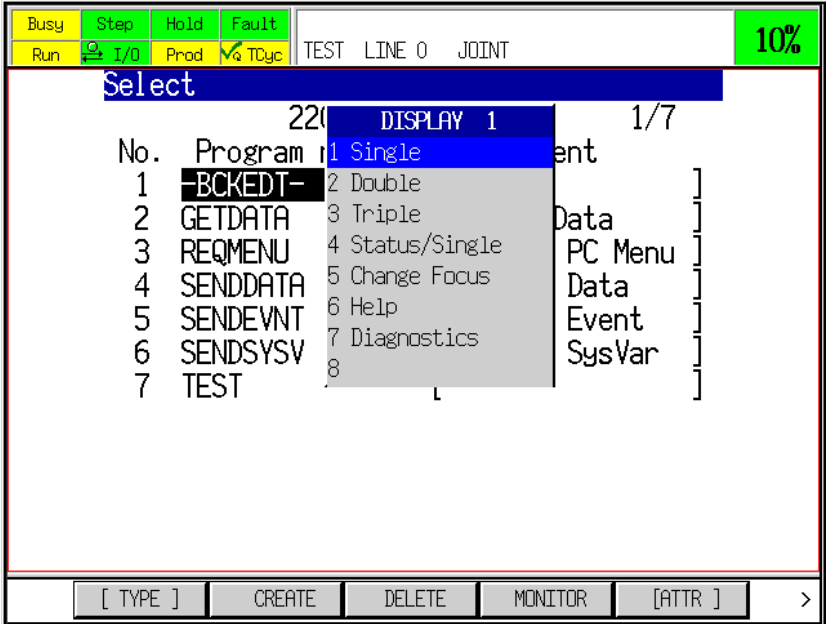


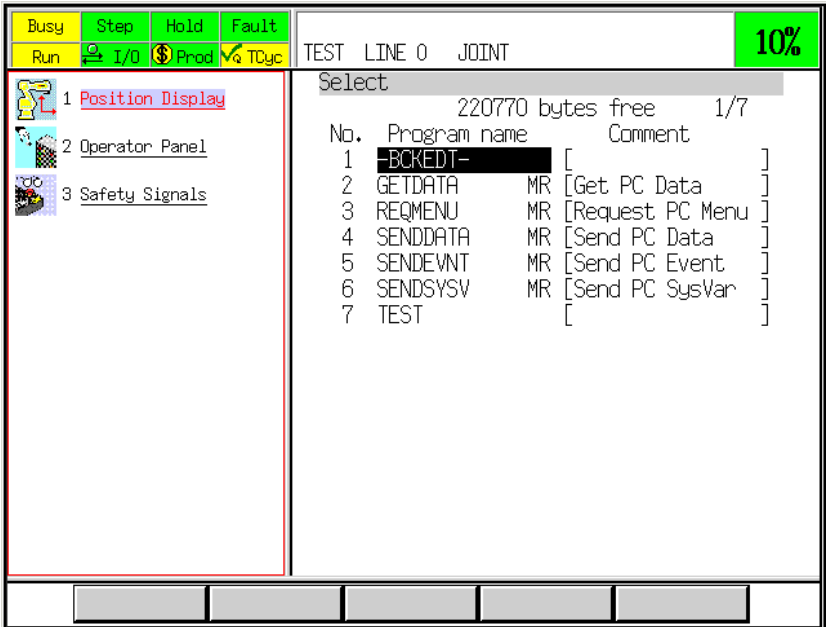
Fig. 11.2.8 (b) Screen menus on the edit screen

11.2.9 Status Sub-window

The status sub-window displays various types of statuses graphically. To display the status sub-window, select “4 Status/Single” from the following screen menu:

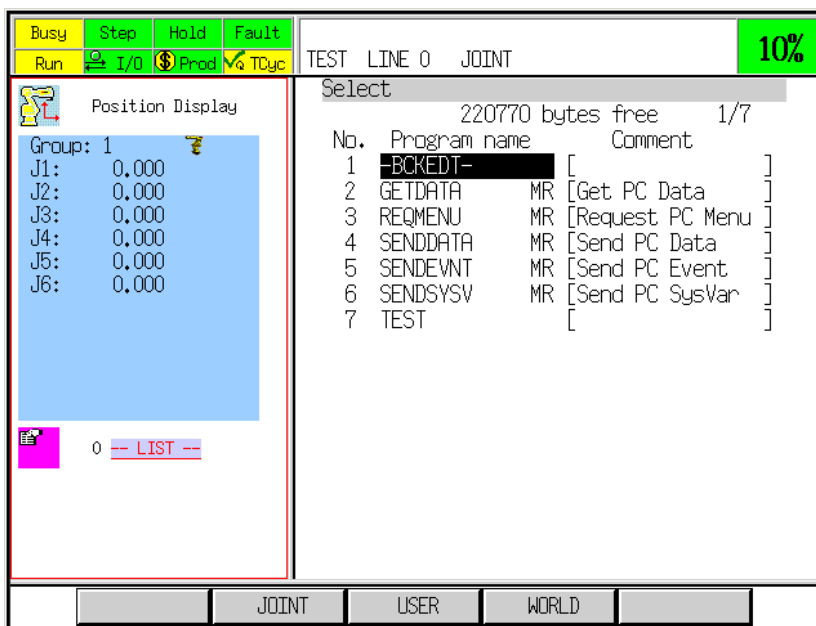


The left screen of the following two screens is the status sub-window. Position the cursor on Position Display, Operator Panel or Safety Signals and press the ENTER key. The corresponding status screen appears.



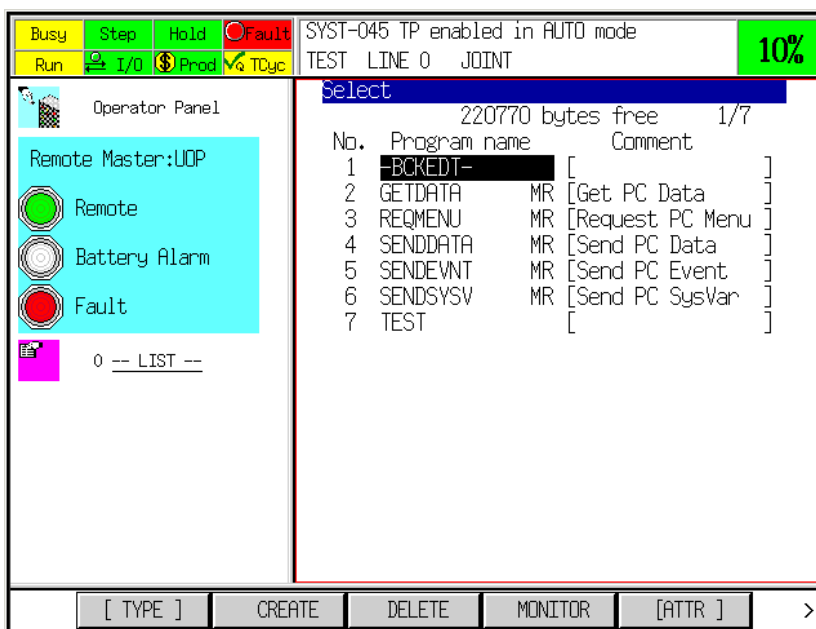
11.2.9.1 Current position display

When “Position Display” is selected, the following screen appears:



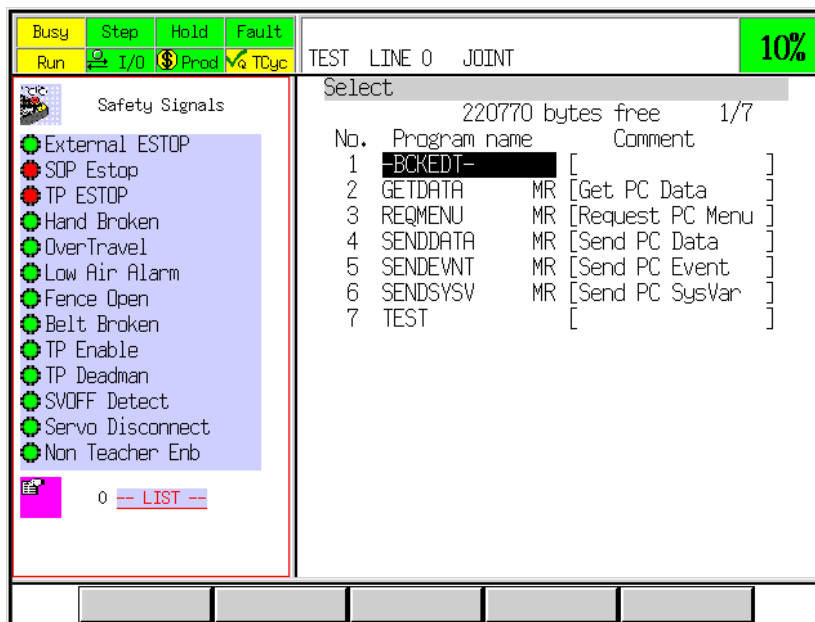
11.2.9.2 Operator panel status display

When “Operator Panel” is selected, the following screen appears. Each graphic indicator is on or off according to the status of the remote device.



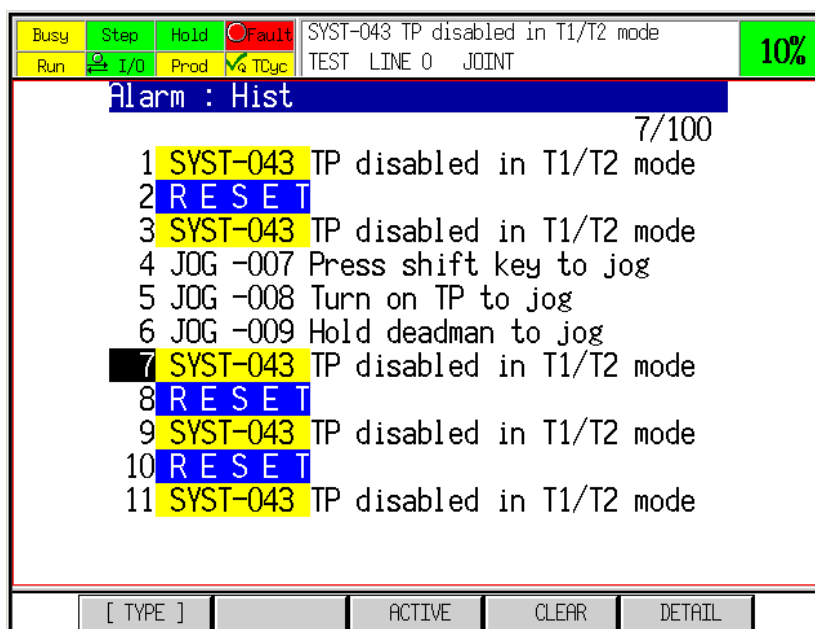
11.2.9.3 Safety signal status display

When “Safety Panel” is selected, the following screen appears. Each graphic indicator is on or off according to the status of the corresponding safety signal.



11.2.10 Color Display According to the Alarm Severity

On the following alarm history screen, each message is displayed in the color specified according to its alarm severity.



The color assigned to each alarm severity is listed below:

Alarm severity	Color	Description
NONE WARN	White	The program being executed is not affected.
PAUSE.L PAUSE.G	Yellow	The program being executed stops, but can be restarted after the cause of the alarm is removed.
STOP.L STOP.G	Yellow	
SERVO SERVO2	Red	The program being executed stops and cannot be restarted.
ABORT.L ABORT.G	Red	
SYSTEM	Red	
RESET(*)	Blue	
SYST-026 System normal power up(*)	Blue	

NOTE

Messages "RESET" and "SYST-026 System normal power up" are displayed in blue.

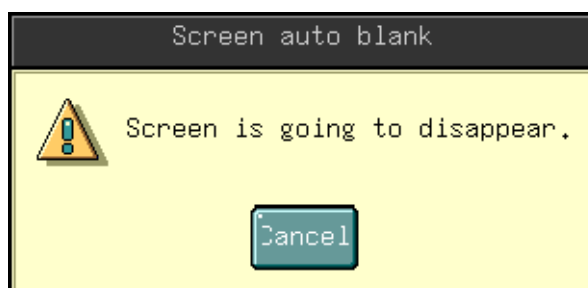
11.2.11 Backlight automatic blanking

The iPendant can turn off the backlight automatically for energy saving when any key is not pressed for a definite period of time. This function is disabled as the default setting. To use this function, please change the following system variables and turn off and on the controller cycle power.

\$TX.\$BLNK_ENABLE Backlight automatic blanking enable/disable
TRUE: enable, FALSE: disable
Default setting: FALSE

\$TX.\$BLNK_TIMER Period of time for waiting any key input. (minutes)
The value from 5 to 10000 is available to input.
Default setting: 5 minutes.

At 10 seconds before the screen blanking, the following message will be displayed as the prior announcement.



NOTE

- When the teach pendant enable switch is turned to the enable position, the automatic blanking is not performed. When any alarm is asserting, the automatic blanking is not performed.
- Until blanking, any key is pressed or an alarm is asserted, the displaying of *i*Pendant will be recovered soon. FANUC recommends to press a left or a right shift key if you want to recover the display.
- During blanking, any display is not visible on the screen. Don't judge the status of power cycle of your controller by whether display is visible or not on the screen. New *i*Pendant has a green LED indicator on a key sheet to show the status of a controller cycle power.

11.2.12 Operational Restrictions

- Two or three program edit screens can be displayed in the split windows at a time. But only the program opened in the left window is executable. To execute the program opened in the left window from *i*Pendant, please choose the left window as the current active window.
- Some screen, for example online touchup screen, can not be displayed at a same time in different windows.

11.2.13 USB Port (Soft option, New *i*Pendant only)

This function is software option. A USB port is available only on new *i*Pendant. Previous *i*Pendant does not have a USB port.

On the right side of *i*Pendant, a USB port is prepared. By using this port, customer can backup and restore controller files to/from a USB memory without going to the front of controller cabinet. The device name of the USB memory mounted on the *i*Pendant USB port is UT1:. Select the menu item "USB on TP (UT1:)" to set the current backup/restore device, then file backup and restore on FILE screen can work to the USB memory mounted on the *i*Pendant USB port.

USB memory (UT1:)

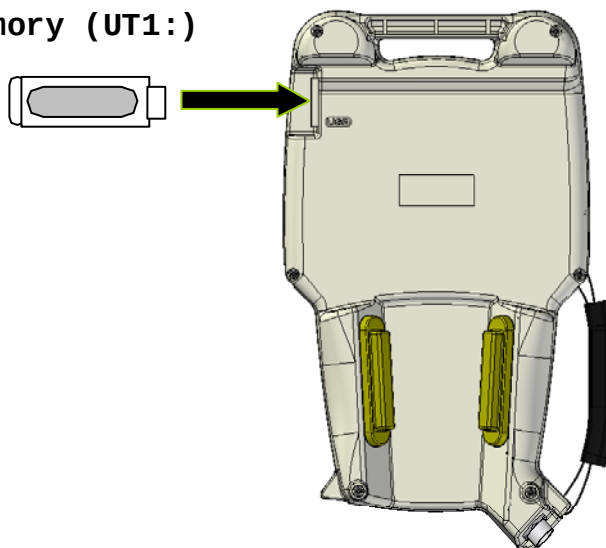


Fig. 11.2.13 USB memory (UT1:)

There are the following attentions to use *i*Pendant USB port.

- The "USB port on *iPendant*" software option (J957) should be ordered in your robot controller software to use *iPendant* USB port. This software option is available only in the case that robot controller software version is the later than 7DA3/18 or the one of 7DA4 or 7DA5 or later.
- Only USB memory is supported. Do not connect a USB device other than a USB memory to the *iPendant* USB port.
- When a USB memory unit is inserted into the USB port and recognized correctly, the message "USB device insert" will be displayed on your *iPendant* screen. If this message is not appeared at USB memory insertion, please remove the memory unit and insert it again gently.
- Confirm the access lamp of a USB memory is turned on and start to use it. If it is flashing, the USB memory is not ready and wait to use until it is turned on. Necessary time to ready differ depending on the USB memory product.
- While a file is being loaded or saved, do not remove the USB memory unit. Otherwise, the files in the USB memory unit may be damaged. If the USB memory unit is provided with an access lamp, make sure that the access lamp is not flashing before removing the unit.
- Image backup and restore function is not supported on the *iPendant* USB port.
- USB memory physical formatting function is not supported on the *iPendant* USB port.

NOTE

- *iPendant* does not have the circuit for calendar and clock. In case that the software series for *iPendant* firmware and the software of the robot controller is 7DA4 or prior to 7DA4, when the files are saved to USB memory using the USB port on *iPendant*, the creation data of the saved files is Jan,1,2098 without fail.
- In case that the software series for *iPendant* firmware and the software of the robot controller is 7DA5 or later, the software function to get the date and time by sending the date and time from the robot controller to *iPendant* at power on is added. Because this is the simulation by software, the date and time of the files saved to USB memory using USB port on *iPendant* is different from the date and time in the robot controller.

11.2.14 Operational improvement in 7DA5 or later series

The following operational improvements are implemented in 7DA5 or later series software.

- ICON menu: Some icons are displayed at the bottom of the screen for favorite screens, user views, and sub-functions.
- ICON editor: Robot program can be edited by ICON operations.
- Easy cursor positioning: Cursor will move to the location touched by a touch panel.
- List for CALL instruction edit: At editing a CALL instruction, a list of sub program is displayed and a sub program can be selected from the list.
- List for MACRO instruction edit: At editing a MACRO instruction, a list of selectable MACRO name is displayed and a MACRO can be selected from the list.
- Current editing line display: Sometimes a current editing line was hidden by a pop up menu and it was hard to check the current edited line. To fix the issue, the current edited line is displayed at the bottom of the screen.
- Sub program display: In program edit screen, a cursor is located at a CALL instruction and press ENTER key, then the sub program is displayed.
- Wide screen mode: Wide screen mode is supported to display some landscape screen with small characters.
- Screen maximization: A split window can be displayed by full screen mode with large characters temporarily and return easily to original display mode.

NOTE

- List for CALL instruction edit, list for MACRO instruction edit and sub program display functions are available on a monochrome screen teach pendant. But other improvements are available on only *i*Pendant.
- ICON menu and ICON editor function are available when *i*Pendant firmware version is 7DA5 (V7.50xxx) or later series. If *i*Pendant firmware is older from 7DA5 then these functions will be disabled.
- ICON editor and easy cursor positioning is available only on *i*Pendant with touch panel optional hardware.
- Design and the kinds of ICONs for ICON menus and ICON editor is application tool specific. In future software version, it might be possible to change the design of ICONs without any pre-notification.

11.2.14.1 ICON menu

During the pop-up menu invoked by pressing MENU, DISP or FCTN key is displayed on a screen, ICON menu is also displayed at the bottom of a screen for operation short cut. Favorite screen selection and user desired window configuration operation can be done quickly to select an ICON in the menu.

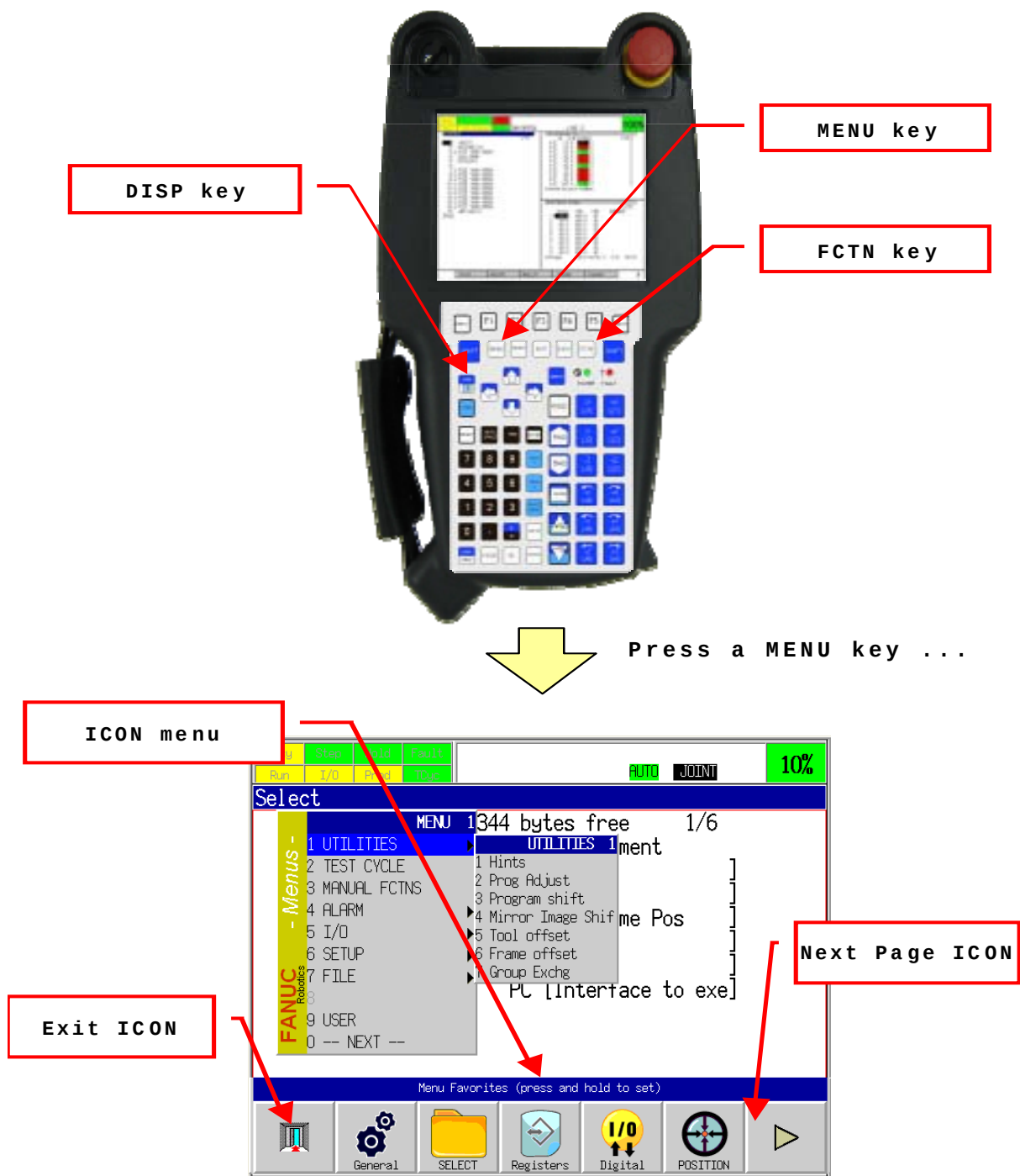


Fig. 11.2.14.1 (a) ICON menu

If your iPendant has a touch panel optional hardware, you can select a ICON by just only screen touching. If your iPendant does not have a touch panel, you can select an ICON by pressing a FUNCTION, PREV, or NEXT key located under a ICON.

The ICON at the right corner of the menu is the next page ICON to change the menu items to next page. The ICON at the left corner of the ICON menu is the exit ICON to escape from the ICON menu.

(A) ICON menu for favorite screen

When a MENU key is pressed, ICON menu for favorite screens will be displayed. By select an ICON of the menu, a favorite screen associated with the ICON will appear.

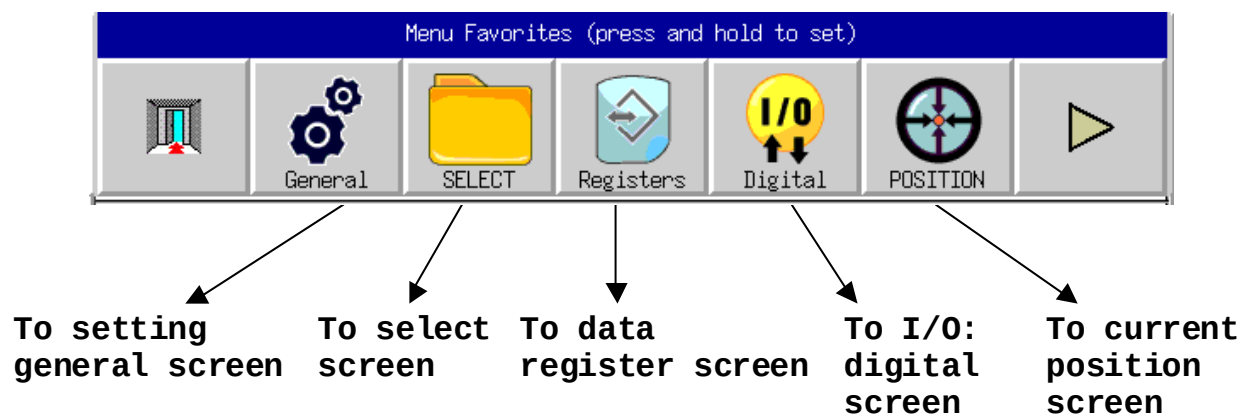


Fig. 11.2.14.1 (b) ICON menu for favorite screen

At initial setting condition, no screen is registered into an ICON menu. To register a short cut to invoke a favorite screen, display your favorite screen and keep to press an ICON of the menu for about 4 seconds. Then the ICON for your favorite screen is registered into the pressed location in the menu. If your iPendant does not have the touch panel optional hardware, please keep to press a function key, PREV or NEXT key for 4 seconds. Up to 10 screens can be registered to your ICON menu.

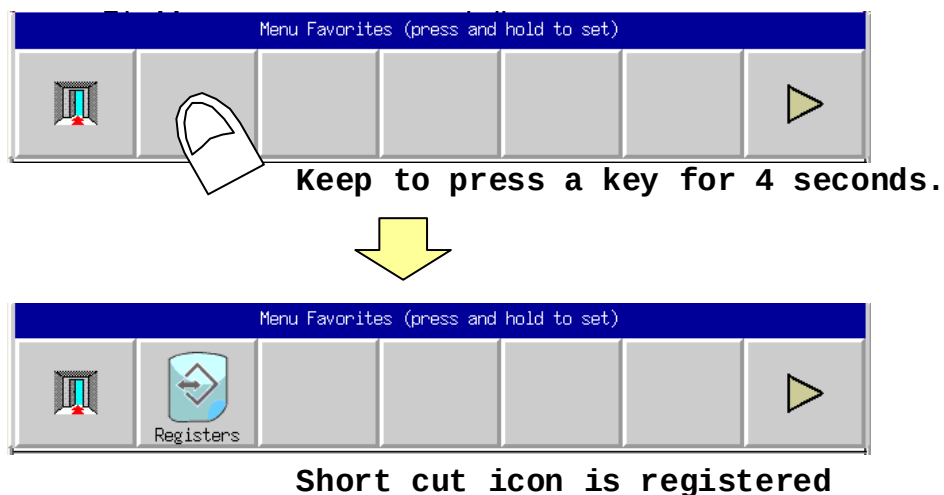


Fig. 11.2.14.1 (c) Registration to ICON menu

In Arc Tool and Spot Tool+, some screens, that are used very often, have been registered into an ICON menu at FANUC shipping as the default setting.

Default setting of ICON menu in SPOT TOOL+



Default setting of ICON menu in ARC TOOL



Fig. 11.2.14.1 (d) Default setting of ICON menu corresponding application tool

(B) ICON menu for User Views

When DISP key and SHIFT key are pressed together, a ICON menu for User Views will be displayed. User Views means the user definable screen configuration and the window split mode and the selected screen in each window can be registered as a short cut ICON in the menu. When a short cut ICON is pressed, window is split and some screens are displayed in the windows according to the screen configuration automatically. Current window configuration is registered by pressing a ICON for about 4seconds. If your *i*Pendant does not have a touch panel, use a FUNCTION key, a PREV key, or a NEXT key. Up to 10 window configurations can be registered.

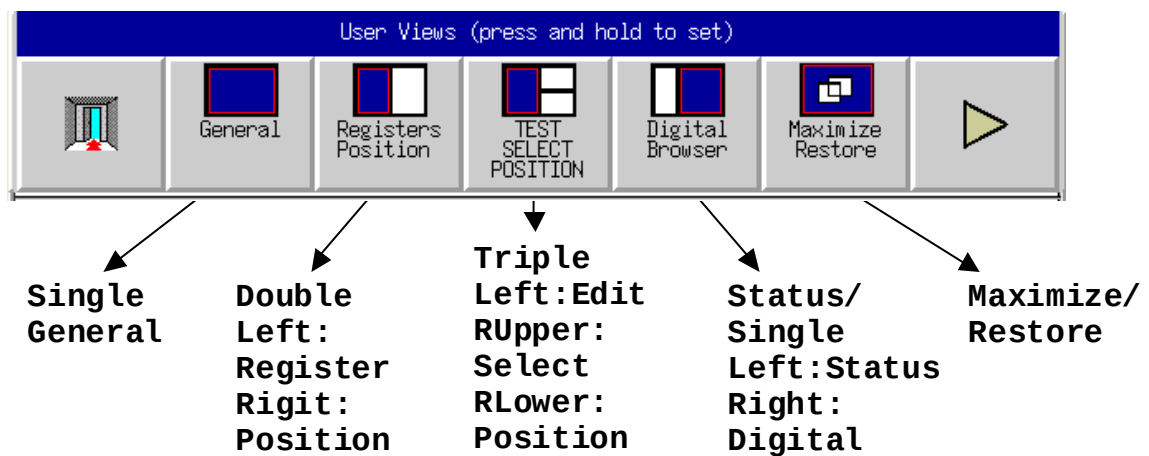


Fig. 11.2.14.1 (e) User View ICON menu

A special ICON is registered as the default setting for window maximization operation at the far right of a ICON menu. When this icon is pressed, current selected window is zoomed and displayed in full screen mode. And the ICON is pressed again, the zoomed window is returned to previous size. If you don't need this maximization operation, you can overwrite this icon for another purpose.

NOTE

If you overwrite the maximization icon but want to recover it, please erase the fifth item of the ICON menu in iPendant setup screen. Then the maximization icon will be recovered at the far right of your ICON menu. The maximization icon is always registered as the far right item and it can not be registered in other menu item location.

(C) ICON menu for sub-function

When a FCTN key is pressed, the ICON menu or sub-function will be displayed. In SPOT TOOL+, short cut icons are registered for some functions, that are used very often for SPOT application.

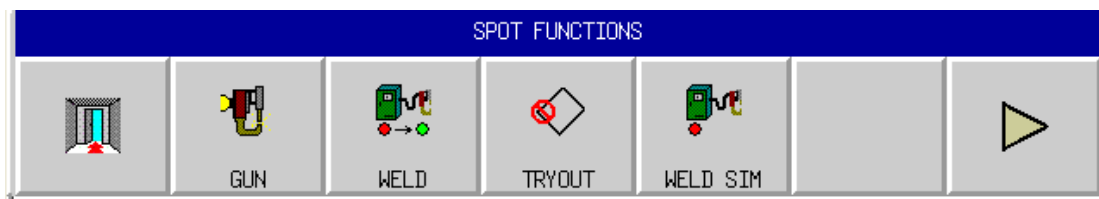


Fig. 11.2.14.1 (f) Sub-functions ICON menu on SPOT TOOL+

NOTE

- All short cut items in ICON menu for sub-function are decided statically and can not be customized.
- ICON menu for sub-function is only available on SPOT TOOL+.

11.2.14.2 ICON editor

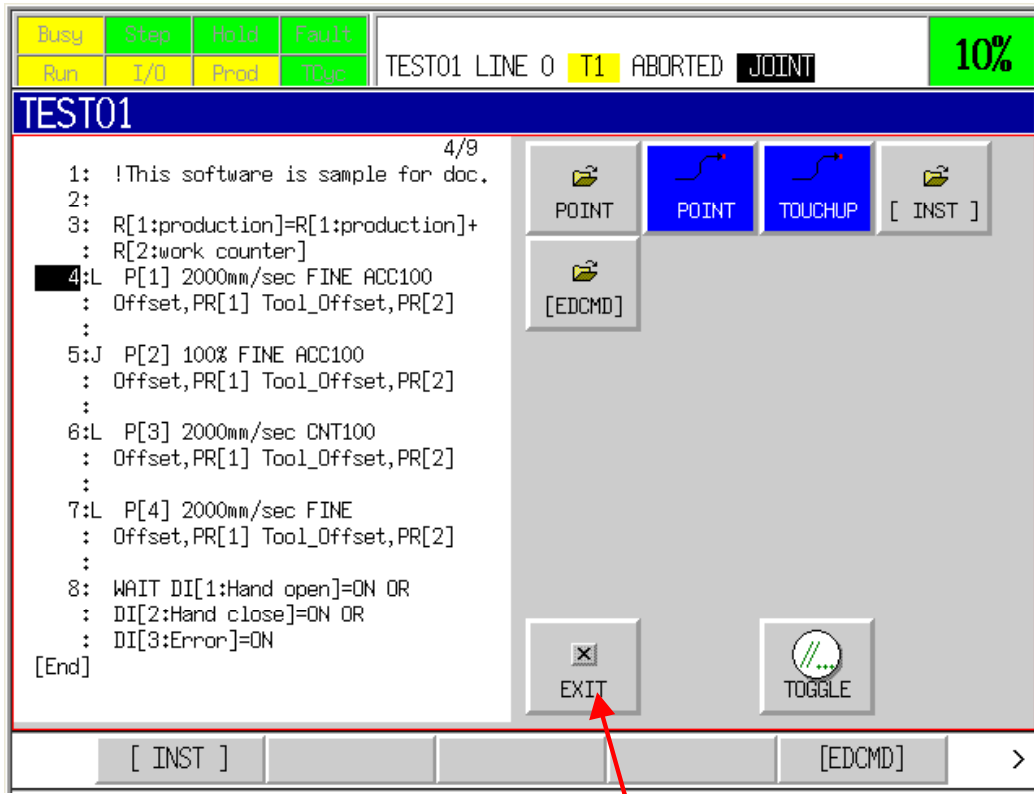
Traditional program editor of robot controller uses pop-up menus to edit instructions of a robot program. From 7DA5 or later series software, new editor using ICON become available.

As the default setting, previous traditional editor is available and new ICON editor is disabled. To use new ICON editor, please enable it by the following procedure in the program edit screen.

1. Press DISP key with SHIFT key to appear DISPLAY menu

2. Move the cursor on the item Related Views.
3. Press RIGHT arrow key then a sub-menu will appear at the right side of the DISP menu.
4. Move the cursor on the item ICON Editor, and press ENTER key.

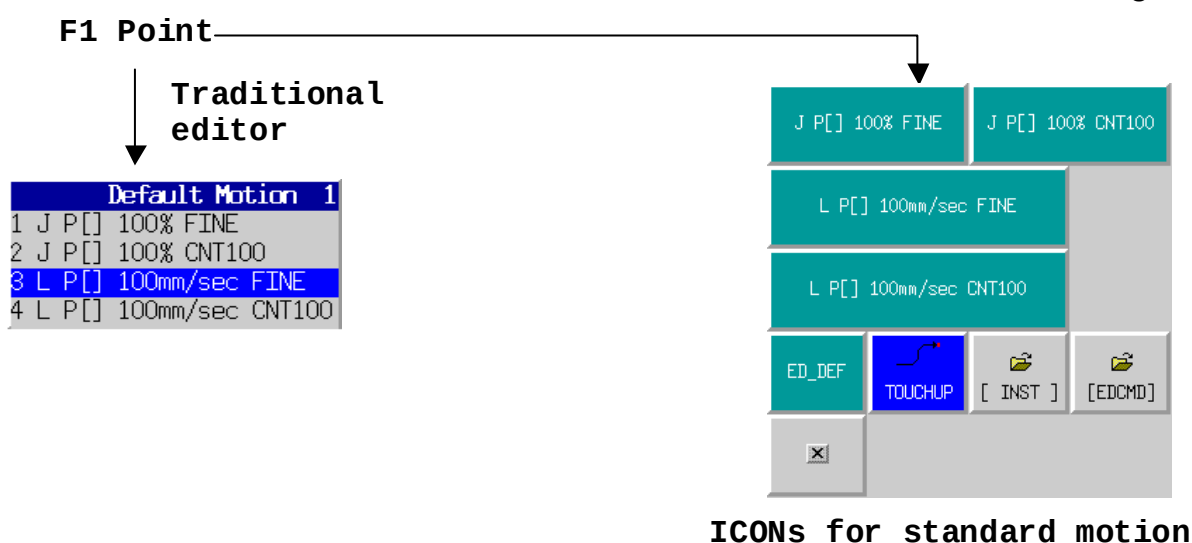
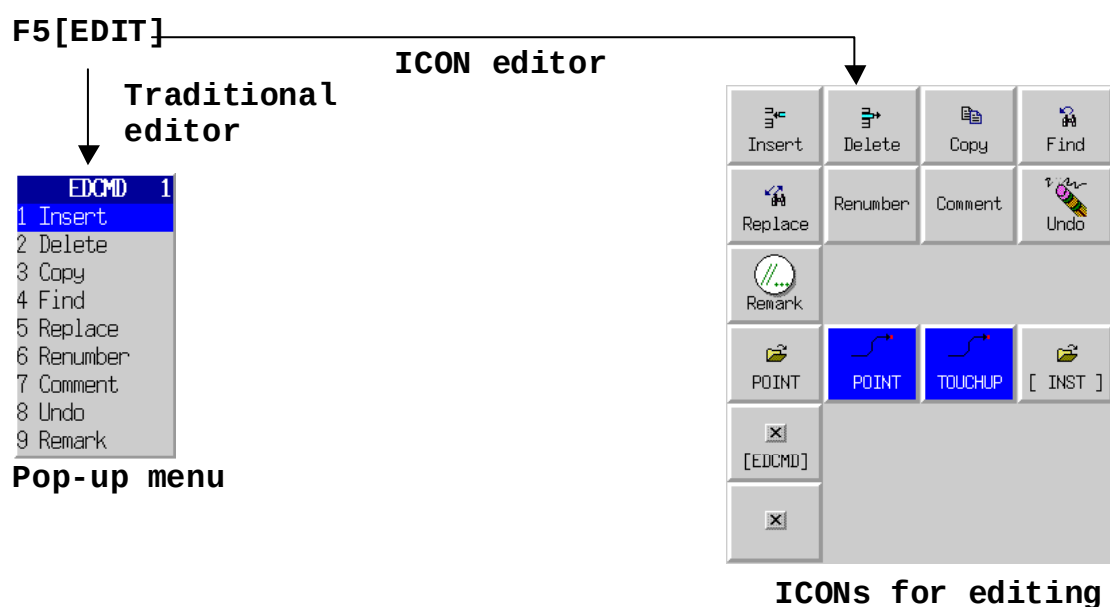
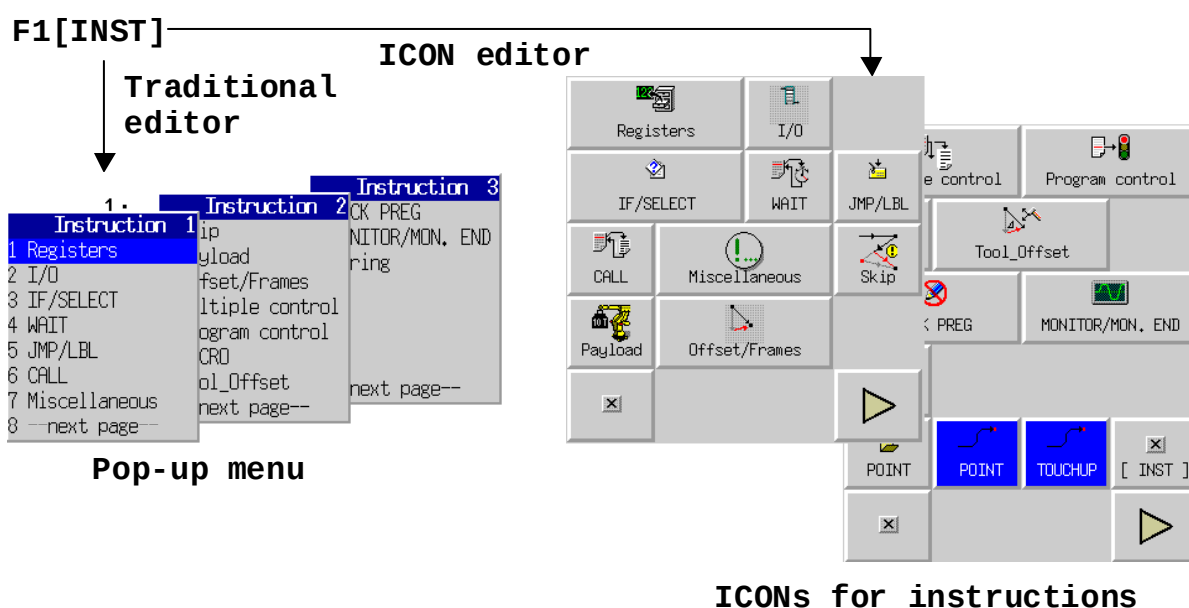
In ICON editor, iPendant screen is split to two windows. At the left window, editing program contents are displayed. At the right window, operational ICONs are displayed. The operational ICONs can be selected by direct finger touch on the touch panel. Therefore the operation from touch panel is more intuitive and more efficient from the previous operations in pop-up menus. ICON editor only supports this window configuration.



This is an ICON to exit from ICON editor and return to the traditional editor.

Fig. 11.2.14.2 ICON editor

ICONS in the right window are the same items displayed in the previous pop-up menu.

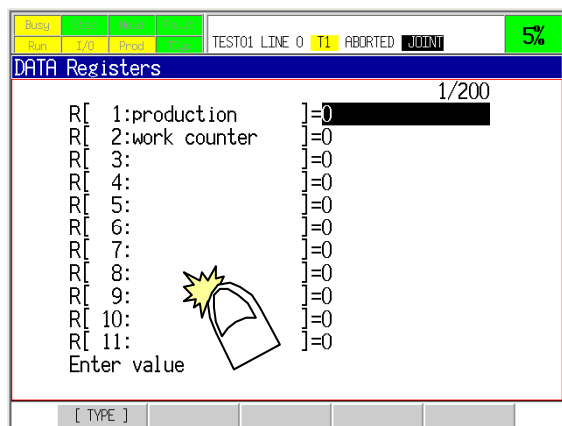


NOTE

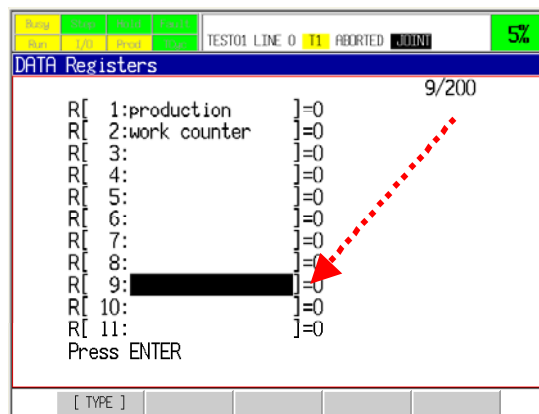
ICON editor can be operational on only iPendant with touch panel option hardware.

11.2.14.3 Cursor positioning

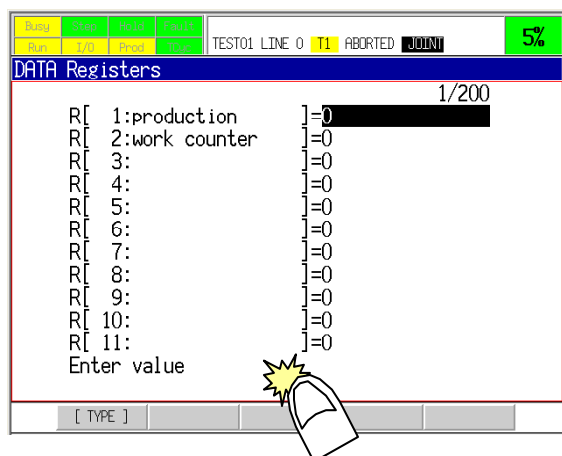
A cursor can be moved by touch on iPendant screen. Menu invoking, menu item selection, and active window selection also can be done by screen touch.



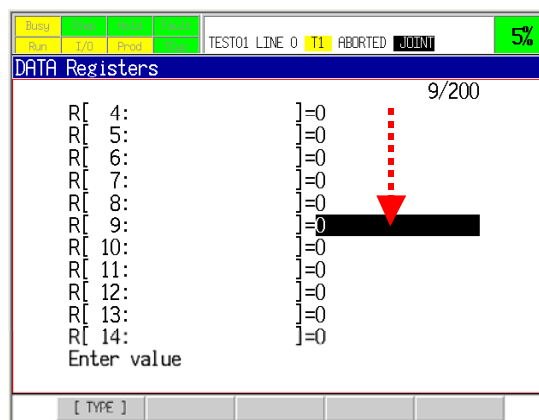
Touch a screen item.



Cursor will move to this item.

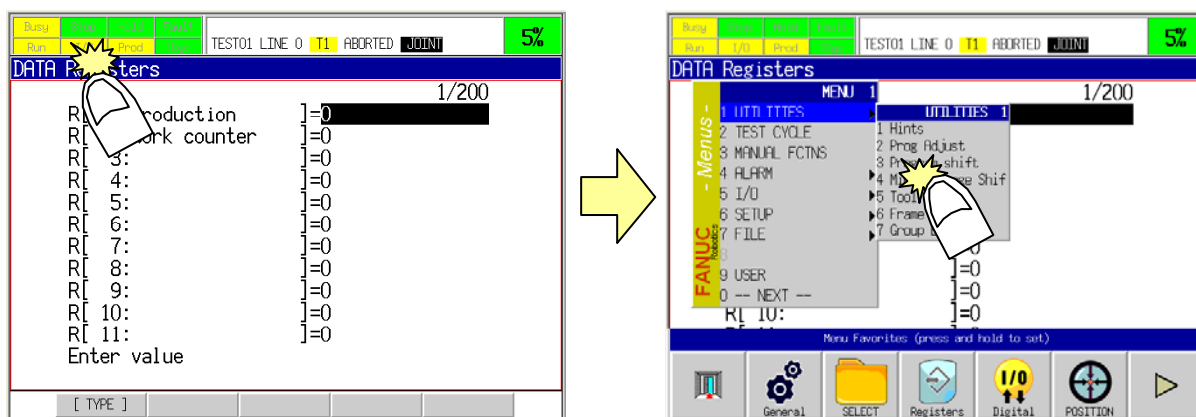


Touch the top or bottom of a screen.



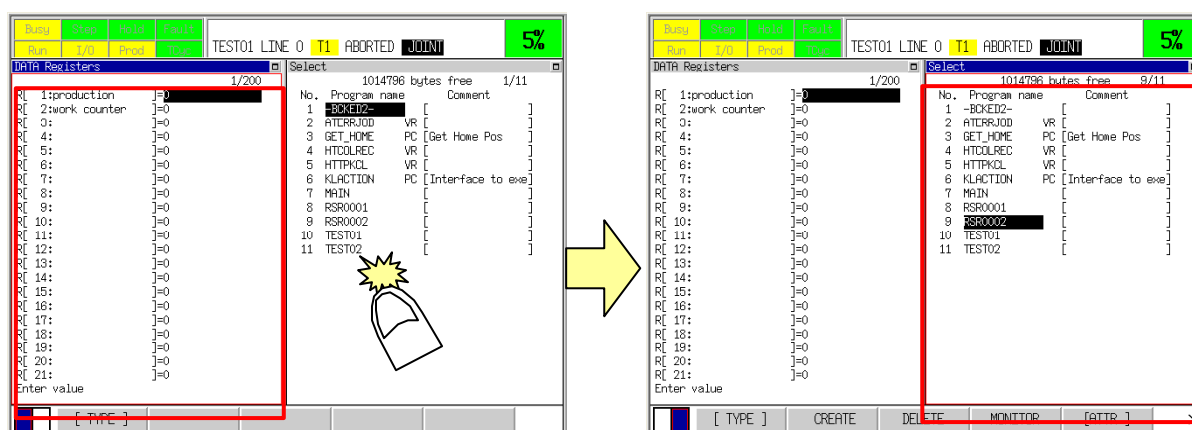
Cursor will move to next or previous page.

Fig. 11.2.14.3 (a) Cursor positioning by screen touch (1)



Touch the left part of a title bar: Open MENU menu.
 Touch the center of a title bar; Open DISP menu.
 Touch the right part of a title bar; Open FCTN menu.

Item of a menu can be selected by touch.



Touch the window

Active window will be changed to the touched side.

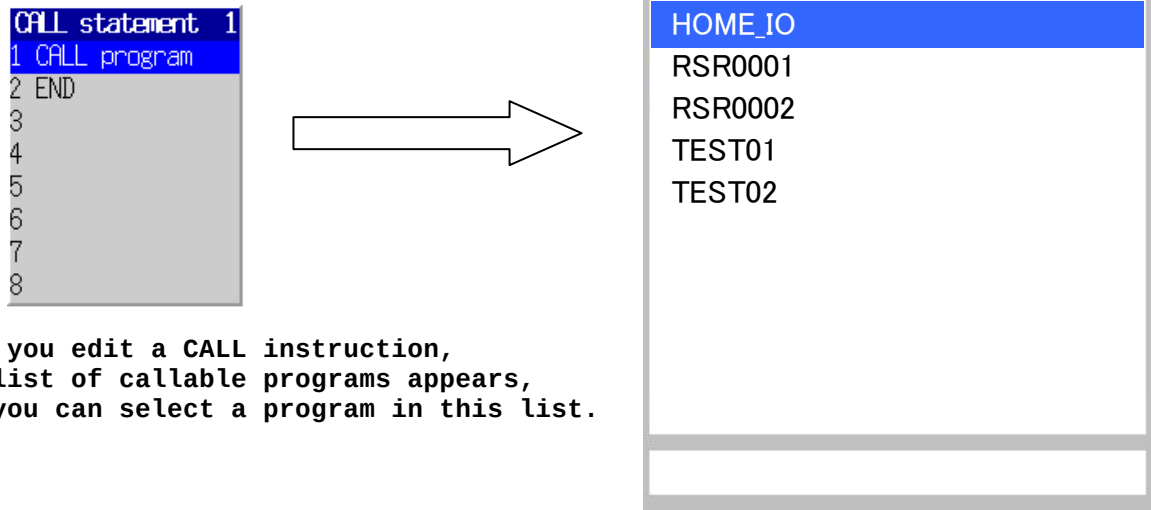
Fig. 11.2.14.3 (b) Cursor positioning by screen touch (2)

NOTE

Touch by a tip of a pen might break the touch panel. Use your finger or the touch pen designed for touch panel operation.

11.2.14.4 List for CALL instruction edit

In edit screen, when you try to edit a CALL instruction, the list of all callable sub program will be displayed and you can select a sub-program from this list.

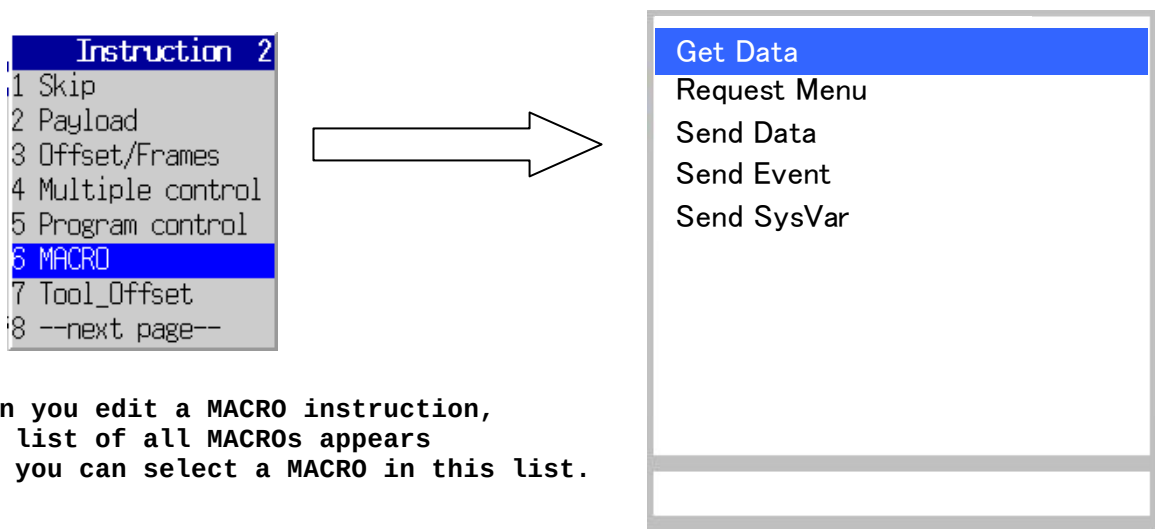


When you edit a CALL instruction,
the list of callable programs appears,
and you can select a program in this list.

Fig. 11.2.14.4 Edit a CALL instruction using a list

11.2.14.5 List for MACRO instruction edit

In edit screen, when you try to edit a MACRO instruction, the list of all MACROs will be displayed and you can select a MACRO program from this list.



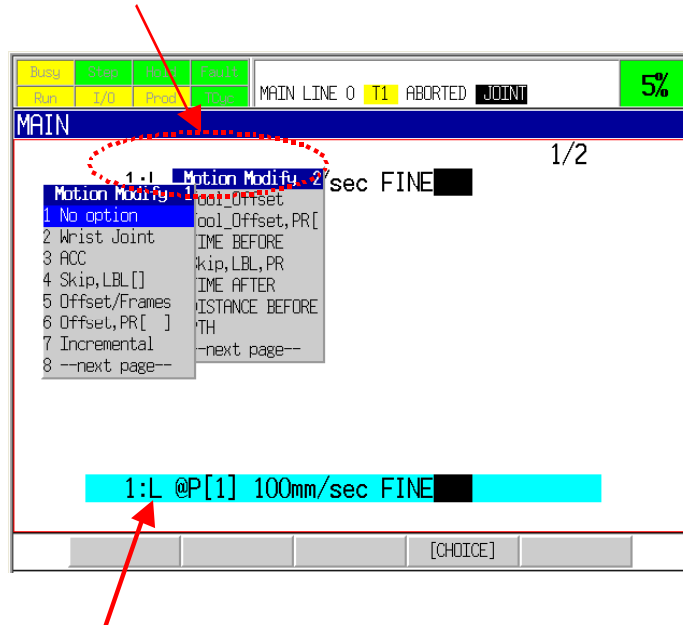
When you edit a MACRO instruction,
the list of all MACROs appears
and you can select a MACRO in this list.

Fig. 11.2.14.5 Edit a MACRO instruction using a list

11.2.14.6 Current edit line display

In edit screen, sometimes current edit line is hide by a pop-up menu for edit operation and a operator can not confirm the contents of a edit line. To improve this inconvenience, the current content of edit line is displayed at the bottom of a screen during edit operation.

The edit line is hide by the pop-up menus.



Current edit line is also displayed at the bottom of the screen.

Fig. 11.2.14 .6 Current edit line display

11.2.14.7 Sub program display

In edit screen, you position a cursor on the sub program name of a CALL instruction and press ENTER key then the sub program contents are displayed. And you press PREV key then the previous program will be displayed again. Up to five times, you can nest the sub program display and return to the previous program.

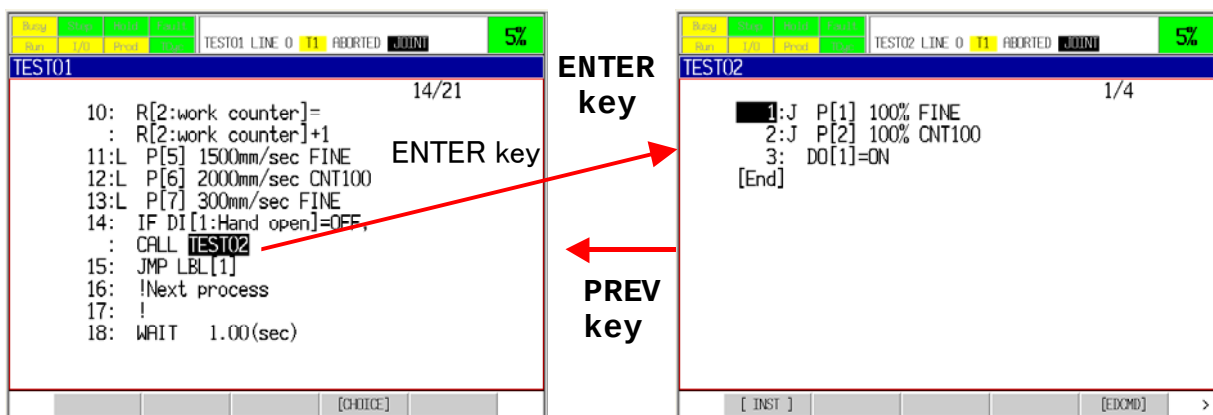


Fig. 11.2.14.7 Sub program display

11.2.14.8 Wide screen mode

New screen mode, 76 characters and 20 lines, is supported to display the wide contents in edit screen and alarm history screen.

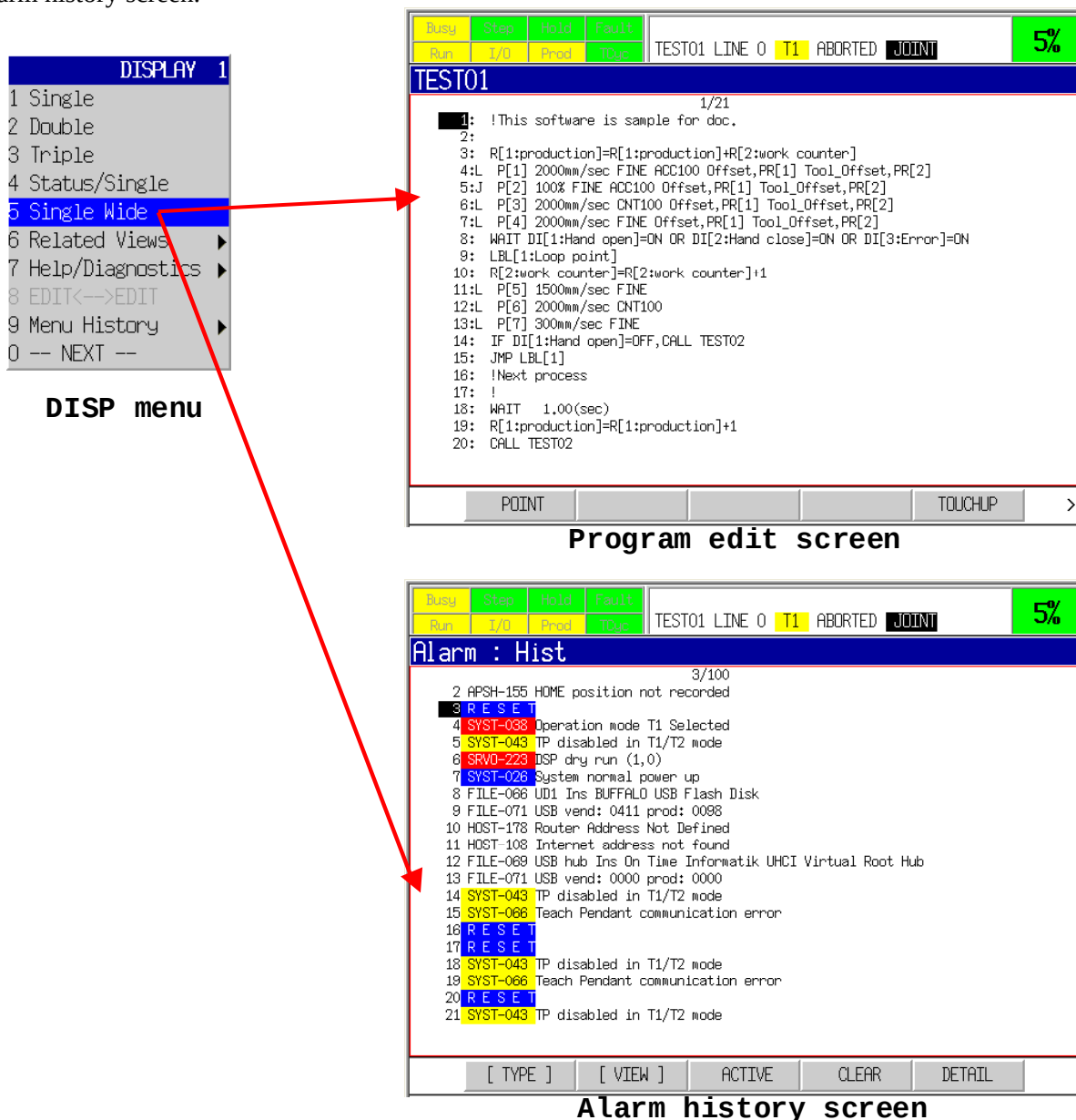


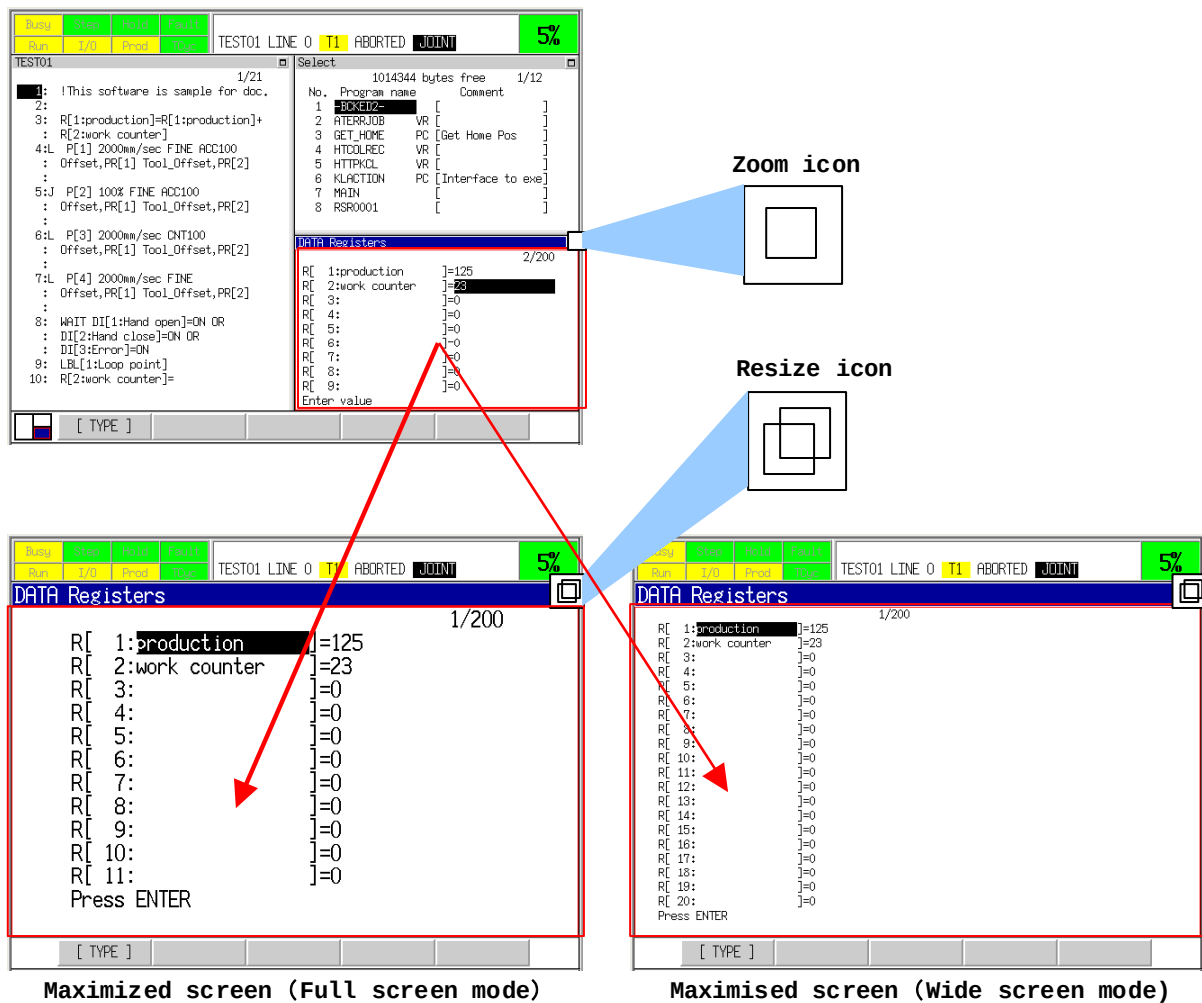
Fig. 11.2.14.8 Wide screen

11.2.14.9 Screen maximization

In double and triple window mode, it might be possible to feel that the screen is not easy to look at because that character font is too small. The screen maximization function can be used to change the current screen to full screen mode temporarily and return to the original screen mode.

There are some methods to maximize and back to original size.

1. Press the zoom icon or the back icon on the right upper of the window by a touch panel.
2. Press the maximization icon and the back icon in ICON menu if these icons are not deleted from your ICON menu.
3. Select the maximization item in DISP menu.

**NOTE**

1. Maximize operation can work in double, triple and single with status screen mode. If current screen mode is single or wide mode and the maximize operation is done but the screen will not be changed.
2. The maximized screen is displayed in the recent of which the single mode or the wide mode.
3. The maximization and resize icons in the window status line is supported from the 7DA5 or later series iPendant firmware. If the firmware is older than 7DA5, these icons do not appear.

11.2.15 Operational improvement in 7DA7 or later series

The following operational improvements are implemented in 7DA7 or later series software of R-30iA/R-30iA Mate.

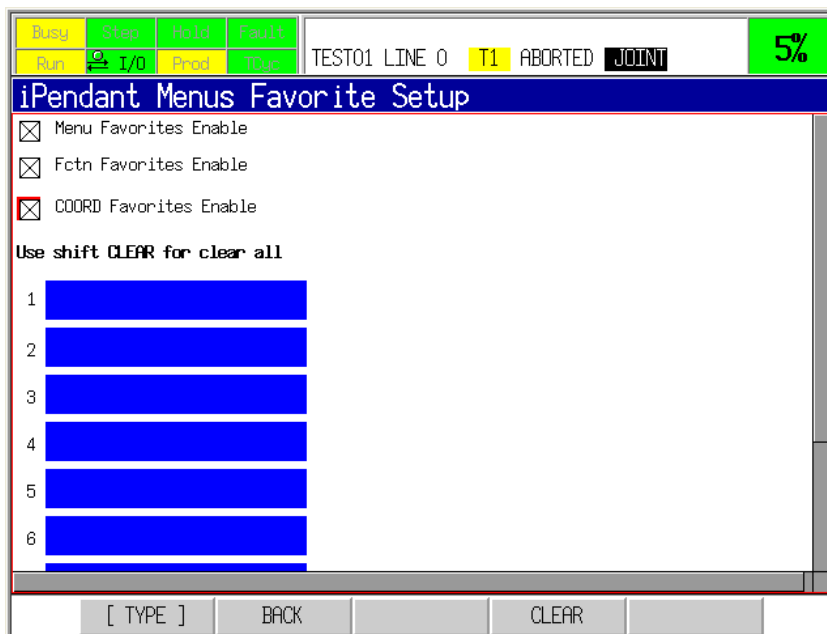
- ICON menu: Some icons are display at the bottom of the screen for coordinate system change.
- Add the check item in iPendant HMI setup screen.

ICON menu of coordinate system

When the SHIFT key and COORD key are pressed together, the ICON menu for coordinate system will be displayed. By select an ICON of menu, quickly change a jog type to the select type.



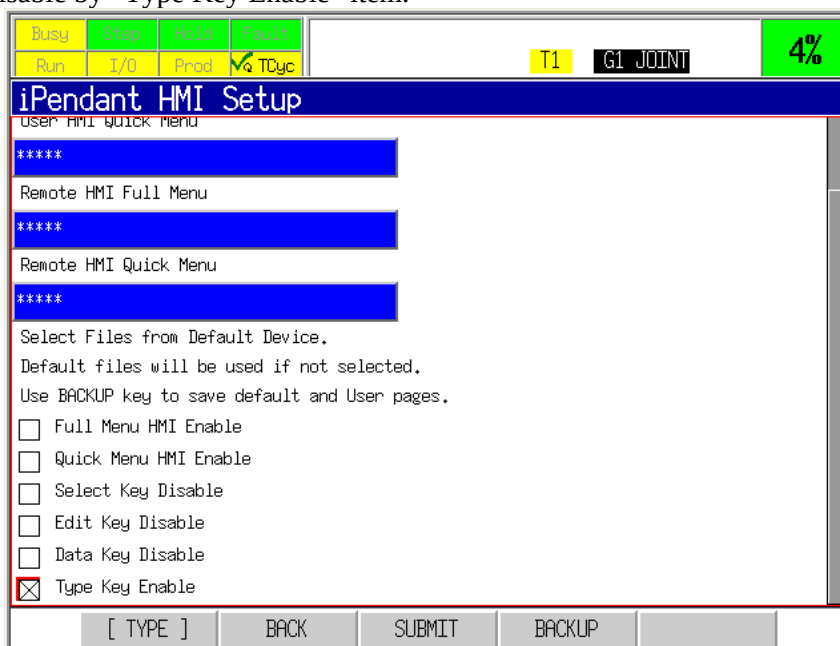
The ICON menu of the coordinate system is registered beforehand and cannot change it. In addition, enable/disable ICON menu can be done in *iPendant* Menus Favorite Setup screen. Select [MENU]->[SETUP]->[iPendant setup], *iPendant* General Setup screen is displayed, press Menu Favorites Setup button. The following screen is displayed.



It is changed Enable/Disable by the check of “COORD Favorites Enable”.

Add the check item in *iPendant* HMI setup screen

Add the check item that is “Type Key Enable” to the end line in *iPendant* HMI Setup screen. Before, it is specification that disable Type key when check “Full/Quick Menu HMI Enable”. Now, it is possible to change Enable/Disable by “Type Key Enable” item.



11.3 SETTING UP iPendant

Overview

It is possible to customize various screens on iPendant.

This section explains how to set up iPendant for customization.

User Views

By using the screen User Views function, it is possible to save up to eight User Views statuses of frequently used user-defined single window or multi-window displays. For example, if an alarm screen, file screen, and I/O screen are usually displayed using a 3-screen window, this set of screens can be defined as screen User Views. When saved, the screen User Views is assigned a name, which is displayed as a User Views item on the display menu, so that User Views can be redisplayed by selecting it later. The User Views described earlier is displayed as Alarm | FILE | Cell I in the User Views list.

To add a set of menus as User Views, use Procedure 11-1. To change a User Views list, use Procedure 11-2.

Procedure 11-1 Adding User Views

Step

- 1 On iPendant, display the set of screens to be added as User Views.
- 2 Press and hold down SHIFT and press the DISP key.
- 3 Select "User Views".

For example, the screen below appears.

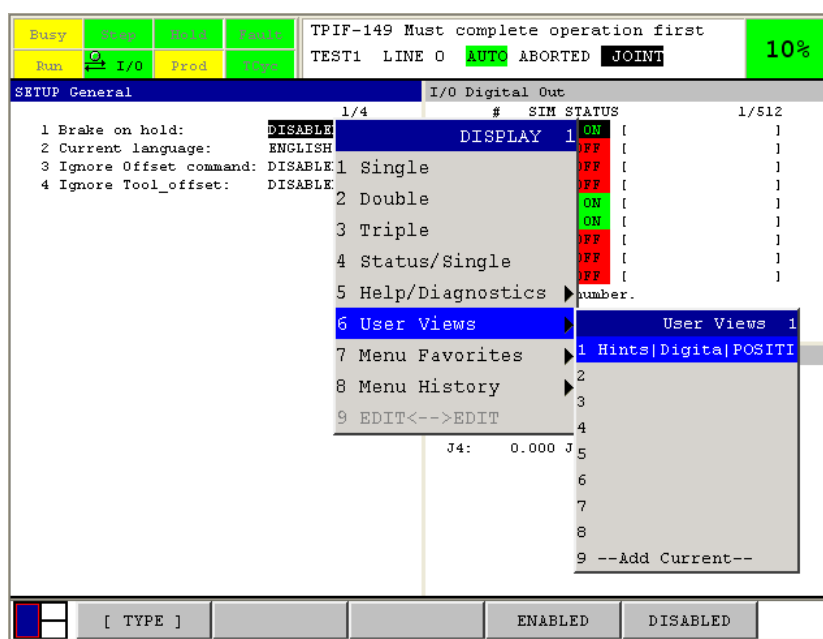


Fig. 11.3 (a) Adding User Views

- 4 Select "Add Current". The currently displayed screen set is added to the list as screen name (for a 1-screen display), screen name|screen name (for a 2-screen display), or screen name|screen name|screen name (for a 3-screen display).
- 5 For each User Views to be added, repeat steps 1 to 4.
- 6 **To display User Views saved in the User Views list**, press and hold down Shift and press the DISP key. Move the cursor to User Views and select the User Views to be displayed from the User Views list.

Procedure 11-2 Changing User Views

Step

- 1 Press MENU.
- 2 Select SETUP.
- 3 Press F1 [TYPE].
- 4 Select *i*Pendant Setup. The *i*Pendant General Setup screen appears.
The screen below appears.

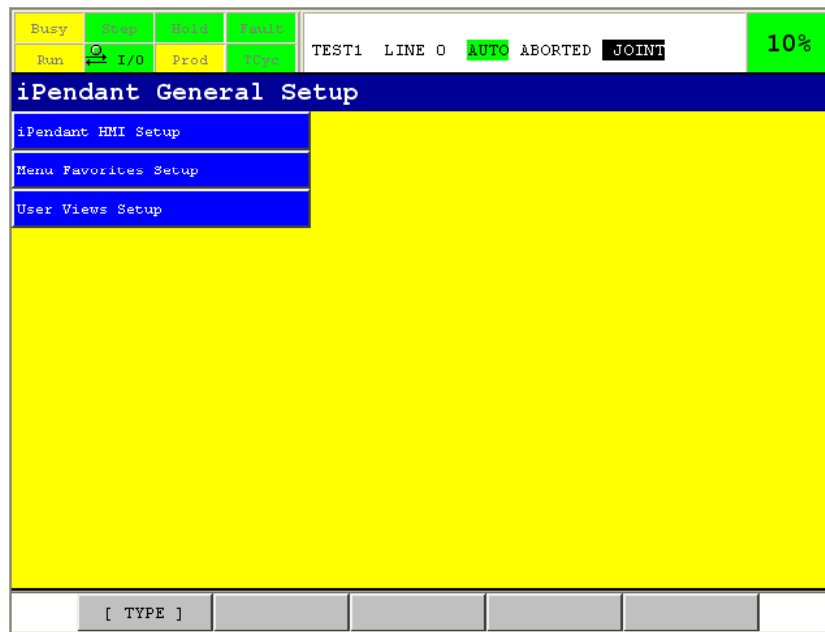


Fig. 11.3 (b) *i*Pendant General Setup screen

- 5 Select *i*Pendant User Views Setup. For example, the screen below appears.

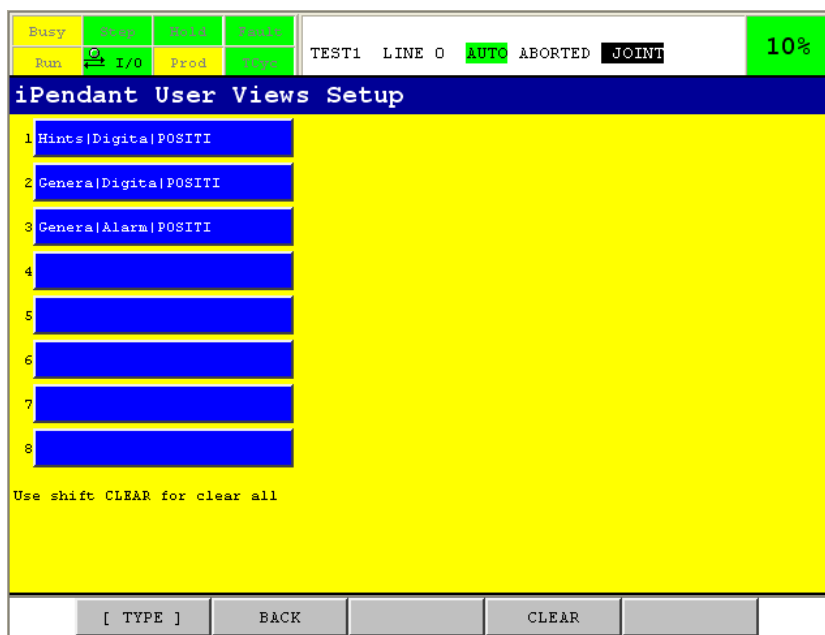


Fig. 11.3 (c) *i*Pendant User Views setup screen

- 6 **To erase one item from the User Views list**, move the cursor to the User Views to be erased and press F4 CLEAR. The User Views list is automatically redisplayed.

- 7 To erase all items from the User Views list, press Shift and F4 CLEAR. The User Views list is automatically erased.
- 8 To display the main iPendant general setup screen, press F2 BACK.

Menu Favorites

By using Menu Favorites, it is possible to generate and save a list of the most frequently displayed menus.

To set a Menu Favorites list, use Procedure 11-3. To change a Menu Favorites list, use Procedure 11-4.

Procedure 11-3 Adding a menu to the Menu Favorites list

Step

- 1 Display the menu to be added as Menu Favorites menu.
- 2 Press and hold down SHIFT and press the DISP key.
- 3 Select Menu Favorites.
- 4 Select "Add Current". The currently displayed menu is added to the list.
See Fig. 11.3 (d).

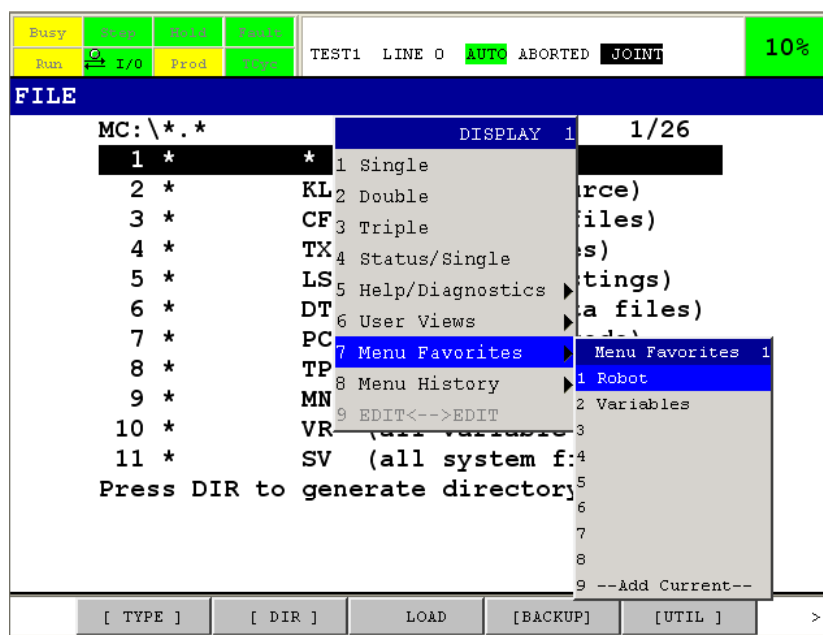


Fig. 11.3 (d) Adding to Menu Favorites

- 5 To display a menu saved in the Menu Favorites list, press and hold down Shift and press the DISP key. Move the cursor to Menu Favorites and select a menu from the list.

Procedure 11-4 Changing Menu Favorites

Step

- 1 Press MENUS.
- 2 Select SETUP.
- 3 Select F1 [TYPE].
- 4 Select iPendant Setup.
- 5 Select iPendant Menus Favorite Setup. The screen below appears.

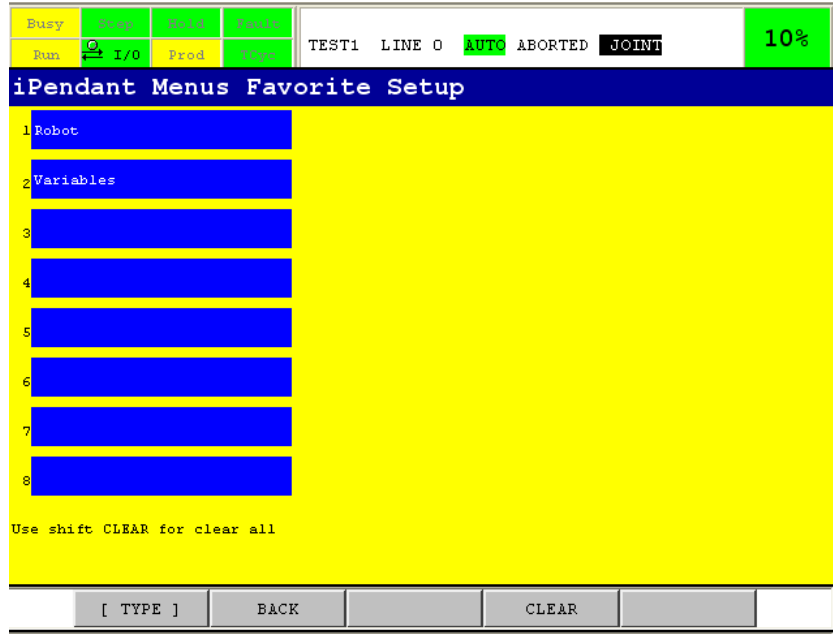


Fig. 11.3 (e) Changing Menu Favorites Setup screen

- 6 **To erase one item from the Menu Favorites list**, move the cursor to the item to be erased and press F4 CLEAR. The Menu Favorites list is automatically redisplayed.
- 7 **To erase all items from the Menu Favorites list**, press Shift and F4 CLEAR. The Menu Favorites list is automatically erased.
- 8 **To display the main iPendant setup screen**, press F2 BACK .

History

The history list records the eight mostly recently displayed menus. This list is automatically generated and cannot be changed. When a name in the list is selected, the screen with that name appears.

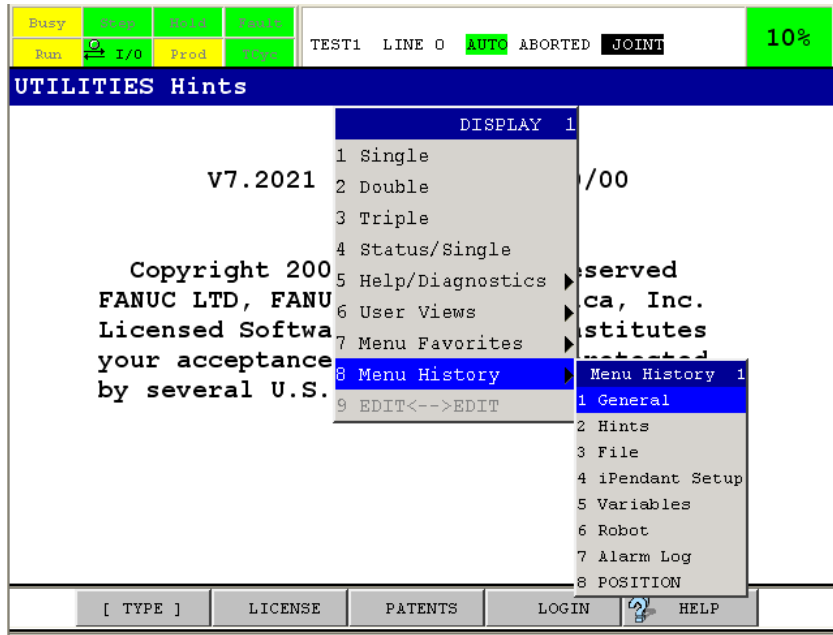


Fig. 11.3 (f) History screen

Setting up the HMI screen

When HMI is used on iPendant, it is possible to customize teach pendant screens according to application. By using the HMI screen customization function, it is possible to set up available menu options so that

specific sets of HTM or STM menus are displayed. It is possible to display a default set of menus and to create a set of .HTM or .STM menus specific to the user and display it.

User menus are saved to FRH:CGTP¥USRHMIQK.HTM and FRH:CGTP¥USRHMIFL.HTM. For remote connection, the default files are FRH:CGTP¥REMHMIQK.HTM and FRH:CGTP¥REMHMIFL.HTM. The default pages, which are created each time the controller is turned on, are FRH:CGTP¥IPHMIQK.HTM and FRH:CGTP¥IPHMIFL.HTM. Usually, operators cannot access FRH:CGTP¥device, so that these files can never be changed or deleted accidentally.

To create a user-defined HMI full menu or quick menu, use Procedure 11-5. To define and set up an HMI full menu, quick menu, and remote menu, use Procedure 11-6.

Procedure 11-5 Creating a user-defined HMI full menu or quick menu

Step

- 1 Press **MENUS**.
- 2 Select **SETUP**
- 3 Press **F1 [TYPE]**.
- 4 Select *i*Pendant Setup. The screen below appears.

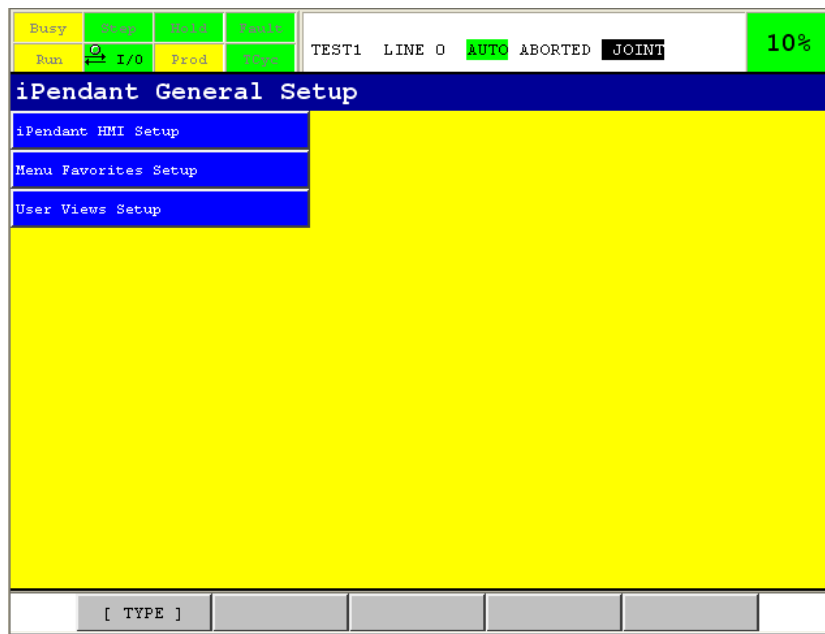
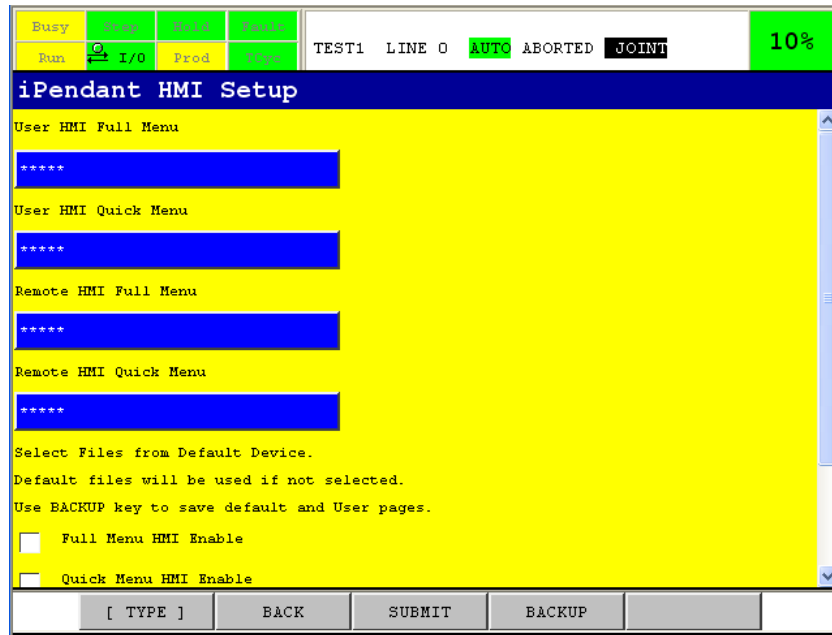


Fig. 11.3 (g) *i*Pendant setup screen

- 5 Select "*i*Pendant HMI Setup". The screen below appears.

Fig. 11.3 (h) *i*Pendant HMI Setup screen

- 6 With the initial settings, the *i*Pendant HMI full menu is saved to IPHMIFL.HTM and the *i*Pendant HMI quick menu is saved to IPHMIQK.HTM. By creating copies of these files and editing them, it is possible to create user-specific full and quick HMI menus. To create copies of the files initially set up, press F4 [BACKUP]. This causes a copy of each file to be saved to a standard device.
- 7 Move the file copies to a personal computer so that they can be edited.

**CAUTION**

The HMI full menu file (IPHMIFL.HTM) must have "iPendant Setup" as a menu item. If passwords can be used on the controller, the HMI quick menu file (IPHMIQK.HTM) must have [Passwords] as a menu item. Keep these menu items if required. If the menu item are missing, created menu files do not operate properly.

- 8 Add or delete menu items to or from these files, as required.
- 9 Return the files to the controller.
- 10 To set up and use the created files, use Procedure 11-6.

Procedure 11-6 Setting up selections on an HMI full menu, quick menu, and remote menu

Step

- 1 Press MENUS.
- 2 Select SETUP.
- 3 Press F1 [TYPE].
- 4 Select *i*Pendant Setup.
A screen similar to the one shown in Fig. 11.3 (g) appears.
- 5 Select "iPendant HMI Setup". A screen similar to the one shown in Fig. 11.3 (h) appears.
- 6 Then, it is possible to select the HTM or STM file to be displayed instead of a full menu, quick menu, remote full menu, and remote quick menu.
 - a To select an HTM or STM file on the default device of the controller, click the blue bar located at the bottom of the menu to be set up. A list of available HTM and STM files appears.
 - b Select the desired file and press "ENTER".
 - c Continue to select files for individual menus ([User HMI Full] , [User HMI Quick] , [Remote HMI Full] , and [Remote HMI Quick]).

- d Press F3 [SUBMIT]. The name displayed on the blue bar is erased to indicate that setup is completed.
- 7 To start a **selected full or quick HMI menu mode or remote full or quick HMI menu mode**, click the square next to the selection located at the bottom of the screen and press F3 [SUBMIT]. Press "F3", and the checkbox disappears to indicate that the selection is made.

To set up an HMI full menu, press the "MENUS" key. The screen below appears.

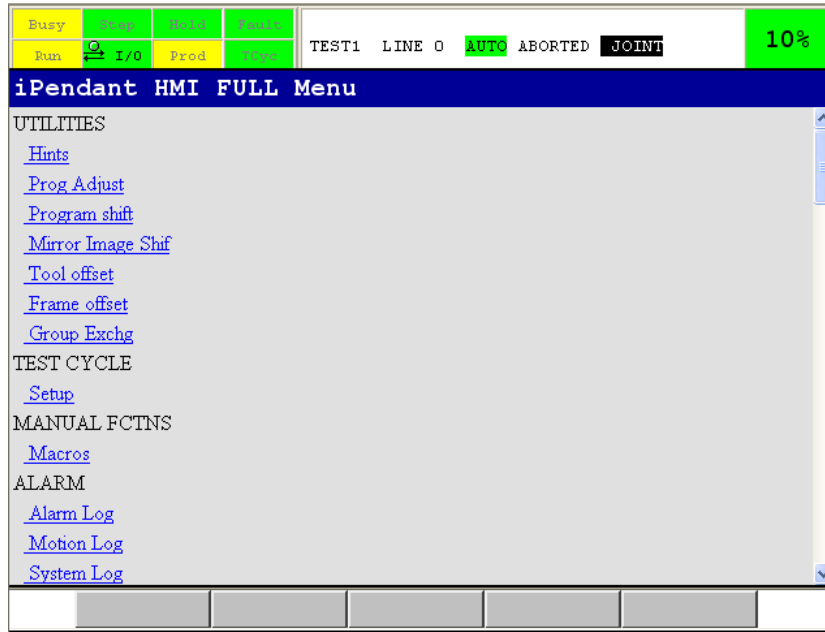


Fig. 11.3 (i) iPendant HMI FULL Menu

To set up an HMI quick menu, press the "FCTN" key and select "QUICK/FULL MENUS". The screen below appears.

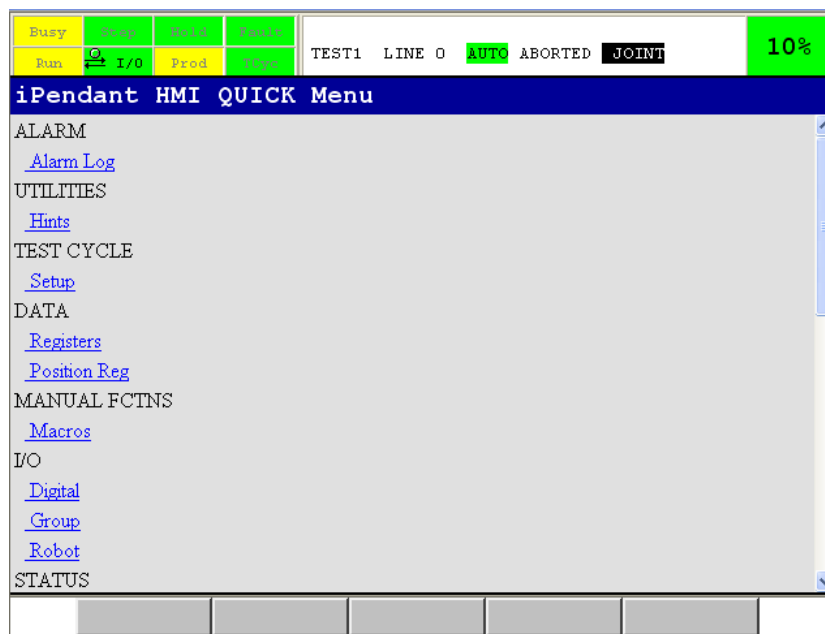


Fig. 11.3 (j) iPendant HMI QUICK Menu

- 8 To disable the "SELECT", "EDIT", or "DATA" key in HMI mode, scroll down and click the square next to the desired key name. For example, the screen below appears.

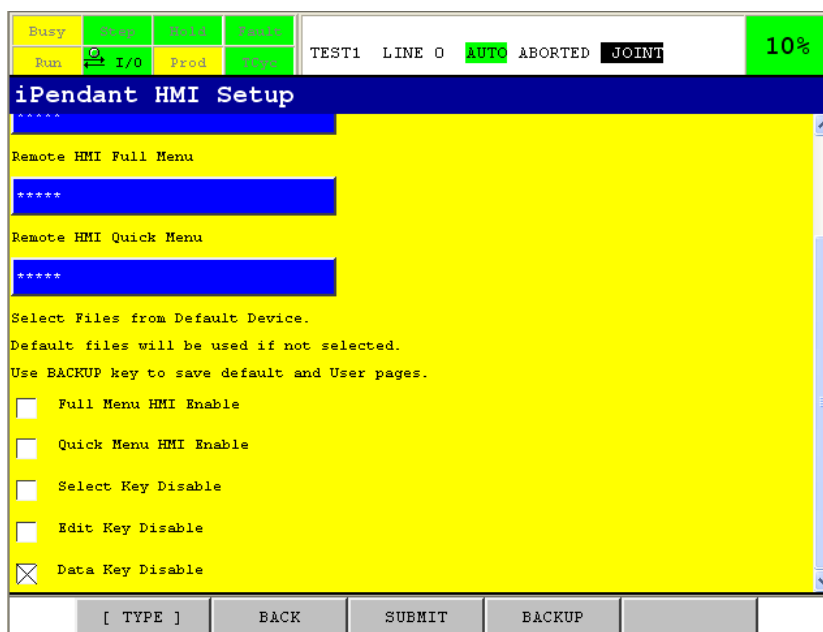


Fig. 11.3 (k) *i*Pendant HMI setup screen, if the "DATA" key is to be disabled

After clicking the desired key, press F3 [SUBMIT]. After [F3] is pressed, the checkbox disappears to indicate that the selection is made.



CAUTION

Each time the HMI setup screen appears, a check mark appears and the current settings are indicated.

- 9 To save all HMI files to the standard device, press F4 [BACKUP]. This causes all files to be copied from FRH:¥CGTP¥ to the standard device in overwrite mode.
- 10 To display the previous menu, press F2 [BACK].

Procedure 11-7 Setting up of 7DA5 related functions

Step

Setting up of new functions supported in 7DA5 or later version software, for example enable/disable ICON menus and delete an item from ICON menus, can be done from *i*Pendant HMI setup screen. Press MENU key, select SETUP, press F1:[TYPE], select 0:NEXT, select *i*Pendant Setup, then *i*Pendant setup screen will appear on a screen. In *i*Pendant screen, three buttons exist. One is to open the "*i*Pendant menu favorite setup" screen to setup the favorite screen ICONs and sub-functions ICONs. Other button is to open the "*i*Pendant user view screen to setup the user view ICONs.

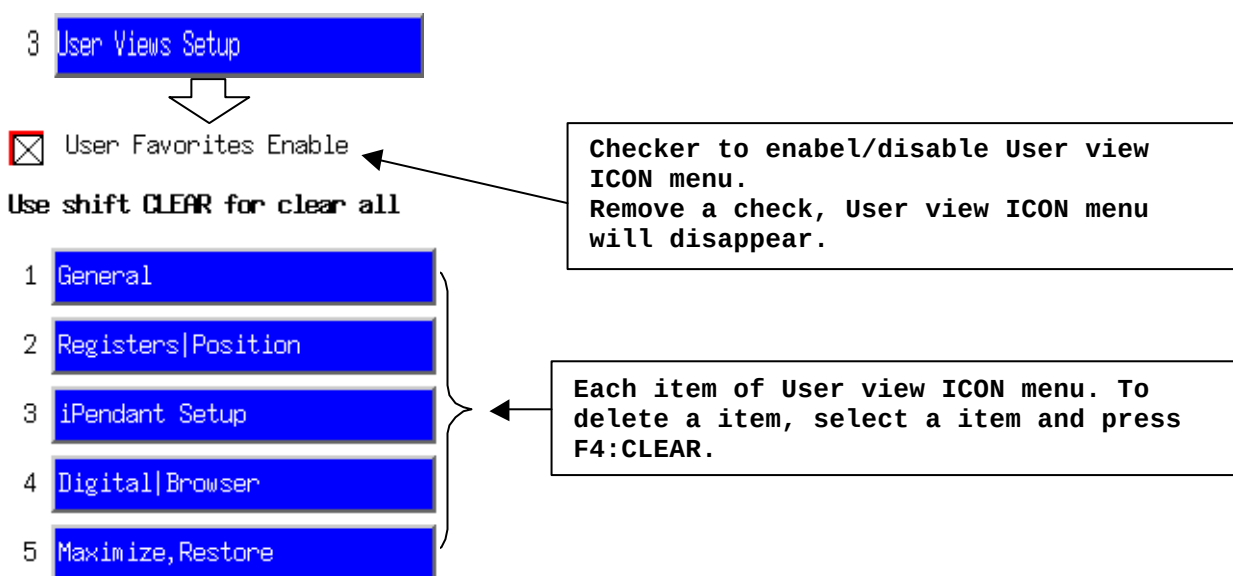
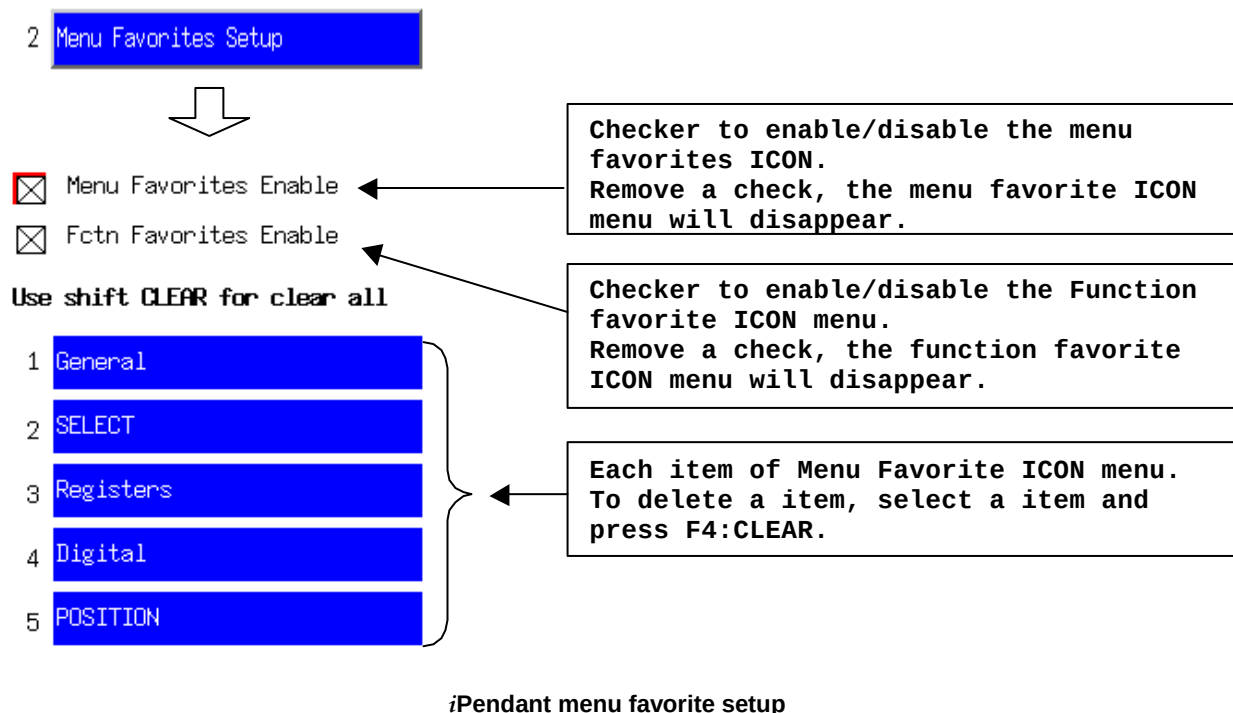


Fig. 11.3(l) iPendant User views setup

iPendant Touch Panel Setup Screen

This is screen to restrict part of operation by touch panel of iPendant.

NOTE

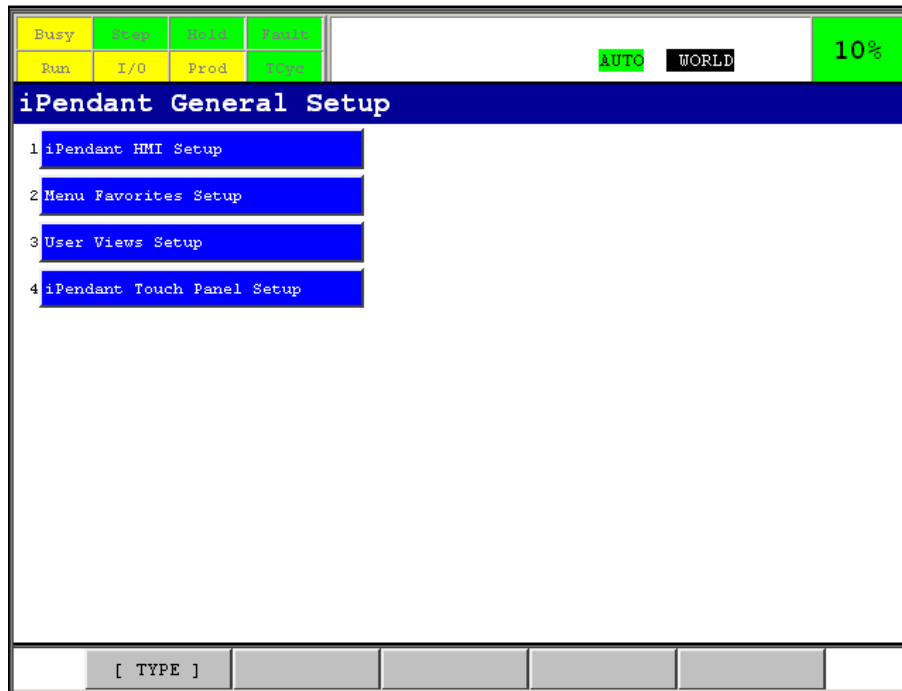
iPendant Touch Panel Setup Screen is supported by system software version of 7DA7/14, 7DA5/16 or later.

Procedure 11-8 Setting of iPendant touch panel

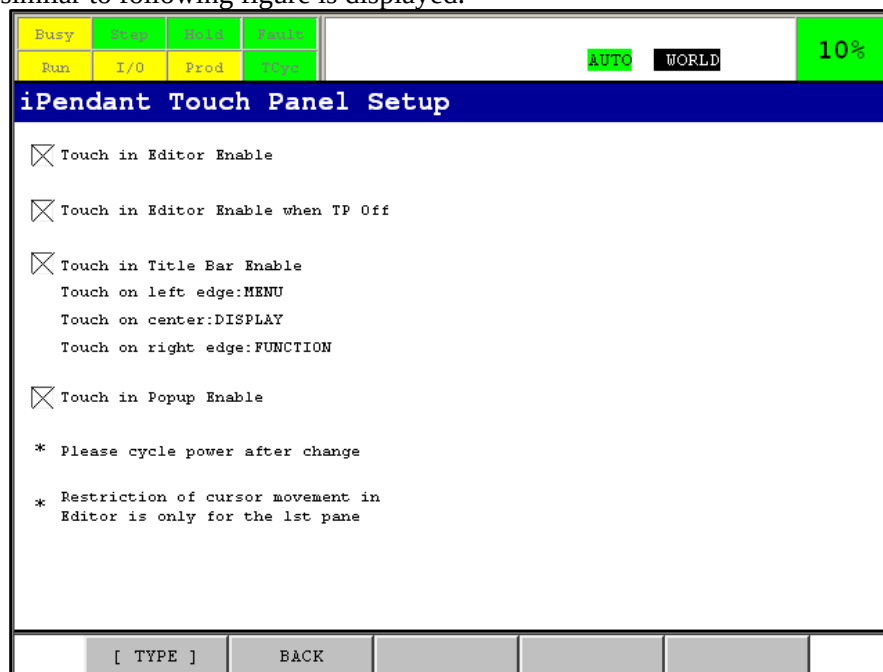
Step

- 1 Press MENUS.
- 2 Select SETUP.
- 3 Press F1 [TYPE].

- 4 Select iPendant Setup.
A screen similar to following figure is displayed.



- 5 Select iPendant Touch Panel Setup.
A screen similar to following figure is displayed.



If check box is checked, corresponding item is enabled. If unchecked, the item is disabled.

Item	Description
Touch in Editor Enable	This is function to move cursor by touch in Program Editor screen.
Touch in Editor Enable when TP Off	This item specifies whether the function of "Touch in Editor Enable" is enabled when TP is Off. If "Touch in Editor Enable" is disabled, this setup item is neglected. In this case, you cannot move cursor when TP is disabled even if this item is checked.
Touch in Title Bar Enable	This is function to display menu when you touch title bar (blue bar which displays title). This item also effects on Maximize/Restore icon on the title bar.
Touch in Popup Enable	When this function is enabled, you can select items in a menu by touch.

NOTE

"Touch in Editor Enable" and "Touch in Editor Enable when TP Off" have effect on only the first pane. If screens are in double pane mode, the items effects only on the left pane. You can move cursor by touch in the right pane regardless of setting of the items.

NOTE

Cycle power is needed for change to take effect. When you change status of check box, "Please cycle power after change" is displayed on error line of TP.

11.4 OPERATION PANEL

11.4.1 Summary

Operation panel function is the software panel for various operation on iPendant.

Operation panel has several lamps indicating the input signal status and several buttons output the signals by manual. The kind and number of signals, the color of lamps indicating ON condition, the label of buttons can be setup. An operation panel has 16 lamps and 16 buttons. UP to four operation panels can be created.

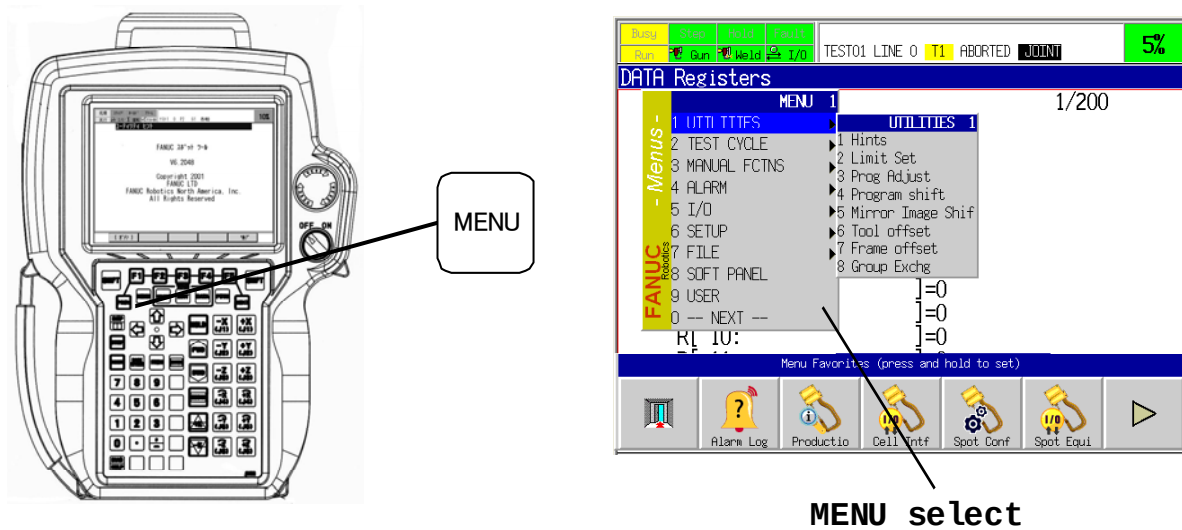
CAUTION:

Do not use operation panel function for a switch impacting human health or equipment damage. Safety switch must be used for a switch impacting human health.

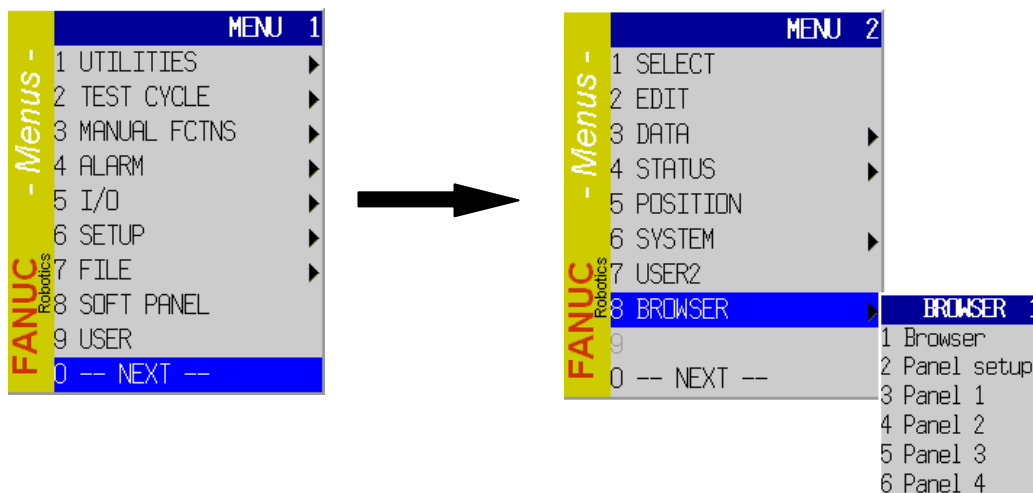
Operation panel function is only available if A05B-2500-R651 Standard setting software option is ordered. This function is not supported with A05B-2500-R650 North America setting.

11.4.2 Basic operation

To display the operation panel screen, press MENU key to display MENU select menu.

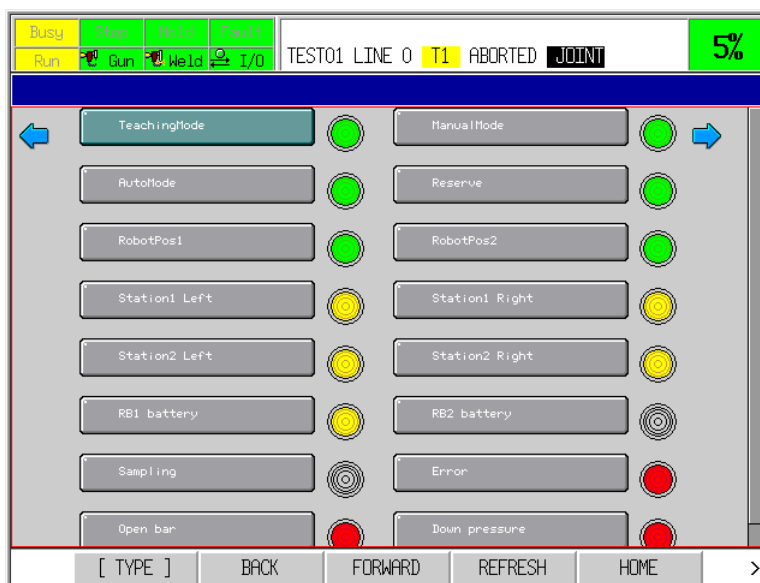


Select 0: NEXT item in MENU select to display next page of the menu, and select BROWSER item. Then the BROWSER sub-menu will appear at the right side of the menu.



Up to four operation panels can be created and the name of each panels are "Panel 1", "Panel 2", "Panel 3" and "Panel 4". Before creating a panel, only the item "Panel setup" exists in the sub-menu. After creating a panel, the created panel will be added into the sub-menu. You select a panel name in the sub-menu, then you can display the selected panel on a screen.

Operation panel has several lamps indicating the input signal status and several buttons output the signals by manual. Additionally panel change buttons are also located.



- Lamp**
 Turn ON or OFF to indicate the input signal status. Input signal of DI, DO, RI, RO, UI or UO can be asserted to a lamp.



- Button**
 By pressing a button, an output signal can be turned ON or OFF alternately by manual. Select a button by using arrow keys on iPendant and press ENTER key, then the output signal assigned to the selected button will be turned ON or OFF. Selected button color is dark green, and no-selected button color is ash.

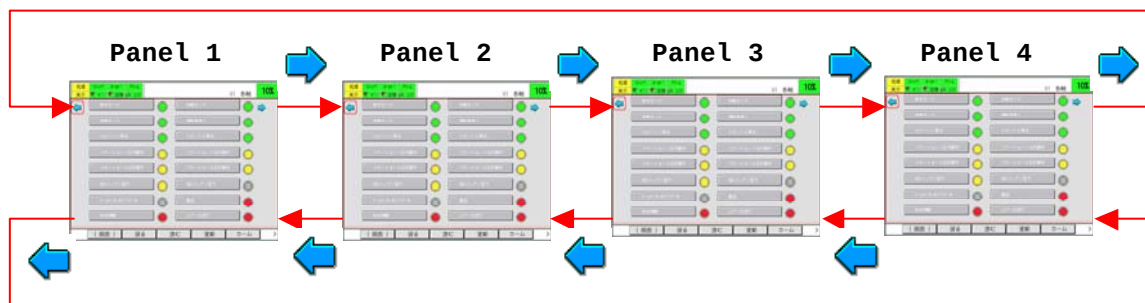
Selected button



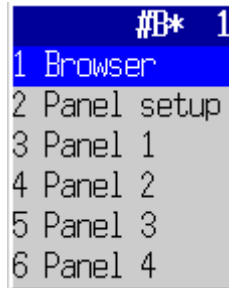
No selected button



- Panel change button**
 At the both side of panel top area, panel change buttons are prepared. Pressing this button, another panel will appear on a screen.



Each panel can be selected directly in the F1:[TYPE] menu.

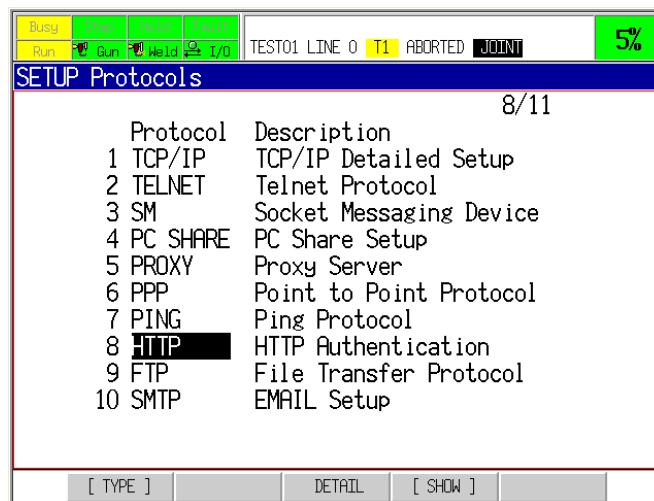


11.4.3 User Authentication

Before using the operation panel function, the following Network setting should be done on your controller.

FANUC robot controller has User Authentication function to protect the injustice access to your controller from external by Network. Operation panel is the content on *i*Pendant Internet Browser. Therefore only authenticated user can operate on a operation panel. If any external injustice access is not possible, you can disable the User Authentication of operation panel by the following setting.

Press MENU key, select SETUP item, select HOST COMM. Then the following SETUP Protocols screen will appear. Move a cursor to the item HTTP and press F3:DETAIL key to display a HTTP setup screen.



Move a cursor to the character "A" at the top of a line and press F3:UNLOCK key. Then the "A" will be changed to "U". Change other "A" to "U" at each line for "*i*Pendant", "KAREL", and "KCL".



If external injustice access is possible, then set a USER name and a PASSWORD for user authentication.

On the HTTP setup, move a cursor to the "Name" and press ENTER key. Then enter a NAME string. The maximum character of NAME is six. Then move a cursor to the "Pwrd" and press ENTER key to enter a password string. The maximum character of password is six. Entered character in the password is not echoed. Same name string and password string are entered on each line of "iPendant", "KAREL", and "KCL". If the first character of each lines of HTTP setup screen is "U", move a cursor to the "U" character and press F4:AUTH key to change it to "A" before entering the name string and the password string. After entering at all lines, please controller cycle power OFF and ON.

At first operation on operation panel, the following screen will be displayed to enter a name and password string for user authentication. Please enter correct name and password and press OK.

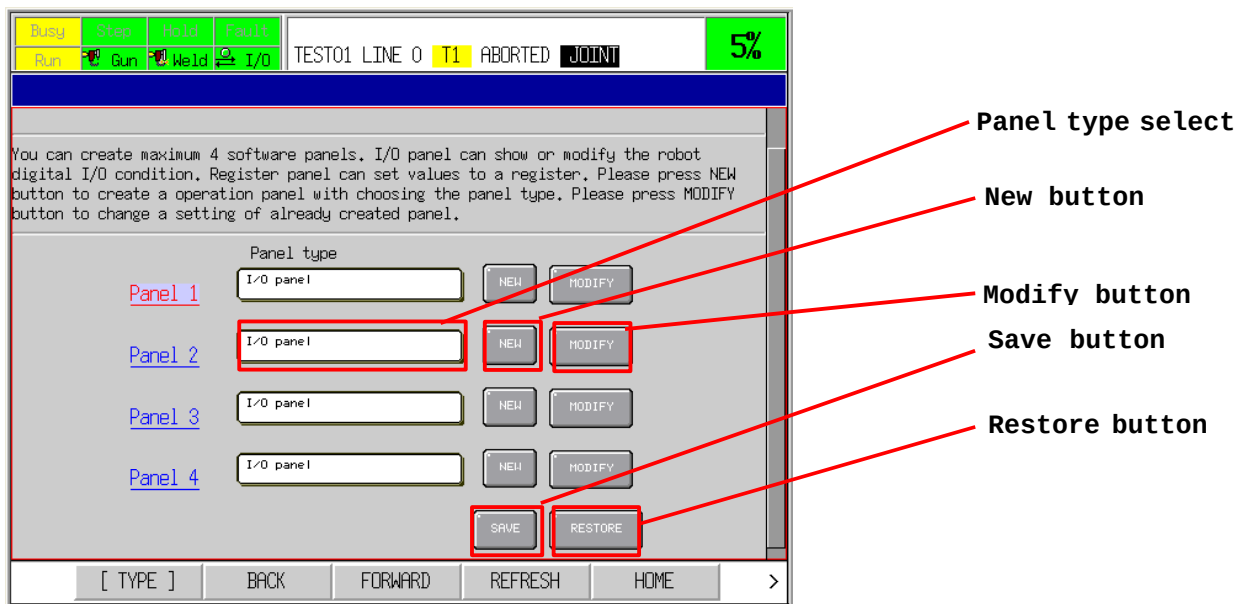


11.4.4 Panel Setup

Before using operation panel, each lamp and button should be assigned to an input and an output signals according to customer system configuration.

11.4.4.1 Operation panel setting screen

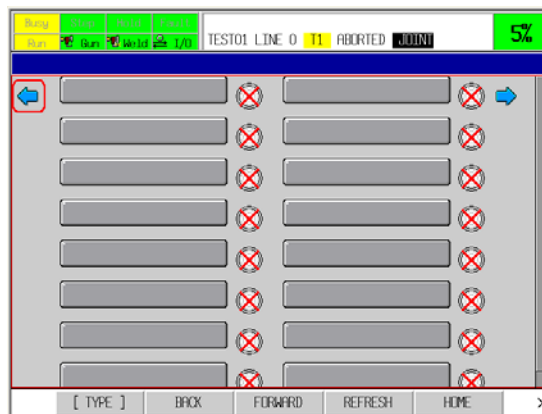
To start setting operation panel screen, select "Panel setup" item from the F1:[TYPE] menu of Internet screen.



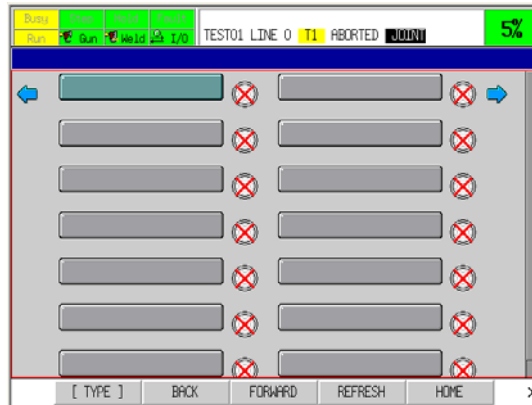
- **Panel type select**
I/O panel, I/O panel (7 lines) and register panel is available.
- **New button**
When you want to create a new panel, at first select a panel type and press this button. When you have created a panel but re-create again as a new panel then select a panel type and press this button also.
- **Modify button**
When you want to change the setting of a created panel, please press this button.
- **Save button**
After panel setting, all setting data can be saved to a memory card by pressing this button.
- **Restore button**
All saved setting data can be restored from a memory card by pressing this button.

As a operation panel, the following three kind of panel are supported.

- **I/O Panel.** This panel has 16 buttons for signal output and 16 lamps for signal input. This panel is useful for the status monitor panel of production execution, and the manual operation panel for external devices.



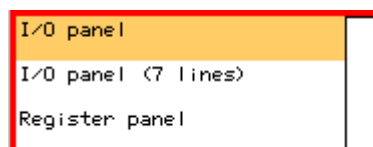
- **I/O Panel (7 lines).** One line is eliminated to expand the space of each lines.



- Register panel. Enter the specified data into a register by pressing a button of this panel. This is useful to select the kind of production workpiece. Entered value will be used for branching the execution in robot programs.



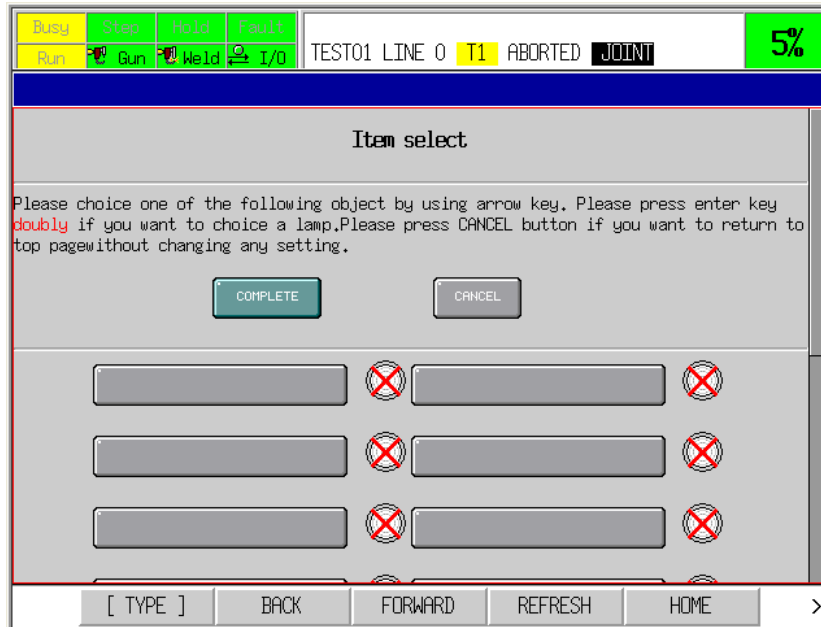
To select a panel type, move a cursor on the field of a panel type on the panel setting screen, and press ENTER key. Then the following selection menu is displayed and you can select a item in this menu by using arrow keys.



11.4.4.2 Item select screen

When you enter a NEW button or a MODIFY button on the panel setting screen, the following item select screen is displayed. From this screen, you need to select a button or a lamp you need to set. Then press ENTER key.

If you want to return to the previous screen without any changing, press CANCEL button. Then any changing is ignored and go to the top page of operation panel.

**NOTE**

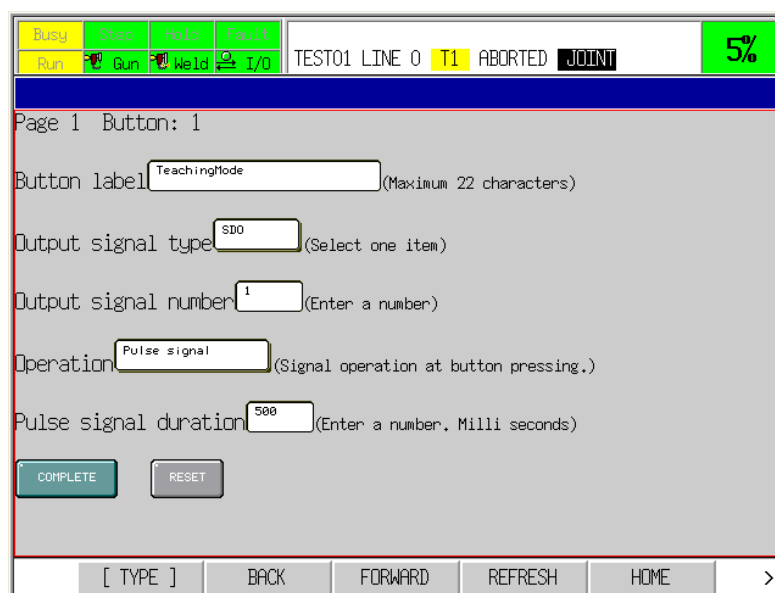
To select a lamp, please press ENTER key twice. When a cursor is over a lamp, the lamp is displayed as follows.



From this condition, press ENTER key, then the cross-shape mark is displayed on the lamp as follows. Press ENTER again from this condition to select the lamp.

**11.4.4.3 Item setting**

When a button is selected, the following button setting screen will be displayed. Move a cursor to an item you need to setup and press ENTER key.



- **Button label**
This is the label string displayed on a button. Up to 22 characters are used as a label.
- **Output signal type**
DO signal or RO signal can be selected. If "No use" is selected as the signal type, only label string is displayed as the string and the button is not displayed.
- **Output signal number**
Specify the output signal number. If 0 is specified, no signal will be output at pressing the button.
- **Operation**
Specify the output signal transition at pressing a button. If ALTER is specified, the output signal will be reversed from the previous status at pressing the button.
- **Pulse signal duration**
If Pulse signal is specified in the operation column, you can specify the duration of pulse signal output. Unit is mili-seconds. When you press a pulse signal button, output signal will be asserted and keep ON for the specified duration, and will be turned OFF automatically.

After setting completion, press COMPLETE button. If you want to cancel all changing done on the screen, press RESET button then all setting values on the setting screen will be returned to previous values.

NOTE about pulse signal

- Precise time pulse signal can not be realized by this function. The duration of signal output can be overdue.
- Maximum setting time of pulse signal duration is 9000 mili-seconds. If a enter value is over from the maximum time, the duration time will be cut to 9000 mili-seconds.
- During signaling pulse signal, other signal output operation can not work.

When a lamp is selected, the following lamp setting screen will be displayed.

- **Lamp color**
RED, AMBER, and GREEN can be selected as the color indicating the signal ON status. If RESERVED is selected, the lamp with cross bar marker will be always displayed. If NO USE is selected, the lamp will not be displayed.
- **Signal type**
Specify the type of a input signal.

- **Signal number**
Specify the number of a input signal.

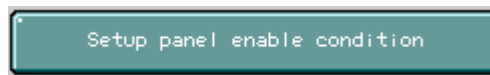
After setting completion, press COMPLETE button. If you want to cancel all changing done on the screen, press RESET button then all setting values on the setting screen will be returned to previous values.

NOTE

String or numeric input data should be entered into an amber item. On the amber item, press ENTER key, then software keyboard will be displayed to enter a string or numeric data. The orange color item has a selectable list. On the orange item, press ENTER key, then a selectable item list will be displayed and you can select a item from this list.

11.4.4.4 Panel enable condition setting

You can use an input signal or some conditions to enable the operation from a operation panel. The following button exists at the bottom of the item select screen.



Press this button then you will get the following screen to setup the panel enable condition.

The followings are supported conditions for enabling the operation from a operation panel.

- **DI, DO, RI, RO**
When a specified input signal is ON, the operation from a panel is enabled. If you need to use this condition, enter a signal number into the index number item. If you enter 0 into the item, this condition is disabled.
- **TP enable**
When a TP enable switch is ENABLE, the panel operation is enabled.
- **Deadman Switch**
When a deadman switch is grasped, the panel operation is enabled.
- **Deadman SW & TP enable**
When a TP enable switch is ENABLE and a deadman switch is grasped, the panel operation is enabled.

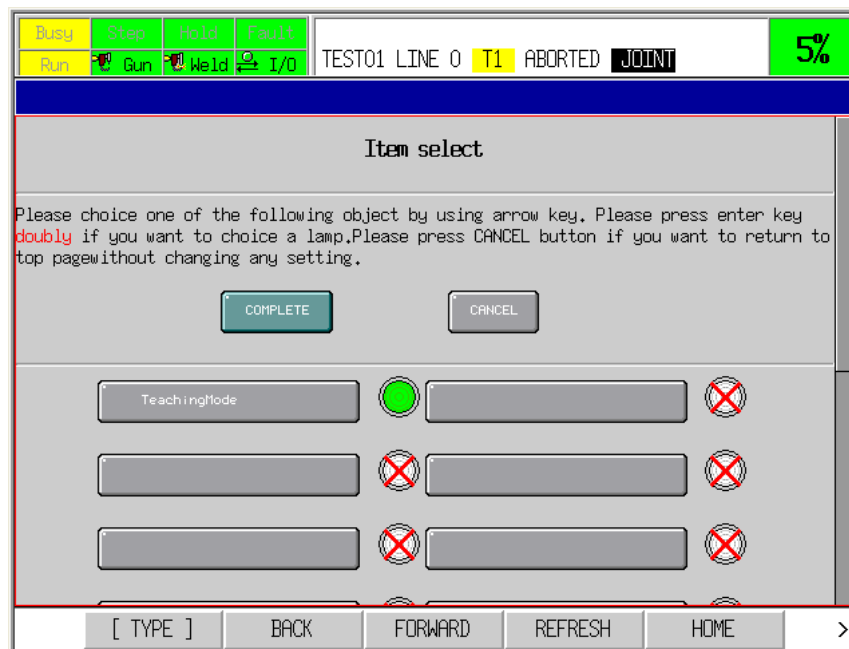
- No condition
Panel enable condition is not specified and always the panel operation is enabled

NOTE Panel enable condition

- Lamps allocated on a panel are independent with the panel enable condition setting and always functional.
- Only one condition is usable for each panel.
- When press a button in disable condition, the button pressing action will occur but in this case, signal will not output. For confirmation of signal output, we recommend to prepare a lamp to confirm signal output and at button operation, operator should check the lamp.

11.4.4.5 Finish setting

On item setting screens, press COMPLETE button, then the item select screen will be displayed. On this screen, you can confirm the setting of lamps and buttons. If you need to continue the setting, you will select another lamp or button.

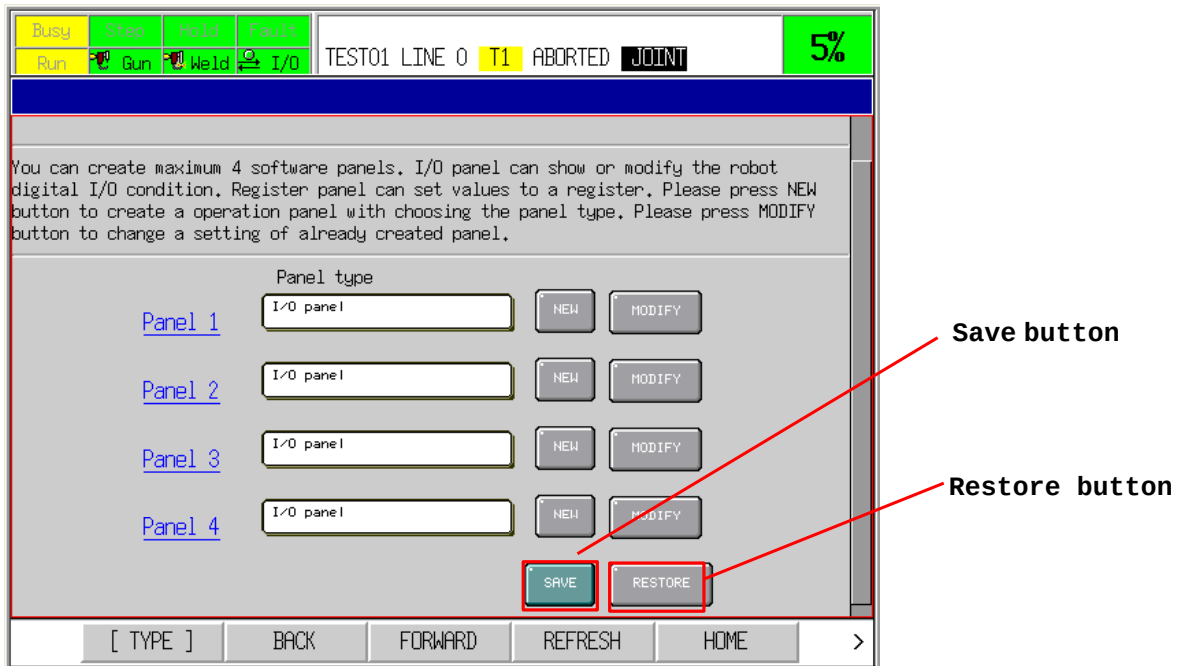


When all setting is finished, press COMPLETE button on the item select screen. All setting values are stored into your controller non-volatile memory and return to the top page of operation panel function.

If you want to cancel all setting, press CANCEL button on the item select screen. All setting value are abandoned and return to the top page of operation panel function.

11.4.4.6 Save and restore setting value

Setting data of all operation panels can be stored to a memory card by pressing a SAVE button on the top page of operation panel function. We recommend saving the setting data continually.



The following files are stored to a memory card when the SAVE button is pressed.

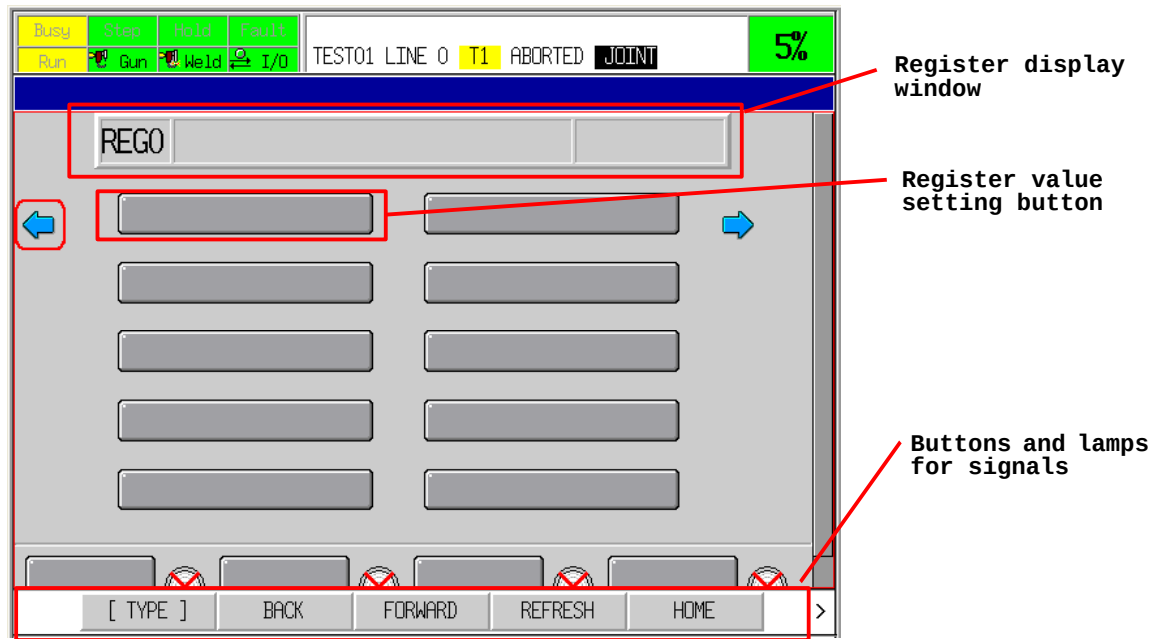
- panel1.stm HTML file for a panel 1
- panel2.stm HTML file for a panel 2
- panel3.stm HTML file for a panel 3
- panel4.stm HTML file for a panel 4
- panel1.dt Setting data for a panel 1
- panel2.dt Setting data for a panel 2
- panel3.dt Setting data for a panel 3
- panel4.dt Setting data for a panel 4

These saved files can be restored when the RESTORE button is pressed.

The above files can be stored by the all file store on FILE screen.

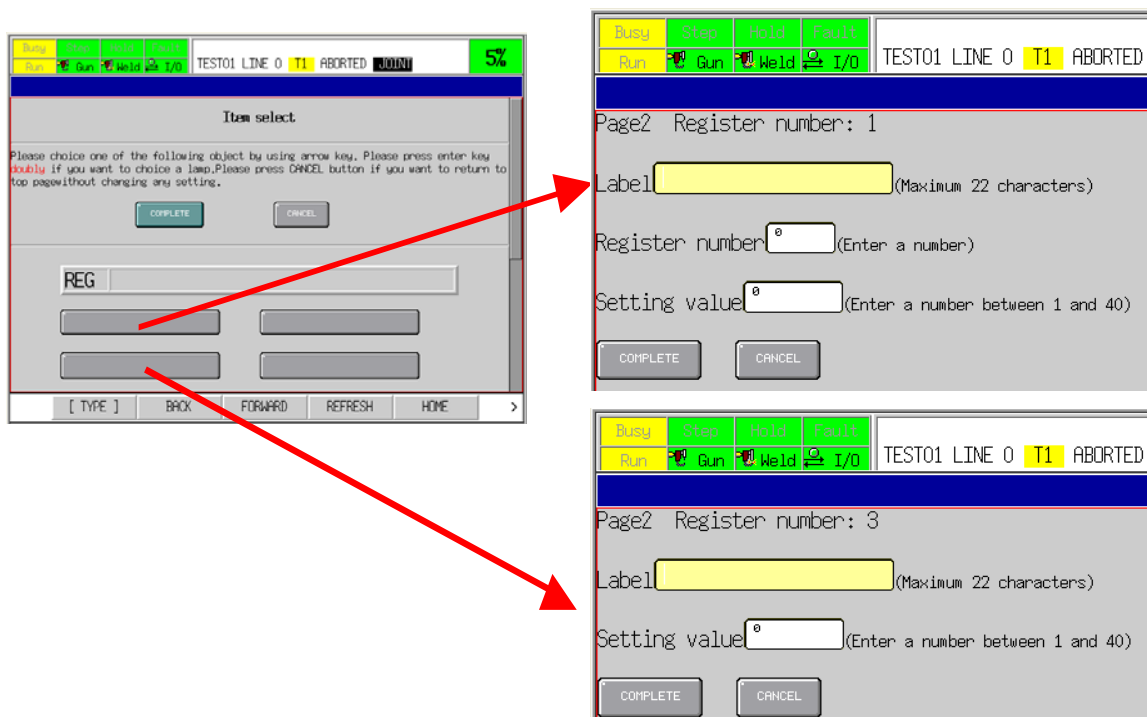
11.4.4.7 Register panel

Register panel can set a specified value to a register when a button is pressed. One register panel can set a value to just only one register. Register panel also has a window for displaying the current value of the register. Register panel also has some buttons and lamps for signal monitoring and setting. But the button is smaller than the I/O panel and maximum character of a button label is 10.

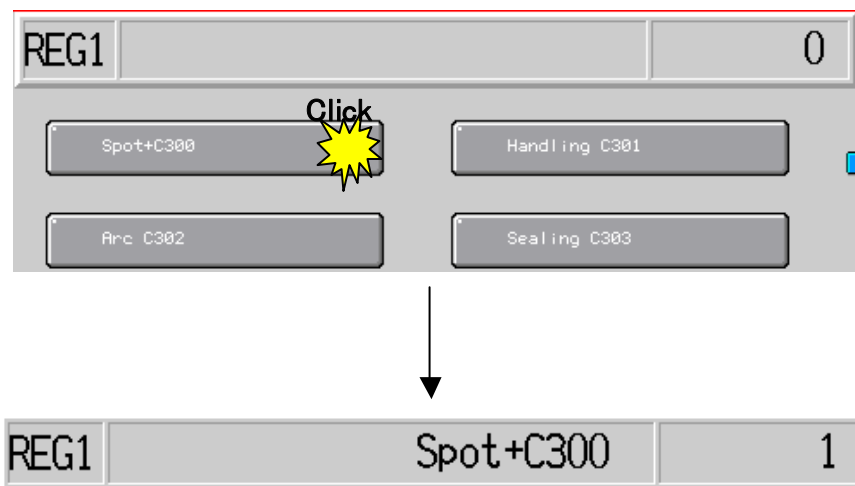


Register panel can be created by the same manner as a I/O Panel. Choose register panel as the type of a panel on the top page of the setting screen.

The label string and the setting value, from 1 to 40, should be specified for each buttons. The register number can be specified in the setting screen only for the upper left button.

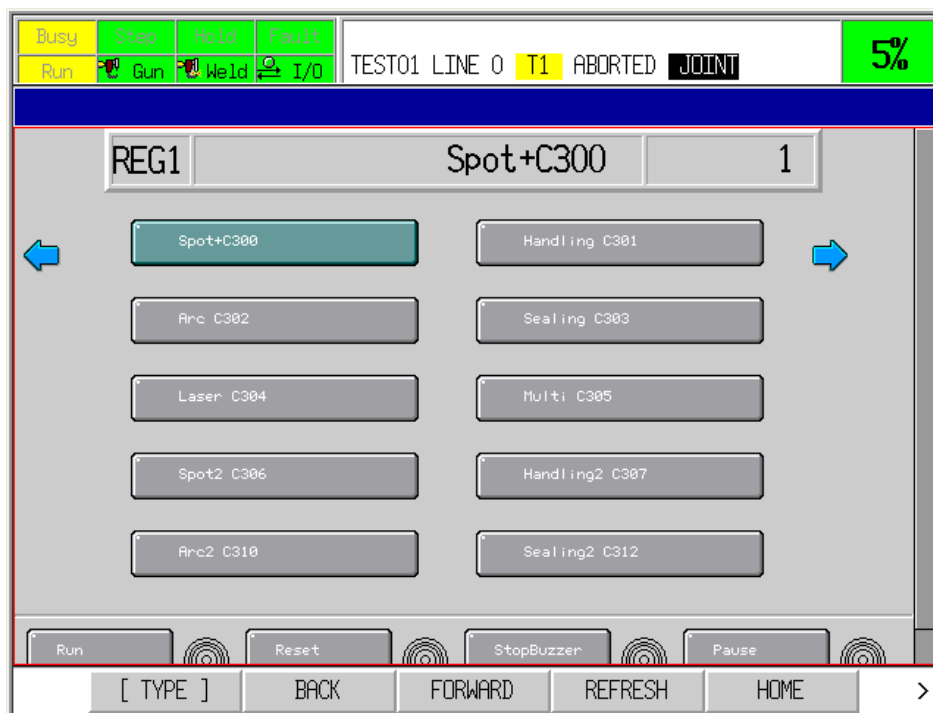


When a button allocated on a register panel is pressed, the specified setting value will be set to the specified register. And the current value of the register is displayed in the register display window.



When a register value is changed by not only the operation from the button of a register panel, but also other method, for example the program instruction execution or the operation from the register data screen, the label of a button, that setting value matches the current register value, will be displayed in the register display window.

Register panel can be used to select the kind of production goods. The setting value is used for branching the program execution in a PNS/RSR program.



The followings are sample program to select a sub program based on the value of a register.

```

SELECT R[1]=1, CALL C300      : If R[1] is 1, then call subprogram C300.
      =2, CALL C301      :
      =3, CALL C302
      =4, CALL C303
      =5, CALL C304
      =6, CALL C305
      =7, CALL C306
      =8, CALL C310
      =9, CALL C312
      OTHERWISE, JUMP LBL[100] : R[1] value does not match with either value, goto LBL[100]

LBL[100]
  UALM[1] : Assert a user defined alarm.

```

11.4.5 Caution

- By the quick button operation, operation will become invalid. Please perform the button operation slowly and surely.
- When you allocated an output signal for a button, please set the lamp for confirmation on a panel. And, at the time of the button operation, please confirm the state of the lamp by all means.
- The soft panel function supports Japanese and English, and the display of the setting screen about the soft panel changes by language. When change language, please do cycle power of the controller. But it is not converted well even if you change language to English when set Japanese on a button label.

11.5 RESTRICTIONS

- Multiple program edit screens can be opened at a time.
A screen can be opened in each window.
The program to be executed is the one selected in the left window.
The execution of a program from the teach pendant is possible only when the left window is the one that the user is working in.
The edit screen is always displayed in the left window.
- When two or more screens are displayed at a time, the same menu screen may not be displayed on the screens.
Example: Online position modification screen

* NetFront by ACCESS Co. Ltd. is adopted for the Internet function of this product.

ACCESS NetFront® v2.6

- * NetFront is registered trademark of Access Co. Ltd. in Japan.
- * Part of the software of this product includes modules developed by Independent JPEG Group.
- * This product use a technology included in LZW patent of Unisys Co. Ltd. Please keep following restrictions.
 - (1) Do not modify or copy the software of this product. Do not sale or provide the software extracted from this product.
 - (2) Do not use the software of this product for different purpose from browser.
 - (3) Do not use a technology included in LZW patent of Unisys Co. Ltd. without this product.

APPENDIX

A

SCREEN/PROGRAM INSTRUCTIONS

This appendix summarizes the items necessary for using this model. It may also be used as an index.

- Contents of this appendix**
- A.1 LIST OF MENUS
 - A.2 TYPES OF SCREENS
 - A.3 LIST OF PROGRAM INSTRUCTIONS
 - A.4 PROGRAM INSTRUCTIONS

A.1 LIST OF MENUS

Miscellaneous menu	Miscellaneous function
1 ABORT (ALL)	Program abort
2 Disable FWD/BWD	Disabling FWD/BWD from teach pendant
3 CHANGE GROUP	Change Motion Group ^{*14}
4 TOGGLE SUB GROUP	Sub group toggle ^{*17}
5 TOGGLE WRIST JOG	Wrist jog toggle
6	
7 RELEASE WAIT	Release Wait
8	
9	
0 — NEXT —	

Fig. A.1 (a) Function menu (Page 1)

Miscellaneous menu	Miscellaneous function
1 QUICK/FULL MENUS	Quick/full menu switch
2 SAVE	Save
3 PRINT SCREEN	Screen print
4 PRINT	Print
5	
6 UNSIM ALL I/O	Release the all of the simulated I/O
7	
8	
9 ENABLE HMI MENUS	Enable the HMI menu
0 — NEXT —	

Fig. A.1 (b) Function menu (Page 2)

Miscellaneous menu	Miscellaneous function
1 REFRESH PANE	Refresh the pane
2	
3	
4	
5	
6	
7 Diagnostic log	Save the diagnostic log data
8 Del Diag Log	Delete the saved diagnostic log data
9	
0 — NEXT —	

Fig. A.1 (c) Function menu (Page 3)

A.SCREEN/PROGRAM INSTRUCTIONS

B-83134EN-1/01

APPENDIX

Screen menu	Screen change menu	Screen
1 UTILITIES	Hints	----Hint F5→ Help
	Prog Adjust ^{*2}	----Program Adjust Schedule List F2→ Program Adjust Schedule Detail
	Program shif ^{*3}	----Program Shift
	Mirror Image ^{*4}	----Mirror Image Shift
	Tool offset ^{*5}	----Tool Offset
	Frame offset ^{*6}	----UFrame Offset
	Angle entry ^{*7}	----Angle Entry Shift
2 TEST CYCLE	Test Cycle	----Test cycle
3 MANUAL FCNTS	Macros ^{*8}	----Manual operation
4 ALARM	Alarm Log	----Alarm occurrence ↔ Alarm history F5→ Alarm detail
	Motion Log	----Motion alarm F5→ Alarm detail
	System Log	----System alarm F5→ Alarm detail
	Appl Log	----Application alarm F5→ Alarm detail
	Password Log	----Password log F5→ Alarm detail
	Comm Log	----Communication alarm F5→ Alarm detail
5 I / O	Cell Intface	----I/O cell input/output screen F2→ Allocation
	Digital	----Digital I/O F2→ Digital configuration F4→ Digital detail
	Group	----Group I/O F2→ Group Configuration F4→ Group detail
	Analog ^{*9}	----Analog I/O F2→ Analog configuration F4→ Analog detail
	Robot	----Robot I/O F2→ Robot I/O detail
	UOP	----Peripheral device I/O
	SOP	----System Operator Panel I/O
	Inter Conect	----DI-to-DO connection setting
	Link Device	----I/O Link Device List F3→ I/O Unit Model B List F3→ I/O Points Setup
	Flag	----Flag screen F2→ Flag detail
6 SETUP	Prog Select	----Program selection screen F3→ Detail
	General	----General item setting
	Frames	----Frame entry F2→ Frame detail

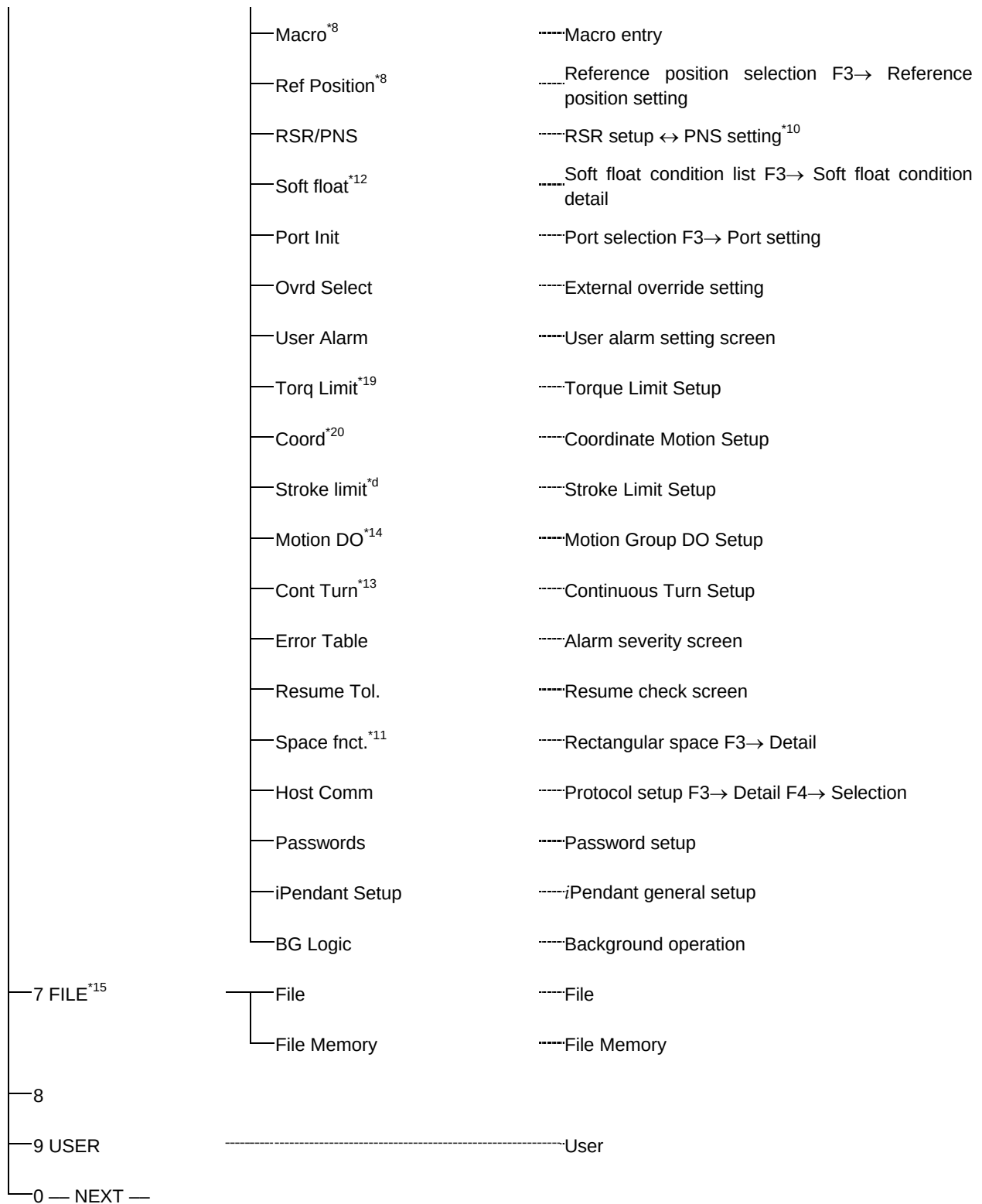


Fig. A.1 (d) Screen Menu (Page 1)

Screen menu	Screen change menu	Screen
1 SELECT		Program list
2 EDIT		Program edit
3 DATA	Registers	Register
	Position Reg ^{*15}	Position register F4→ Position data information
	Pallet regis ^{*1}	Palletizing register
	KAREL Vars	Karel variable
	KAREL Posns	Karel position variable
4 STATUS	Axis	Robot axis status
	Version ID	Software version
	Prg Timer ^{*16}	Program Timer Status List F2→ Program Timer Detail
	Sys Timer ^{*16}	System Timer Status
	Safety Signl	Safety Signal Status
	Order File ^{*a}	Order file
	Exec-hist	Execution history
	Memory	Memory status list F2→ Memory status detail
	Condition ^{*18}	Condition Monitor
	Robot ready	Robot ready screen
5 POSITION		Current position
6 SYSTEM	Clock	Calendar
	Variables	System variable
	Servo Param ^{*c}	Servo parameter
	Master/Cal ^{*b}	Positioning
	OT Release	Over Travel Release
	Axis Limits	Joint operating area setting
	Config	System configuration
	Motion	Motion Performance
7 USER2		
8 BROWER (only when iPendant is used)		

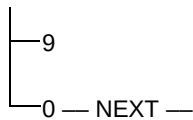


Fig. A.1 (e) Screen Menu (Page 2)

Setting

The menu items indicated by * (alphabetical character) in Fig. A.1 (d) and Fig. A.1 (e) are displayed when the corresponding setting is made, as indicated below:

Table A.1 (a) Option list

*	Setting
a	Can be displayed by setting \$ODRDSP_ENB to 1.
b	Can be displayed by setting \$MASTER_ENBL to 1.
c	Can be displayed by setting \$SVPRM_ENB to 1.
d	Depend on the robot type.

Options

The menu items indicated by * (numeral) in Fig. A.1 (a) to Fig. A.1 (e) are displayed when the corresponding option is added, as indicated below:

Table A.1 (b) Option list

*	Option	Specification
1	Palletizing	A05B-2500-J500
2	Online touchup	A05B-2500-H510
3	Program shift	A05B-2500-H510
4	Mirror image	A05B-2500-H510
5	Tool offset	A05B-2500-H510
6	User frame input	A05B-2500-H510
7	Angle entry shift	A05B-2500-J614
8	Option command	A05B-2500-H510
9	Analog I/O	A05B-2500-H510
10	External program selection	A05B-2500-H510
11	Space check	A05B-2500-H510
12	Soft float	A05B-2500-J612
13	Continuous turn	A05B-2500-J613
14	Multi motion group	A05B-2500-J601
15	Position register	A05B-2500-H510
16	Hour meter	A05B-2500-H510
17	Extended axis control	A05B-2500-J518
18	Condition monitor function	A05B-2500-H510
19	Torque limit function	A05B-2500-J611
20	Coordinate motion package	A05B-2500-J686

A.2 TYPES OF SCREENS

Hint Screen

UTILITIES Hints
JOINT 10%

HandlingTool
V7.70P/13 7DA7/13

Copyright 2010, All Rights Reserved
FANUC LTD, FANUC Robotics America, Inc.
Licensed Software: Your use constitutes
your acceptance. This product protected
by several U.S. patents.

TYPE
LICENSE
PATENTS
HELP

Hints

Program Adjust Schedule List Screen

UTILITIES Prog Adj
JOINT 10%

Program Adjust Schedule 1/10

#	Program	Lines	Status
1	TEST1	1-2	EDIT
2	*****	0-0	*****
3	*****	0-0	*****
4	*****	0-0	*****
5	*****	0-0	*****
6	*****	0-0	*****
7	*****	0-0	*****
8	*****	0-0	*****
9	*****	0-0	*****

TYPE
DETAIL
>

Prog Adjust

COPY
CLR_ADJ
CLR_ALL
>

Program Adjust Schedule Detail Screen

UTILITIES Prog Adj
JOINT 10%

1/14

Current schedule: 1 Status:EDIT

Program Name:

1 TEST1

2 Starting line number: 1

3 Ending line number: 2

4 Offset relative to: USER

5 X adjustment: 26.000 mm

6 Y adjustment: 0.000 mm

7 Z adjustment: 0.000 mm

8 W adjustment: 0.000 deg

[TYPE]
UNITS
SCHED
[CHOICE]
>

COPY
CLR_ADJ
CLR_ALL
[CHOICE]
>

Program Shift (Program name) Screen

PROGRAM SHIFT
JOINT 10%

Program 1/6

Original program:

1 Test1

2 Range: WHOLE

3 Start line: (not used) *****

4 End line: (not used) *****

New program:

5 Test2

6 Insert line:(not used) *****

Use shifted up,down arrows for next page

TYPE
>

Program shif

CLEAR
>

Program Shift (Position) Screen

PROGRAM SHIFT
JOINT 10%

Shift amount/Teach 1/3

Position data

X :***** Y :***** Z :*****

1 Rotation OFF

2 Source position P1:

3 Destination position Q1:

[TYPE]
EXECUTE
ON
OFF
>

CLEAR
DIRECT
ON
OFF
>

Fig. A.2 (a) [1 UTILITIES]

Program Shift (Direct entry) Screen

PROGRAM SHIFT		JOINT 10%
Shift amount/Direct entry		1/3
1 X (mm):		100.00
2 Y (mm):		0.00
3 Z (mm):		0.00
[TYPE] EXECUTE		>
CLEAR TEACH		>

Mirror Image Shift (Program name) Screen

MIRROR IMAGE SHIFT		JOINT 10%
Program		1/6
Original program:		
1	Test1	
2	Range:	WHOLE
3	Start line: (not used)	*****
4	End line: (not used)	*****
New program:		
5	Test3	
6	Insert line:(not used)	*****
Use shifted up,down arrows for next page		
TYPE		>
Mirror Image		
CLEAR		>

Mirror Image Shift (Position) Screen

MIRROR IMAGE SHIFT		JOINT 10%
Shift amount/Teach		1/3
Position data		
X :***** Y :***** Z :*****		
1 Rotation		OFF
2 Source position	P1:	
3 Destination position	Q1:	
[TYPE] EXECUTE		ON OFF >
CLEAR		ON OFF >

Tool Offset (Program name) Screen

TOOL OFFSET		JOINT 10%
Program		5/6
Original program:		
1	Test1	
2	Range:	WHOLE
3	Start line: (not used)	*****
4	End line: (not used)	*****
New program:		
5	Test4	
6	Insert line:(not used)	*****
Use shifted up,down arrows for next page		
TYPE		>
Tool offset		
CLEAR		>

Tool Offset (Utool number) Screen

TOOL OFFSET		JOINT 10%
UTOOL number		1/3
1 Old UTOOL number:		0
2 New UTOOL number:		0
3 Convert type:		TCP fixed
[TYPE] EXECUTE		>
CLEAR		>

Fig. A.2 (b) [1 UTILITIES]

Uframe Offset (Program name) Screen <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">UFRAME OFFSET</td> <td style="width: 40%; text-align: right;">JOINT 10%</td> </tr> <tr> <td>Program</td> <td style="text-align: right;">5/6</td> </tr> <tr> <td colspan="2">Original program:</td> </tr> <tr> <td>1 Test1</td> <td></td> </tr> <tr> <td>2 Range:</td> <td style="text-align: right;">WHOLE</td> </tr> <tr> <td>3 Start line: (not used)</td> <td style="text-align: right;">*****</td> </tr> <tr> <td>4 End line: (not used)</td> <td style="text-align: right;">*****</td> </tr> <tr> <td colspan="2">New program:</td> </tr> <tr> <td>5 Test4</td> <td></td> </tr> <tr> <td>6 Insert line:(not used)</td> <td style="text-align: right;">*****</td> </tr> </table> Use shifted up,down arrows for next page TYPE > Frame offset CLEAR >	UFRAME OFFSET	JOINT 10%	Program	5/6	Original program:		1 Test1		2 Range:	WHOLE	3 Start line: (not used)	*****	4 End line: (not used)	*****	New program:		5 Test4		6 Insert line:(not used)	*****	Uframe Offset (Uframe number) Screen <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">UFRAME OFFSET</td> <td style="width: 40%; text-align: right;">JOINT 10%</td> </tr> <tr> <td>UFRAME number</td> <td style="text-align: right;">1/3</td> </tr> <tr> <td colspan="2">1 Old UFRAME number: 0</td> </tr> <tr> <td colspan="2">2 New UFRAME number: 0</td> </tr> <tr> <td colspan="2">3 Convert Position data (Y/N): NO</td> </tr> </table> [TYPE] EXECUTE > CLEAR >	UFRAME OFFSET	JOINT 10%	UFRAME number	1/3	1 Old UFRAME number: 0		2 New UFRAME number: 0		3 Convert Position data (Y/N): NO									
UFRAME OFFSET	JOINT 10%																																						
Program	5/6																																						
Original program:																																							
1 Test1																																							
2 Range:	WHOLE																																						
3 Start line: (not used)	*****																																						
4 End line: (not used)	*****																																						
New program:																																							
5 Test4																																							
6 Insert line:(not used)	*****																																						
UFRAME OFFSET	JOINT 10%																																						
UFRAME number	1/3																																						
1 Old UFRAME number: 0																																							
2 New UFRAME number: 0																																							
3 Convert Position data (Y/N): NO																																							
Angle Entry Shift (Program name) Screen <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">ANGLE ENTRY SHIFT</td> <td style="width: 40%; text-align: right;">JOINT 10%</td> </tr> <tr> <td>Program</td> <td style="text-align: right;">5/6</td> </tr> <tr> <td colspan="2">1 Original program: [TEST1]</td> </tr> <tr> <td>2 Range:</td> <td style="text-align: right;">WHOLE</td> </tr> <tr> <td>3 Start line: (not used)</td> <td style="text-align: right;">*****</td> </tr> <tr> <td>4 End line: (not used)</td> <td style="text-align: right;">*****</td> </tr> <tr> <td colspan="2">5 New program: [Test6]</td> </tr> <tr> <td>6 Insert line: (not used)</td> <td style="text-align: right;">*****</td> </tr> </table> Use shifted up,down arrows for next page TYPE > Angle entry CLEAR >	ANGLE ENTRY SHIFT	JOINT 10%	Program	5/6	1 Original program: [TEST1]		2 Range:	WHOLE	3 Start line: (not used)	*****	4 End line: (not used)	*****	5 New program: [Test6]		6 Insert line: (not used)	*****	Angle Entry Shift (Shift amount) Screen <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">ANGLE ENTRY SHIFT</td> <td style="width: 40%; text-align: right;">JOINT 10%</td> </tr> <tr> <td>Shift amount</td> <td style="text-align: right;">1/7</td> </tr> <tr> <td colspan="2">Position data of P1</td> </tr> <tr> <td colspan="2">X :***** Y :***** Z :*****</td> </tr> <tr> <td>1 Rotation plane</td> <td style="text-align: right;">P1:</td> </tr> <tr> <td>2</td> <td style="text-align: right;">P2:</td> </tr> <tr> <td>3</td> <td style="text-align: right;">P3:</td> </tr> <tr> <td>4 Rotation axis enable:</td> <td style="text-align: right;">FALSE</td> </tr> <tr> <td>5 Rotation axis</td> <td style="text-align: right;">P0: Not used</td> </tr> <tr> <td>6 Angle(deg):</td> <td style="text-align: right;">0.00</td> </tr> <tr> <td>7 Repeating times:</td> <td style="text-align: right;">1</td> </tr> </table> [TYPE] EXECUTE REFER RECORD > CLEAR >	ANGLE ENTRY SHIFT	JOINT 10%	Shift amount	1/7	Position data of P1		X :***** Y :***** Z :*****		1 Rotation plane	P1:	2	P2:	3	P3:	4 Rotation axis enable:	FALSE	5 Rotation axis	P0: Not used	6 Angle(deg):	0.00	7 Repeating times:	1
ANGLE ENTRY SHIFT	JOINT 10%																																						
Program	5/6																																						
1 Original program: [TEST1]																																							
2 Range:	WHOLE																																						
3 Start line: (not used)	*****																																						
4 End line: (not used)	*****																																						
5 New program: [Test6]																																							
6 Insert line: (not used)	*****																																						
ANGLE ENTRY SHIFT	JOINT 10%																																						
Shift amount	1/7																																						
Position data of P1																																							
X :***** Y :***** Z :*****																																							
1 Rotation plane	P1:																																						
2	P2:																																						
3	P3:																																						
4 Rotation axis enable:	FALSE																																						
5 Rotation axis	P0: Not used																																						
6 Angle(deg):	0.00																																						
7 Repeating times:	1																																						
Angle Entry Shift (Rotation axis direct entry) Screen <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">ANGLE ENTRY SHIFT</td> <td style="width: 40%; text-align: right;">JOINT 10%</td> </tr> <tr> <td>Shift amount</td> <td style="text-align: right;">1/4</td> </tr> <tr> <td colspan="2">Rotation center axis direct entry</td> </tr> <tr> <td>1 Frame:</td> <td style="text-align: right;">WORLD FRAME</td> </tr> <tr> <td>1 X (mm):</td> <td style="text-align: right;">100.00</td> </tr> <tr> <td>2 Y (mm):</td> <td style="text-align: right;">0.00</td> </tr> <tr> <td>3 Z (mm):</td> <td style="text-align: right;">0.00</td> </tr> </table> [TYPE] EXECUTE [CHOICE] > CLEAR TEACH [CHOICE] >		ANGLE ENTRY SHIFT	JOINT 10%	Shift amount	1/4	Rotation center axis direct entry		1 Frame:	WORLD FRAME	1 X (mm):	100.00	2 Y (mm):	0.00	3 Z (mm):	0.00																								
ANGLE ENTRY SHIFT	JOINT 10%																																						
Shift amount	1/4																																						
Rotation center axis direct entry																																							
1 Frame:	WORLD FRAME																																						
1 X (mm):	100.00																																						
2 Y (mm):	0.00																																						
3 Z (mm):	0.00																																						

Fig. A.2 (c) TEST CYCLE]

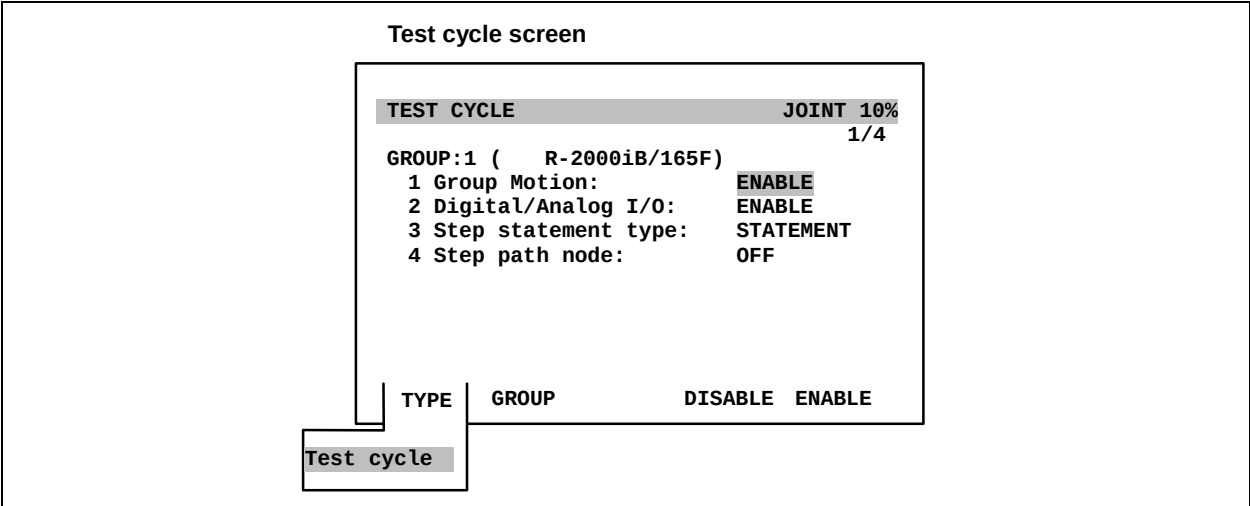


Fig. A.2 (d) [2 TEST CYCLE]

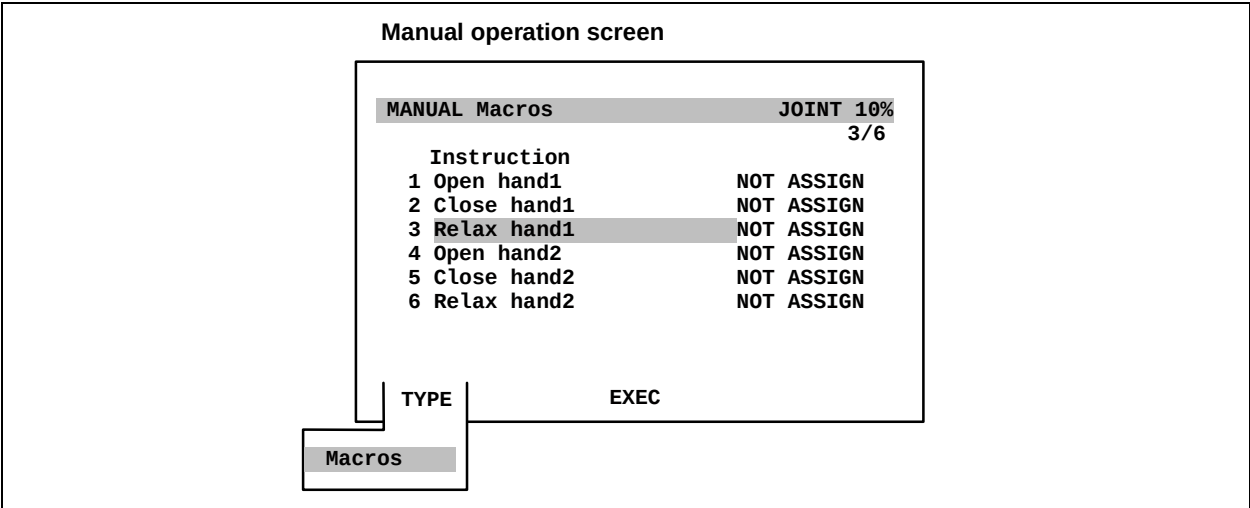


Fig. A.2 (e) [3 MANUAL FCTNS]

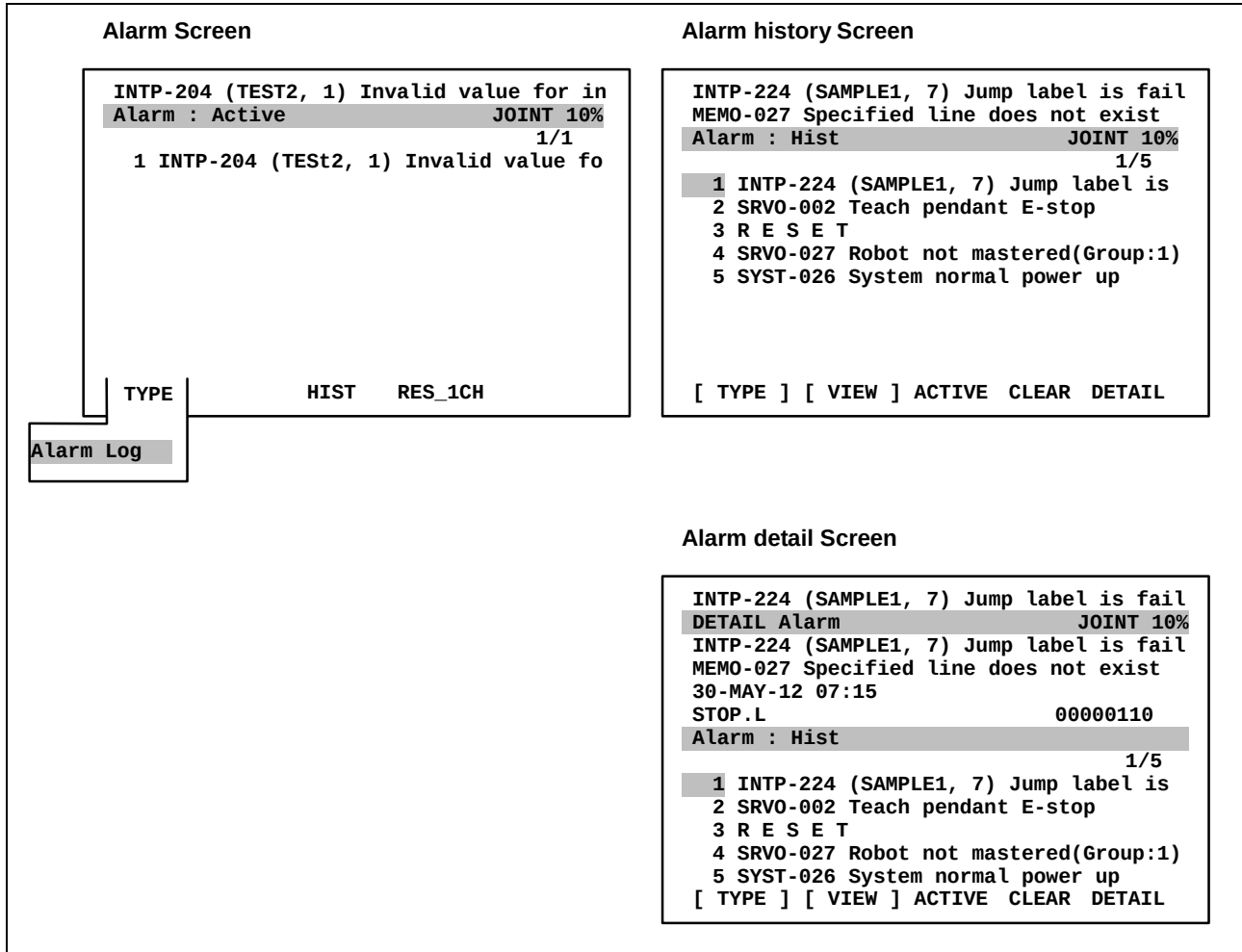


Fig. A.2 (f) [4 ALARM]

Digital input screen

I/O Digital In				JOINT 30%
#	SIM	STATUS		
DI[1]	U	ON	[	]
DI[2]	U	OFF	[	]
DI[3]	U	OFF	[	]
DI[4]	U	ON	[	]
DI[5]	U	ON	[	]
DI[6]	U	OFF	[	]
DI[7]	U	OFF	[	]
DI[8]	U	ON	[	]
DI[9]	U	ON	[	]

TYPE CONFIG IN/OUT ON OFF

Digital

Digital output configuration screen

I/O Digital Out				JOINT 30%
#	RANGE	RACK	SLOT	START 2/9 STAT.
1	DO[1- 8]	0	1	21 ACTIV
2	DO[9- 16]	0	1	29 ACTIV
3	DO[17- 24]	0	1	37 ACTIV
4	DO[25- 32]	0	2	1 ACTIV
5	DO[33- 40]	0	2	9 ACTIV
6	DO[41- 48]	0	2	17 ACTIV
7	DO[49- 56]	0	2	25 ACTIV
8	DO[57- 64]	0	2	33 ACTIV
9	DO[65- 72]	*	*	* UNASG

TYPE MONITOR IN/OUT DETAIL HELP >

Group output screen

I/O Group Out				JOINT 30%
#	SIM	VALUE		
GO[1]	S	1	[	]
GO[2]	U	10	[	]
GO[3]	U	23	[	]
GO[4]	*	0	[	]
GO[5]	*	0	[	]
GO[6]	*	0	[	]
GO[7]	*	0	[	]
GO[8]	*	0	[	]
GO[9]	*	0	[	]
GO[10]	*	0	[	]

TYPE CONFIG IN/OUT SIMULATE UNSIM

Group

Group output configuration screen

I/O Group Out				JOINT 30%
GO #	RACK	SLOT	START PT	NUM PTS
1	0	2	17	4
2	0	2	21	4
3	0	2	25	8
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

[TYPE] MONITOR IN/OUT HELP >

Analog input screen

I/O Analog In				JOINT 30%
#	SIM	VALUE		1/25
AI[1]	S	85	[	]
AI[2]	U	0	[	]
AI[3]	*	*	[	]
AI[4]	*	*	[	]
AI[5]	*	*	[	]
AI[6]	*	*	[	]
AI[7]	*	*	[	]
AI[8]	*	*	[	]
AI[9]	*	*	[	]
AI[10]	*	*	[	]

TYPE CONFIG IN/OUT SIMULATE UNSIM

Analog

Analog input configuration screen

I/O Analog In				JOINT 30%
AI #	RACK	SLOT	CHANNEL	1/25
1	0	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	

[TYPE] MONITOR IN/OUT HELP >

Fig. A.2 (f) [5 I / O]

Robot data output screen

I/O Robot Out
JOINT 30%
1/8

#	SIM	STATUS	
RO[1]	U	OFF	[]
RO[2]	U	OFF	[]
RO[3]	U	OFF	[]
RO[4]	U	ON	[]
RO[5]	U	ON	[]
RO[6]	U	OFF	[]
RO[7]	U	OFF	[]
RO[8]	U	ON	[]

TYPE
IN/OUT
ON
OFF
>

Robot

Robot output detail screen

I/O Robot Out
JOINT 30%
1/3

Port Detail

Robot Dig. Output [1]

1 Comment: []

2 Polarity: NORMAL

3 Complementary: FALSE [1 - 2]

[TYPE] PRV-PT NXT-PT

Peripheral device data input screen

I/O UOP In
JOINT 30%

#	STATUS	
UI[1]	ON	[*HOLD]
UI[2]	OFF	[FAULT reset]
UI[3]	OFF	[Start]
UI[4]	ON	[Enable]
UI[5]	ON	[PNS1]
UI[6]	OFF	[PNS2]
UI[7]	OFF	[PNS3]
UI[8]	ON	[PNS4]
UI[9]	*	[]
UI[10]	*	[]

TYPE
CONFIG
IN/OUT
ON
OFF

UOP

System Operation Panel Output screen

I/O SOP Out
JOINT 30%

#	STATUS	
SO[0]	ON	[Remote LED]
SO[1]	OFF	[Cycle start]
SO[2]	OFF	[Hold]
SO[3]	ON	[Fault LED]
SO[4]	ON	[Batt alarm]
SO[5]	OFF	[]
SO[6]	OFF	[]
SO[7]	OFF	[TP enabled]
SO[8]	OFF	[]
SO[9]	OFF	[]

TYPE
IN/OUT
ON
OFF

SOP

DI-to-DO connection setting screen(RI->DO)

INTERCONNECT
JOINT 30%
1/8

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]

TYPE
[SELECT]
ENABLE
DISABLE

Interconnect

DI-to-DO connection setting screen(DI->RO)

INTERCONNECT
JOINT 30%
1/8

No.	Enb/Disabl	INPUT	OUTPUT
1	DISABLE	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]

TYPE
[SELECT]
ENABLE
DISABLE

Interconnect

Fig. A.2 (g) [5 I / O]

DI-to-DO connection setting screen(DI->DO)

INTERCONNECTJOINT 30%1/32

No.	Enb/Disabl	INPUT	OUTPUT
1	DISABLE	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]
7	DISABLE	DI[0]->	DO[0]
8	DISABLE	DI[0]->	DO[0]
9	DISABLE	DI[0]->	DO[0]

TYPE

[SELECT] ENABLE DISABLE

Interconnect

I/O Link Device Screen

I/O Link DeviceJOINT 30%1/4

Device name	Comment	Rack	Slot
1 PrcI/O CA	[]	0	1
2 Model A	[]	1	0
3 Model B	[]	2	0
4 Model C	[]	3	0

TYPE

DETAIL CLR_ASG

Link Device

I/O Link Device (Model B) List Screen

I/O Link DeviceJOINT 30%1/30

Model B

Slot	Base	Exp.	Comment
1	*****	*****	[]
2	*****	*****	[]
3	*****	*****	[]
4	*****	*****	[]

[TYPE]

LIST CLR_ASG

I/O Link Device (I/O Points Setup) Screen

I/O Link DeviceJOINT 100%1/2

90-30 PLC Rack 1 Slot 1

Port name	Points
1 Digital Input:	0
2 Digital Output:	0

[TYPE]

LIST CLR_ASG

Fig. A.2 (h) [5 I / O]

- 950 -

General item setting screen

SETUP General JOINT 30% 1/4

1 Break on hold: **DISABLED**

2 Current language: **DEFAULT**

3 Ignore Offset command: **DISABLED**

4 Ignore Tool_offset: **DISABLED**

5 Enable VOFFSET: **DISABLED**

TYPE
ENABLED
DISABLED

General

Tool frame entry screen

SETUP Frame JOINT 30% 1/10

Tool Frame	X	Y	Z	Direct Entry	Comment
1:	0.0	0.0	140.5	[	]
2:	0.0	0.0	200.0	[	]
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	[	]

1 Tool Frame

2 Jog Frame

3 User Frame

Active Tool 1

TYPE
DETAIL
OTHER
CLEAR
SETIND

Frames

Tool frame setup screen (Three Point Method)

SETUP Frames JOINT 30% 3/4

Tool Frame Three Point

Frame Number: 3

X: 0.0 Y: 0.0 Z: 0.0

W: 0.0 P: 0.0 R: 0.0

Comment: REF FRM

Approach point 1: RECORDED

Approach point 2: UNINIT

Approach point 3: UNINIT

1 Three Point

2 Six Point

3 Direct Entry

Active Tool UM[1] = 1

[TYPE]
METHOD
FRAME
MOVE_TO
RECORD

Frames

Tool frame setup screen (Six Point Method)

SETUP Frames JOINT 30% 4/7

Tool Frame Six Point

Frame Number: 3

X: 100.0 Y: 0.0 Z: 300.0

W: 0.0 P: 0.0 R: 0.0

Comment: *****

Approach point 1: RECORDED

Approach point 2: RECORDED

Approach point 3: RECORDED

Or

X 1 Three Point UNINIT

Y 2 Six Point UNINIT

Z 3 Direct Entry UNINIT

Active Tool UM[1] = 1

[TYPE]
METHOD
FRAME
MOVE_TO
RECORD

Frames

Tool frame setup screen (Direct Entry Method)

SETUP Frames JOINT 30% 3/7

Tool Frame Direct Entry

Frame Number: 3

1: Comment: *****

2: X: 0.000

3: Y: 0.000

4: Z: 200.000

5: W: 0.000

6: P: 0.000

7: R: 0.000

N D B, 0, 0, 0

1 Three Point

2 Six Point

3 Direct Entry

Active Tool UM[1] = 1

[TYPE]
METHOD
FRAME

Frames

User frame entry screen

SETUP Frames JOINT 30% 1/9

User Frame / Three Point

User Frame	X	Y	Z	Comment
1:	1243.6	0.0	43.8	[REF FRM]
2:	1243.6	525.2	43.8	[
3:	1243.6	-525.2	43.8	[
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	[

1 Tool Frame

2 Jog Frame

3 User Frame

Active UFRAME = 0

TYPE
DETAIL
OTHER
CLEAR
SETIND

Frames

Fig. A.2 (i) [6 SETUP]

User frame setup screen (Three Point Method)

SETUP Frames				JOINT 30%
User Frame	Three Point			1/4
Frame Number: 1				
X: 1243.6	Y: 0.0	Z: 43.8		
W: 0.1	P: 2.3	R: 3.2		
Comment:*****				
Orient Origin Point: UNINIT				
X Direction Point: UNINIT				
Y Direction Point: UNINIT				
1 Three Point 2 Four Point 3 Direct Entry				
Active	AMNUM[1] = 0			
[TYPE]	METHOD	FRAME MOVE_TO RECORD		

User frame setup screen (Four Point Method)

SETUP Frames				JOINT 30%
User Frame	Four Point			5/5
Frame Number: 1				
X: 1243.6	Y: 0.0	Z: 10.0		
W: 0.1	P: 2.3	R: 3.2		
Comment: REF FRM				
Orient Origin Point: USED				
X Direction Point: USED				
Y Direction Point: USED				
1 Three Point 2 Four Point 3 Direct Entry				
Active	AMNUM[1] = 0			
[TYPE]	METHOD	FRAME		

Jog frame entry screen

SETUP Frames				JOINT 30%
Jog Frame	/ Three Point			1/5
X	Y	Z	Comment	
1: 1243.6	0.0	0.0	[WORK AREA 1]	
2: 1003.0	525.2	60.0	[WORK AREA 2]	
3: 1003.0	236.0	90.0	[WORK AREA 3]	
4: 0.0	0.0	0.0	[]	
5: 0.0	0.0	0.0	[]	
1 Tool Frame 2 Jog Frame 3 User Frame				
Active	JOG			
TYPE	DETAIL	OTHER	CLEAR SETIND	
Frames				

Jog frame setup screen (Direct Entry Method)

SETUP Frames				JOINT 30%
Jog Frame	Direct Entry			4/7
Frame Number: 1				
1: Comment:	WORK AREA 1			
2: X:	1243.600			
3: Y:	0.000			
4: Z:	10.000			
5: W:	0.123			
6: P:	2.340			
7: R:	3.200			
N D B,0,0,0				
1 Three Point 2 Direct Entry				
Active	1			
[TYPE]	METHOD	FRAME MOVE_TO RECORD		

Reference position selection screen

REF POSN				JOINT 30%
No.	Enb/Dsbl	@Pos	Comment	1/3
1	ENABLE	TRUE	[REFPOS1]	
2	DISABLE	FALSE	[REFPOS2]	
3	DISABLE	FALSE	[REFPOS3]	
TYPE				
DETAIL				
ENABLE				
DISABLE				
Ref Position				

Reference position setting screen

REF POSN				JOINT 30%
Reference Position				1/13
Ref.Position Number: 1				
1	Comment:	[REFPOS1]		
2	Enable/Disable:	ENABLE		
3	Is a valid HOME:	FALSE		
4	Signal definition:	DO[0]		
5	J1:	129.000 +/-	2.000	
6	J2:	-31.560 +/-	2.000	
7	J3:	3.320 +/-	2.000	
8	J4:	179.240 +/-	2.000	
9	J5:	1.620 +/-	2.000	
[TYPE]				RECORD

Fig. A.2 (j) [6 SETUP]

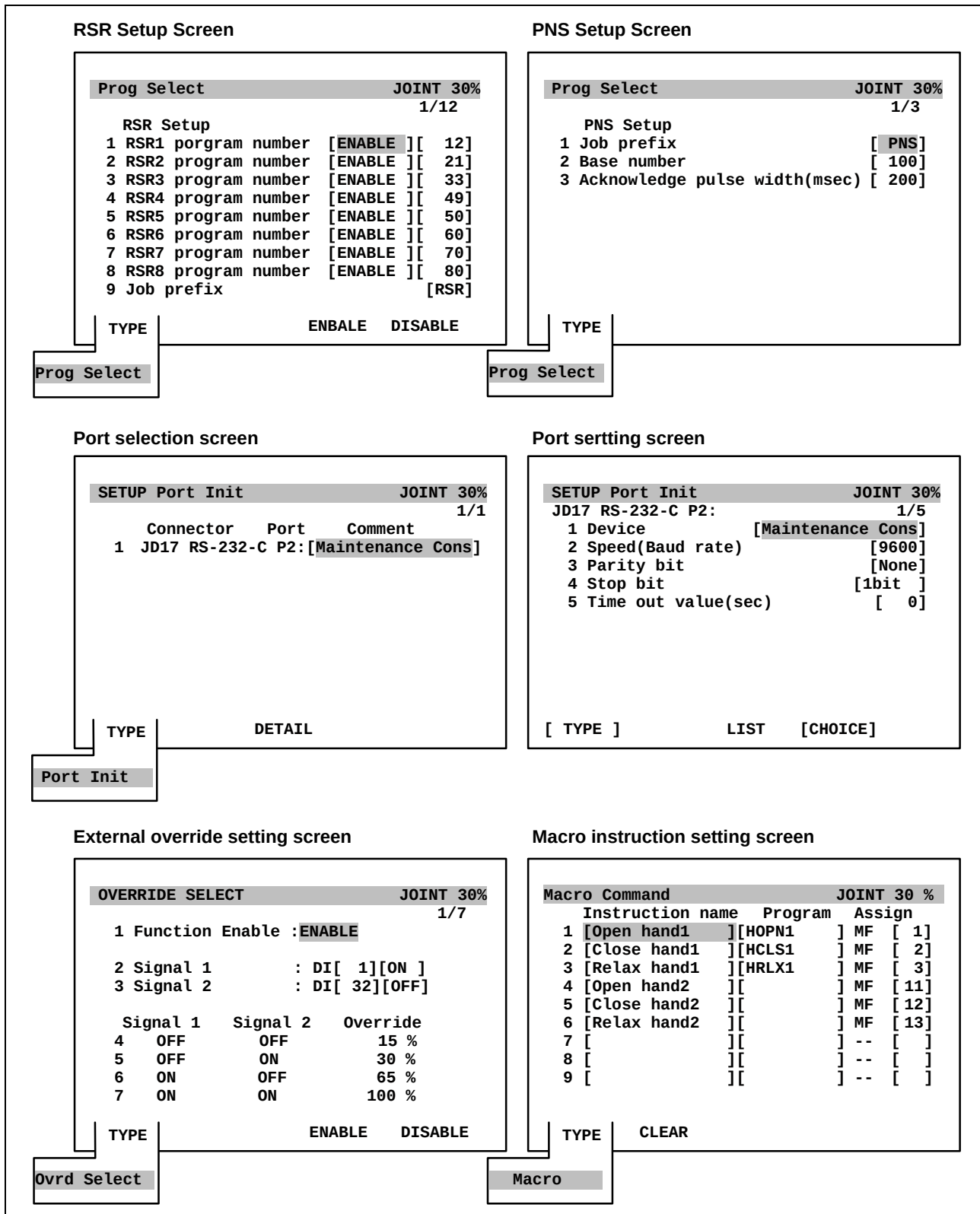


Fig. A.2 (k) [6 SETUP]

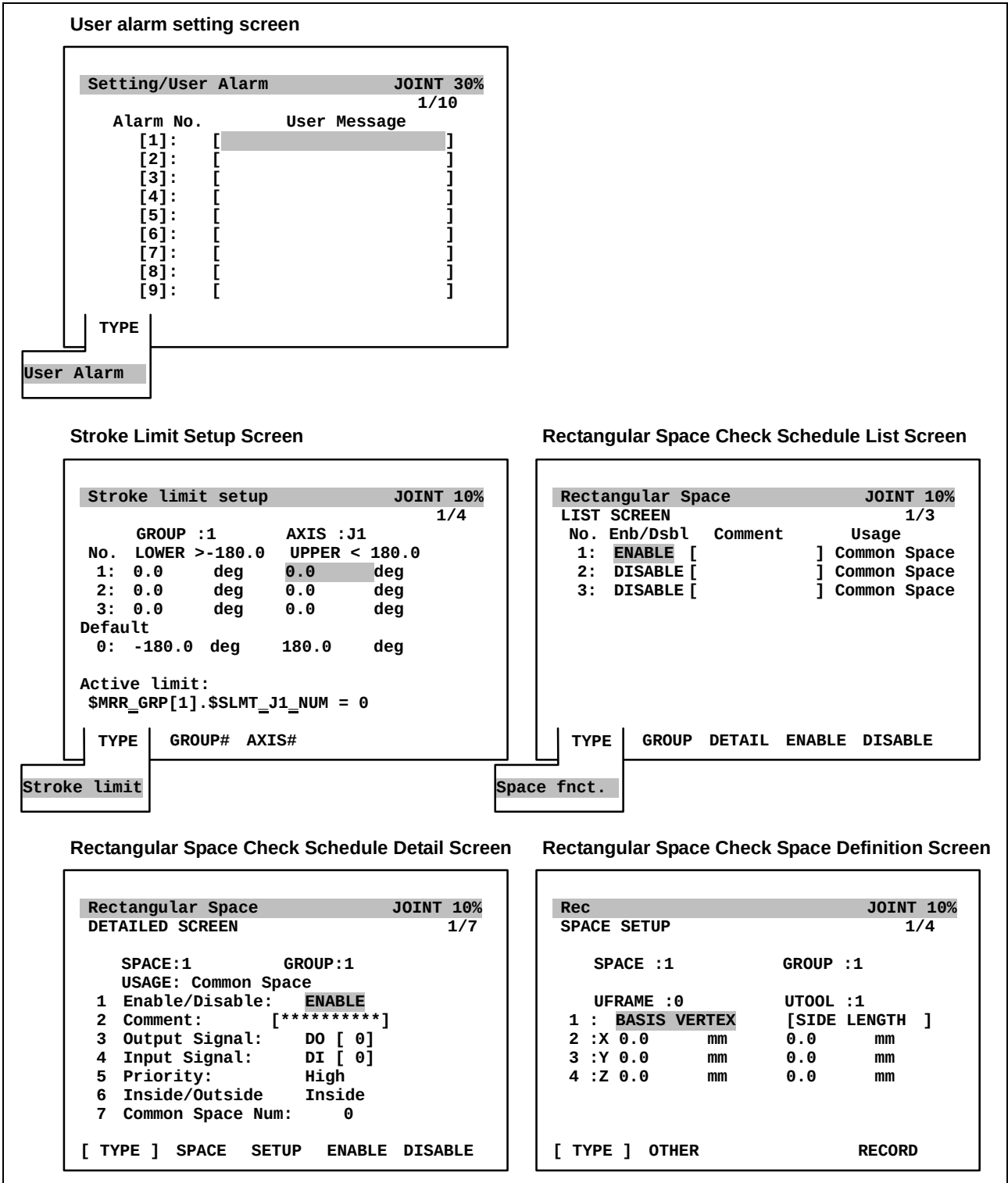


Fig. A.2 (I) [6 SETUP]

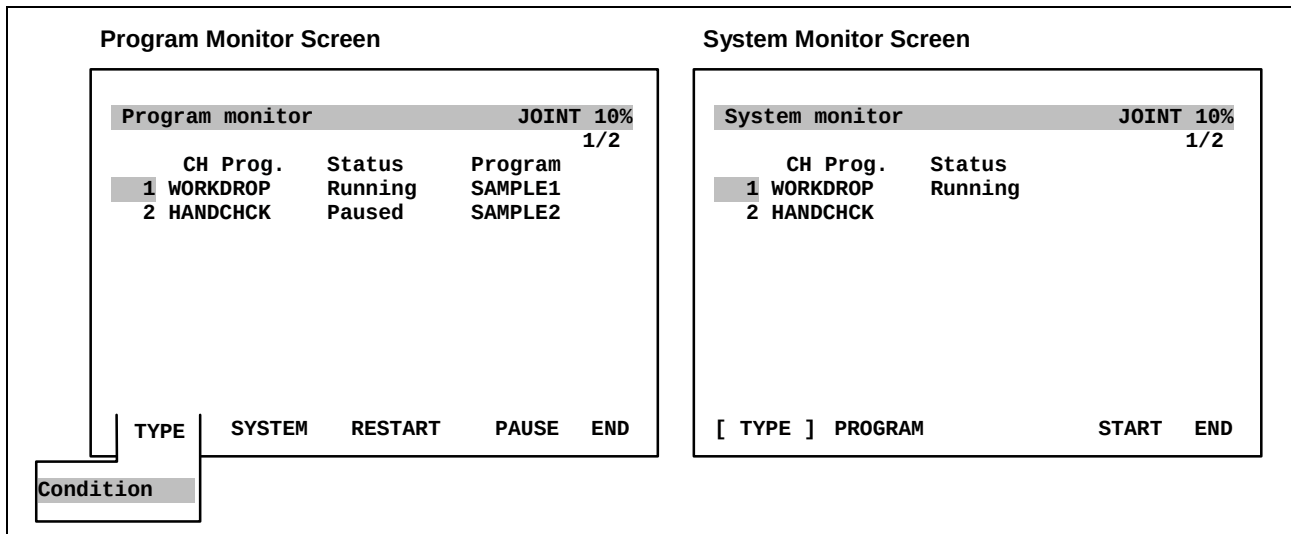


Fig. A.2 (m) [6 SETUP]

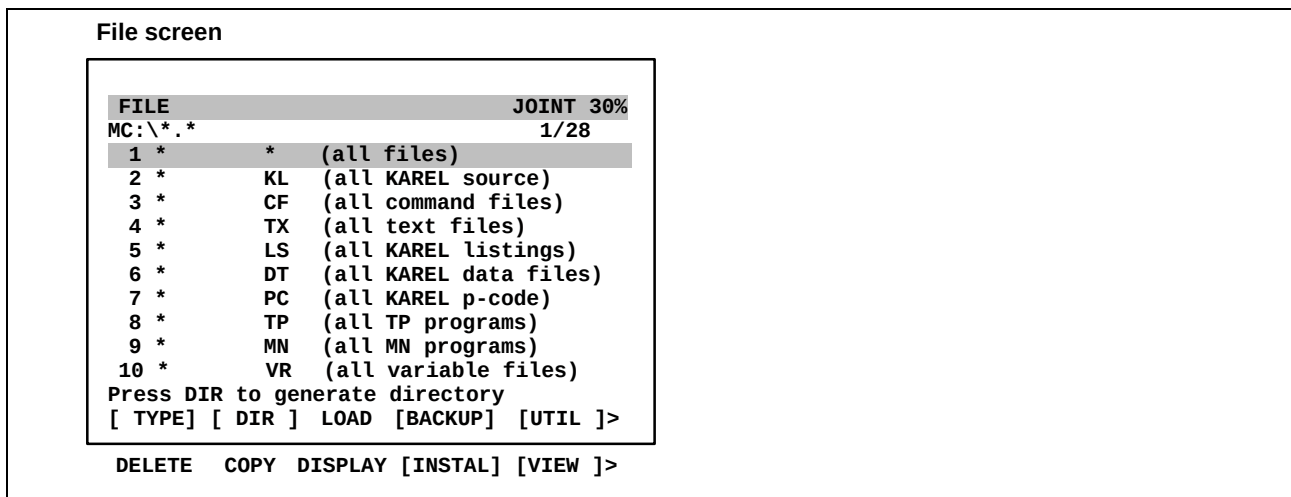


Fig. A.2 (n) [7 FILE]

Program list screen

Select		JOINT 30%
58740 bytes free		3/5
No.	Program name	Comment
1	SAMPLE1	JB[SAMPLE PROGRAM1]
2	SAMPLE2	JB[SAMPLE PROGRAM2]
3	SAMPLE3	JB[SAMPLE PROGRAM3]
4	PROG001	PR[]
5	PROG002	PR[]

1 Comment
 2 Protection
 3 Last Modified
 4 Size
 5 Copy Source
 6 Name Only

[TYPE] CREATE DELETE MONITOR ATTR >

COPY DETAIL LOAD SAVE AS PRINT >

Program registration screen

1 Words		JOINT 30%
2 Upper Case		
3 Lower Case		
4 Options		--Insert--

Select

--- Create Teach Pendant Program ---

Program Name
SAMPLE3

-- End --

Enter program name

RSR	PNS	STYLE	JOB	TEST

Program information screen

Program detail		JOINT 30%
		1/7
Creation Date:	10-MAR-1998	
Modification Data:	11-MAR-1998	
Copy Source:		
Positions: FALSE	Size:	312 Bytes
Program name:		
1	SAMPLE3	
2	Sub Type:	[None]
3	Comment:	[SAMPLE PROGRAM 3]

END PREV NEXT

Fig. A.2 (o) [1 SELECT]

Program edit screen

SAMPLE1		JOINT 30%
		1/6
1:	J P[1] 100% FINE	
2:	J P[2] 70% CNT50	
3:	L P[3] 1000cm/min C	
4:	L P[4] 500mm/sec FI	
5:	J P[1] 100% FINE	
[End]		

1 Insert
 2 Delete
 3 Copy
 4 Find
 5 Replace
 6 Renumber
 7 Comment
 8 Undo
 9 Remark

[INST] EDCMD >

POINT TOUCHUP>

Program edit screen/control instruction list screen

Instruction		JOINT 30%
1 Register	5 JMP/LBL	
2 I/O	6 CALL	
3 IF/SELECT	7 Palletizing	
4 WAIT	8 --- next page ---	

PROGRAM1

3: L P[3] 500mm/sec CNT10

[End]

[INST] [EDCMD] >

Fig. A.2 (p) [2 EDIT]

Register screen

DATA Registers		JOINT 30%
		1/200
R [1:		=0
R [2:		=5
R [3:		=12
R [4:		=50000
R [5:		=0
R [6:		=0
R [7:		=0
R [8:		=0
R [9:		=0
R [10:		=0

Press ENTER

TYPE

Registers

Position register screen

DATA Position Reg		JOINT 30%
		1/100
PR[1:REF POS 1		=*
PR[2:REF POS 2		=*
PR[3:REF POS 3		=*
PR[4:REF POS 4		=*
PR[5:		=*
PR[6:		=*
PR[7:		=*
PR[8:		=*
PR[9:		=*
PR[10:		=*

Press Enter

TYPE

MOVE_TO RECORD POSITION CLEAR

Position Reg

Position register screen/ position data information screen

Position Detail		JOINT 30%
		1/100
PR[1]	UF:F UT:1	CONF:FUT 000
X:	1500.374 mm	W: 40.000 deg
Y:	-342.992 mm	P: 10.000 deg
Z	956.895 mm	R: 20.000 deg

DATA Position Reg		JOINT 30%
		1/100
PR[1:REF POS		= R
PR[2:		=*
PR[3:		=*
PR[4:		=*
PR[5:		=*

Enter value

CONF DONE [REPRE]

Fig. A.2 (q) [3 DATA]

Palletizing register screen

DATA: Pallet Regist		JOINT 30%
		1/32
PL[1:pallet work		]=[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]

Enter value

[TYPE]

Fig. A.2 (r) [3 DATA]

Robot axis status screen (Status 1 screen)

STATUS Axis		JOINT 30%	
		GRP[1]	
FLAG Bits 1/2		History	
J1:	0000000000000000 (0000000000000000)		
J2:	0000000000000000 (0000000000000000)		
J3:	0000000000000000 (0000000000000000)		
J4:	0000000000000000 (0000000000000000)		
TYPE		STATUS1 STATUS2 PULSE [UTIL]>	
Axis	MONITOR TRACKING DISTURB[UTIL]>		
	REG.DIS DUTY DIAG >		

Robot axis status screen (Torque monitor screen)

STATUS Axis		JOINT 30%	
		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
[TYPE]		MONITOR TRACKING DISTURB[UTIL]>	

Software version screen (Software version screen)

STATUS Version ID		JOINT 30%	
SOFTWARE		ID 1/19	
1	Handling Tool	7DA7/16	
2	S/W Serial No.	DEV_SW	
3	Controller ID	F00000	
4	Default; Personality (from FD)		
5	R-2000iB/165F	V7.70P/16	
6	Servo Code	12345678	
7	Cart. Mot. Parameter	V3.00	
8	Joint Mot. Parameter	V3.00	
9	DCS	V2.0.9	
10	Stop pattern	A	
[TYPE]		SOFTWARE CONFIG MOTOR SERVO >	
UPDATES ORDER FI			

Software version screen (Motor ID screen)

STATUS Version ID		JOINT 30%	
GR: AX: MOTOR ID AND INFO:		1/56	
1	1 ACaiSR30/3000 80A H1 DSP1-L		
2	1 2 ACaiSR30/3000 80A H2 DSP1-M		
3	1 3 ACaiSR30/3000 80A H3 DSP1-J		
4	1 4 ACaiS12/4000 40A H4 DSP1-K		
5	1 5 ACaiS12/4000 40A H5 DSP1-L		
6	1 6 ACaiS12/4000 40A H6 DSP1-M		
7	** ** *****		
8	** ** *****		
9	** ** *****		
10	** ** *****		
[TYPE]		SOFTWARE CONFIG MOTOR SERVO >	

Program Timer List Screen

PRG TIMER LISTING		JOINT 10%	
		1/10	
TIMER	count (s) [comment]		
1	3.20 (s) []		
2	0.00 (s) []		
3	0.00 (s) []		
4	0.00 (s) []		
5	0.00 (s) []		
6	0.00 (s) []		
7	0.00 (s) []		
8	0.00 (s) []		
9	0.00 (s) []		
TYPE		DETAIL	
Prg Timer			

Program Timer Detail Screen

PRG TIMER DETAIL		JOINT 10%	
		1/3	
Timer[1]			
Comment	: [*****]		
Count	: 4.01(sec)		
Local/Global	: LOCAL		
Start program	:		
[TEST2		]	
line	: 1		
Stop program	:		
[TEST2		]	
line	: 6		
[TYPE]		LISTING	

Fig. A.2 (s) [4 STATUS]

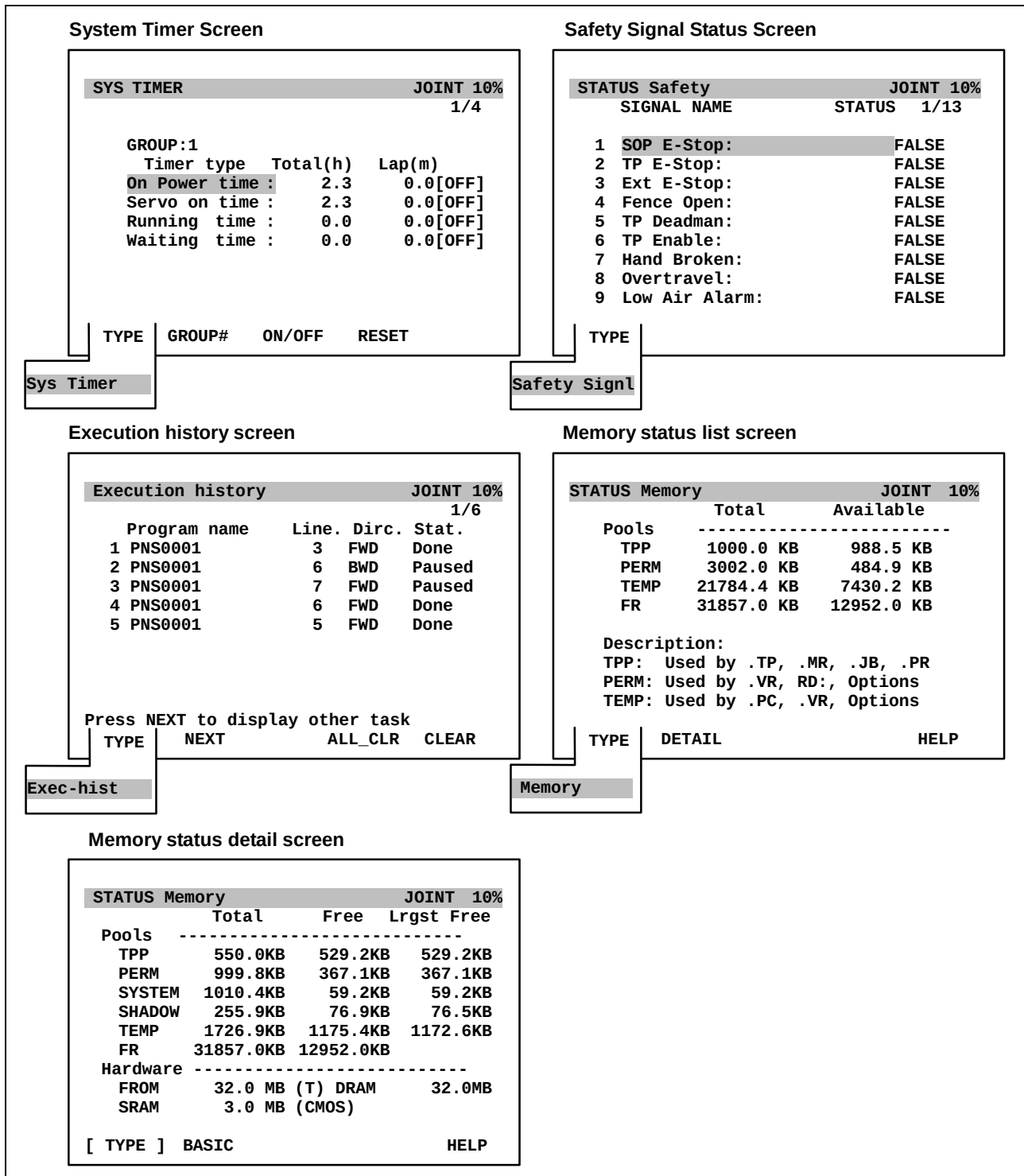


Fig. A.2 (t) [4 STATUS]

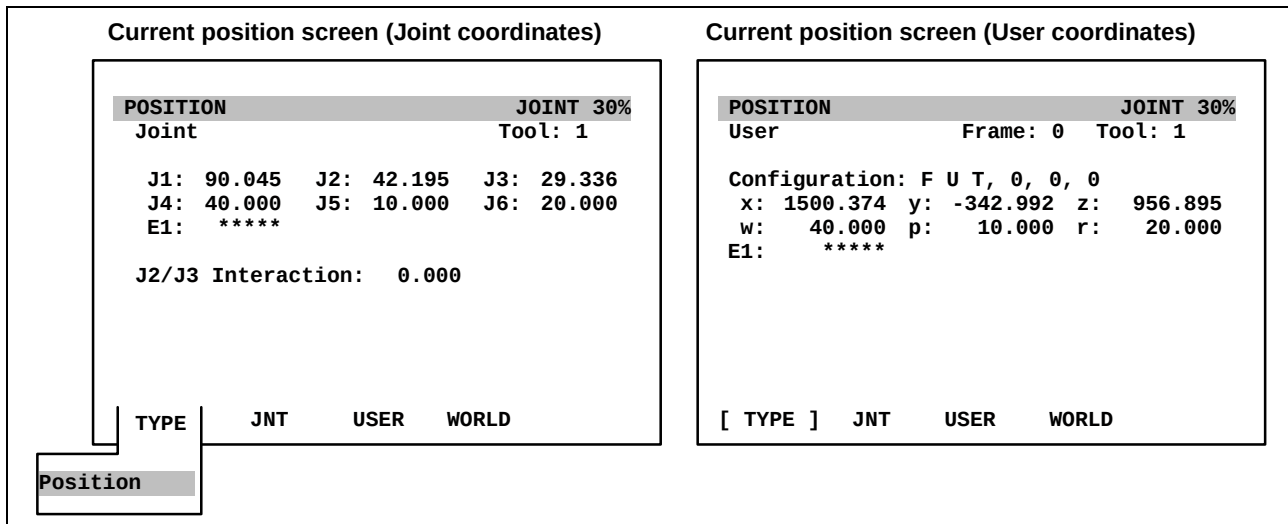


Fig. A.2 (u) [5 POSITION]

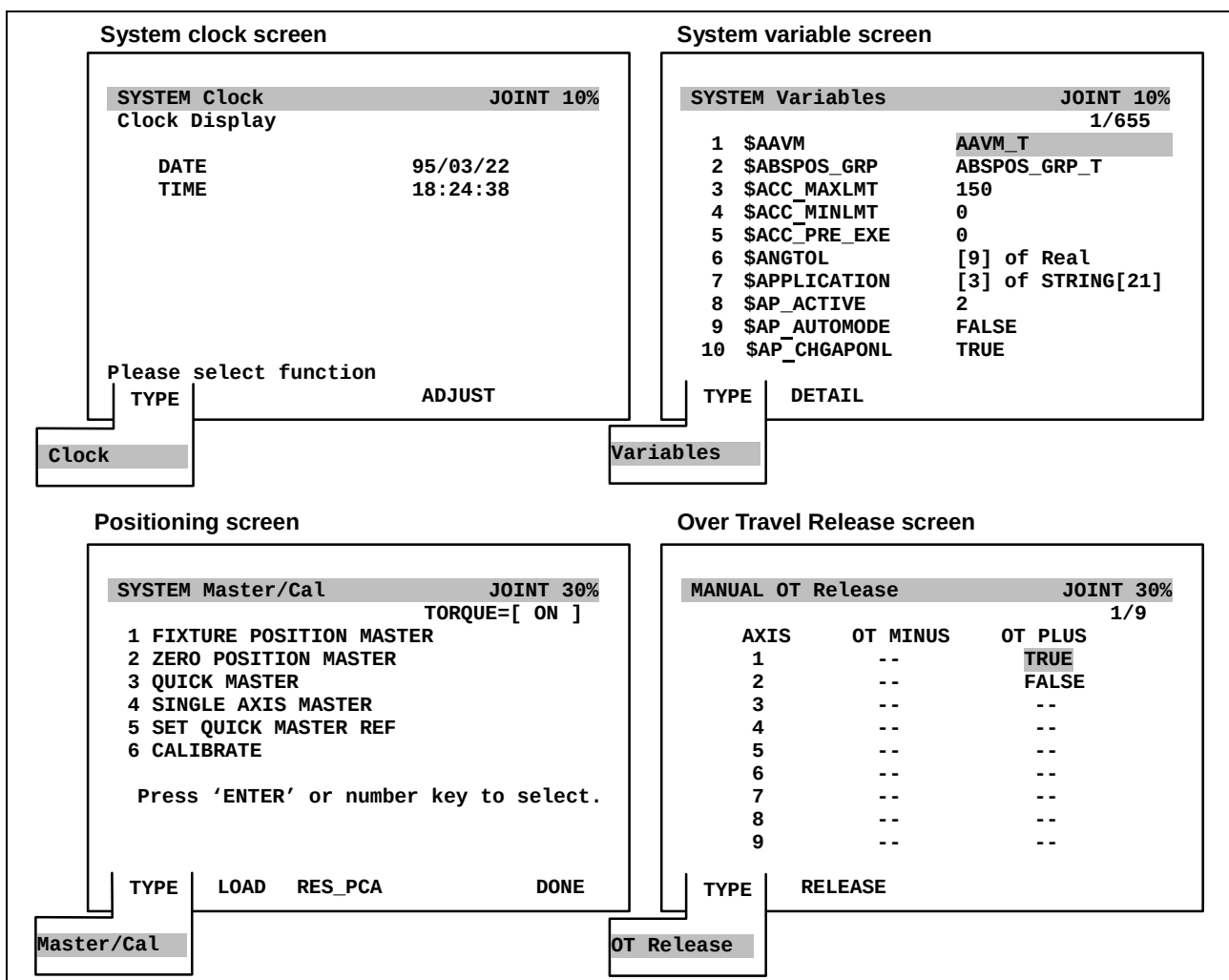


Fig. A.2 (v) [6 SYSTEM]

Joint operating area setting screen

SYSTEM Axis Limits			JOINT 10%	
AXIS	GROUP	LOWER	UPPER	1/56
1	1	-150.00	150.00	dg
2	1	-60.00	100.00	dg
3	1	-110.00	50.00	dg
4	1	-240.00	240.00	dg
5	1	-120.00	120.00	dg
6	0	-450.00	450.00	mm
7	0	0.00	0.00	mm
8	0	0.00	0.00	mm
9	0	0.00	0.00	mm
10	0	0.00	0.00	mm

TYPE

Axis Limits

Sysstem Configuration Screen

System/Config	Joint 30%
	1/48
1 Use HOT START:	FALSE
2 I/O power fail recovery:	RECOVER ALL
3 COLD START Autoexec program:	[*****]
4 HOT START Autoexec program:	[*****]
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 CSTPI:	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 Return to top of program:	TRUE
17 Original program name(F1):	[RSR]
18 Original program name(F2):	[PNS]
19 Original program name(F3):	[STYLE]
20 Original program name(F4):	[JOB]
21 Original program name(F5):	[TEST]
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 DOUT:	DO[0]
32 Signal to set in T1 mode DOUT:	DO[0]
33 Signal to set in T2 mode DOUT:	DO[0]
34 Signal to set if ESTOP DOUT:	DO[0]
35 Set if INPUT SIMULATED	DO[0]
36 Set if OUTPUT SIMULATED	DO[0]
37 Sim. Input Wait Delay	0.00 sec
38 Set if Sim. Skip Enabled	DO[0]
39 Set when prompt displayed	DO[0]
40 WAIT for DI range:	DI[0- 0]
41 WAIT for DI time:	0.00 sec
42 WAIT for DI output:	DO[0]
43 Signal if OVERRIDE = 100	DO[0]
44 Hand broken:	<*GROUPS*>
45 Remote/Local setup:	Remote
46 External I/O(ON:Remote):	DI[0]
47 UOP auto assignment:	FULL
48 Multi Program Selection:	FALSE

TYPE

TRUE FALSE

Config

Fig. A.2 (w) (Continued) [6 SYSTEM]

A.3 LIST OF PROGRAM INSTRUCTIONS

Motion instructions

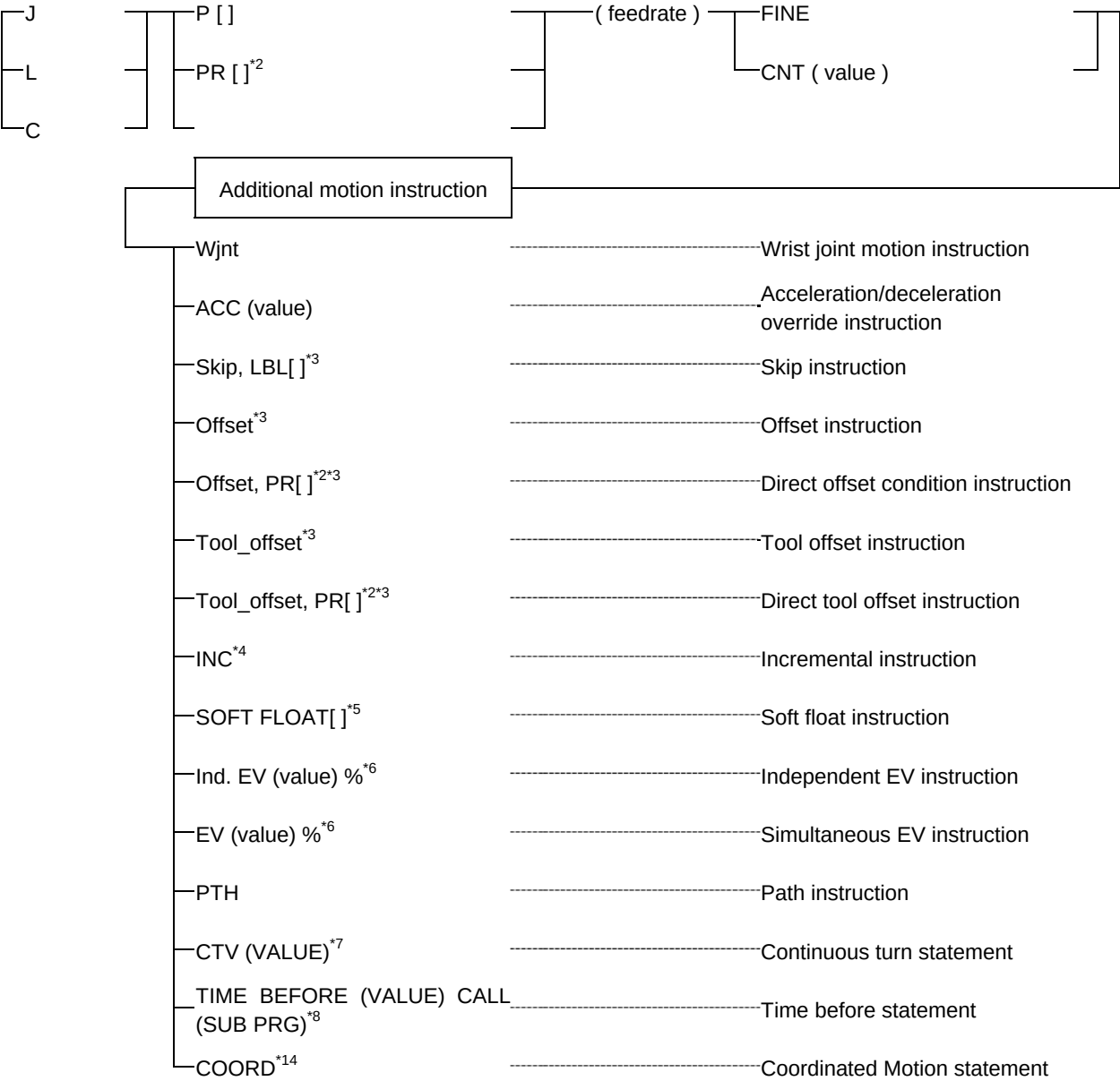


Fig. A.3 (a) Motion (option) instruction

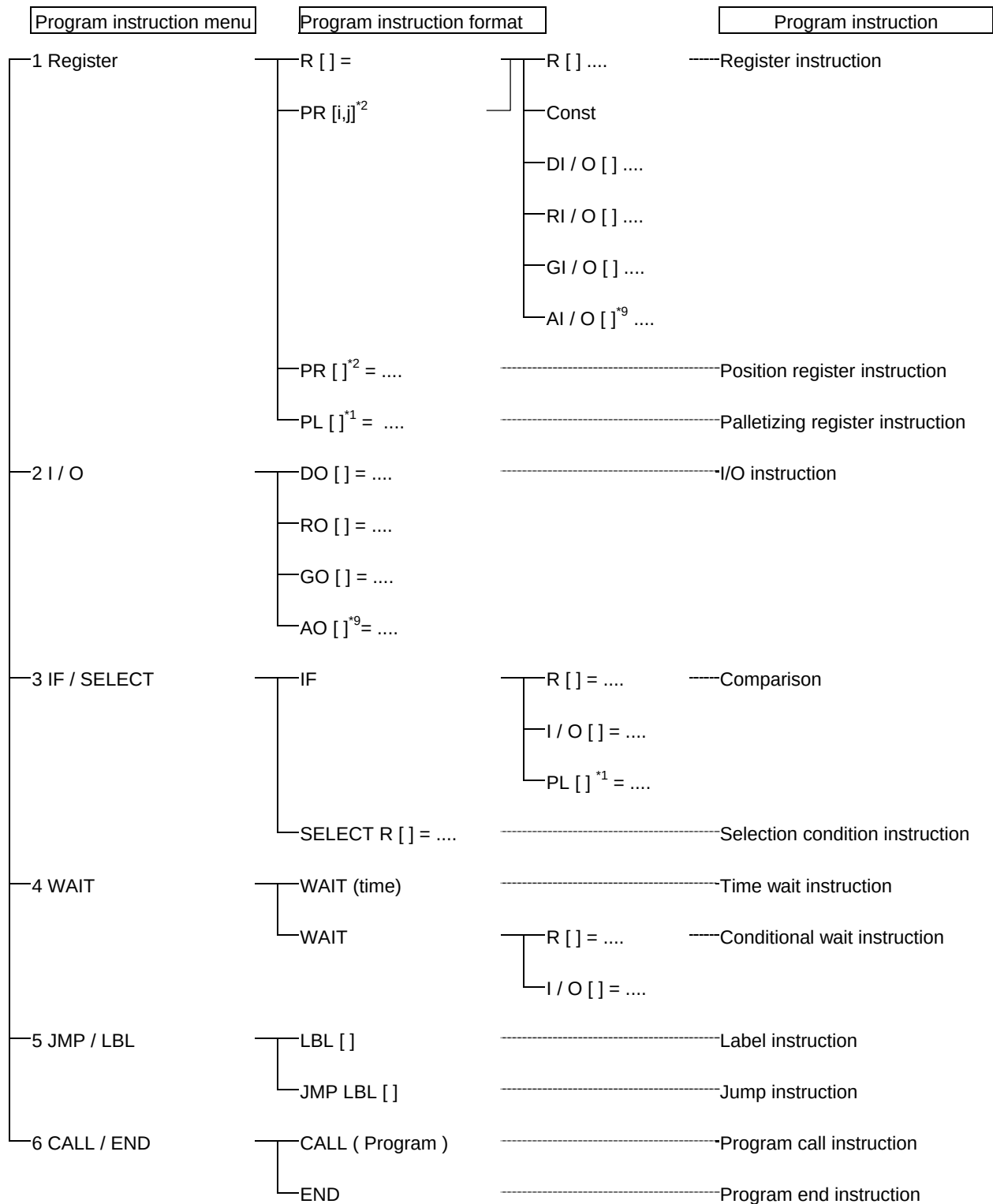


Fig. A.3 (b) Program instruction menu

Program instruction menu	Program instruction format	Program instruction
7 Palletizing ^{*1}	Palletizing – B	----Palletizing instruction
	PALLETIZING – BX	----Palletizing BX statement
	PALLETIZING – E	----Palletizing E statement
	PALLETIZING – EX	----Palletizing EX statement
	Palletizing – END	----Palletizing end instruction
8 Program Control	PAUSE	----Halt instruction
	ABORT	----Abort instruction
9 Miscellaneous	RSR [] = ...	----RSR statement
	UALM []	----User alarm instruction
	TIMER [] = ...	----Timer instruction
	OVERRIDE = ...	----Override instruction
	Remark	----Comment instruction
	Message []	----Message instruction
	\$ (Parameter) = ...	----Parameter instruction
10 Skip ^{*3}	SKIP CONDITION I/O [] = ...	____Skip condition instruction
11 Offset ^{*1}	OFFSET CONDITION PR [] ^{*2} = ...	----Offset condition instruction
	UFRAME [] = ...	----User frame setup instruction
	UFRAME_NUM = ...	____User frame selection instruction
	UTOOL [] = ...	----Tool frame setup instruction
	UTOOL_NUM = ...	____Tool frame selection instruction
12 Tool_Offset ^{*3}	TOOL_OFFSET CONDITION PR [] ^{*2} = ...	____Tool offset condition instruction
13 MACRO ^{*3}		-----Macro instruction
14 Multiple control ^{*10}	RUN (PRG)	----Run program statement
15 SENSOR ^{*11}	SEND R[]	----Send register statement
	RCV R[], LBL[]	----Receive register statement
16 LOCK PREG ^{*2}	LOCK PREG	____Position register lock instruction
	UNLOCK PREG	____Position register unlock instruction
17 SOFTFLOAT ^{*5}	SOFTFLOAT []	----Soft float start instruction

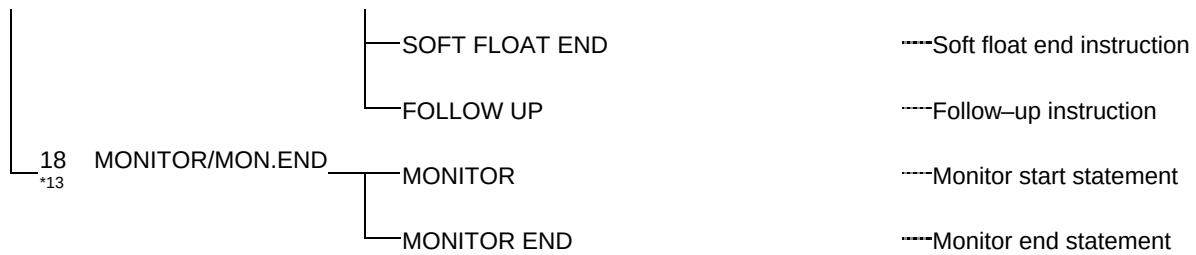


Fig. A.3 (c) Program instruction menu

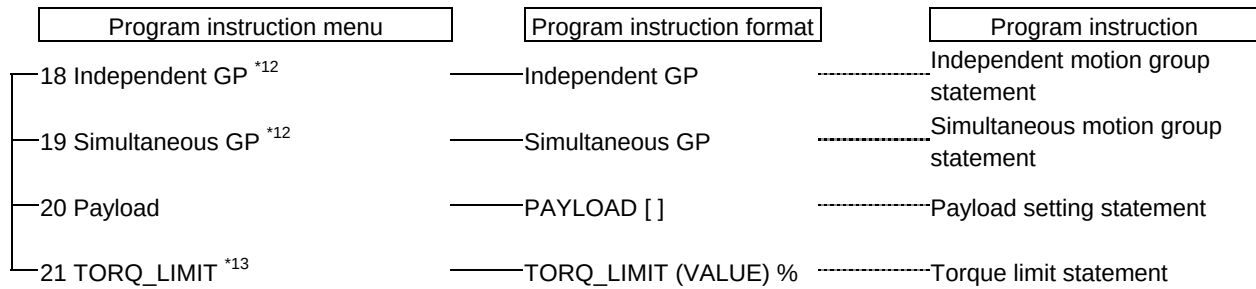


Fig. A.3 (d) Program instruction menu

Option

The items indicated by * (numeral) in Fig. A.3 (a) to A.3 (d) are displayed when the corresponding option is added, as indicated below:

Table A.3 Option list

*	Option	Specification
1	Palletizing	A05B-2500-J500
2	Position register	A05B-2500-H510
3	Option command	A05B-2500-H510
4	Incremental input	A05B-2500-J510
5	Softfloat	A05B-2500-J612
6	Extended axis control	A05B-2500-J518
7	Continuous turn	A05B-2500-J613
8	Condition monitor function	A05B-2500-H510
9	Analog I/O	A05B-2500-H510
10	Multi task	A05B-2500-H510
11	Sensor interface	A05B-2500-J502
12	Multi motion group	A05B-2500-J601
13	Torque limit	A05B-2500-J611
14	Coordinated motion package	A05B-2500-J686

A.4 PROGRAM INSTRUCTIONS

A.4.1 Motion Instructions

Table A.4.1 Motion instructions

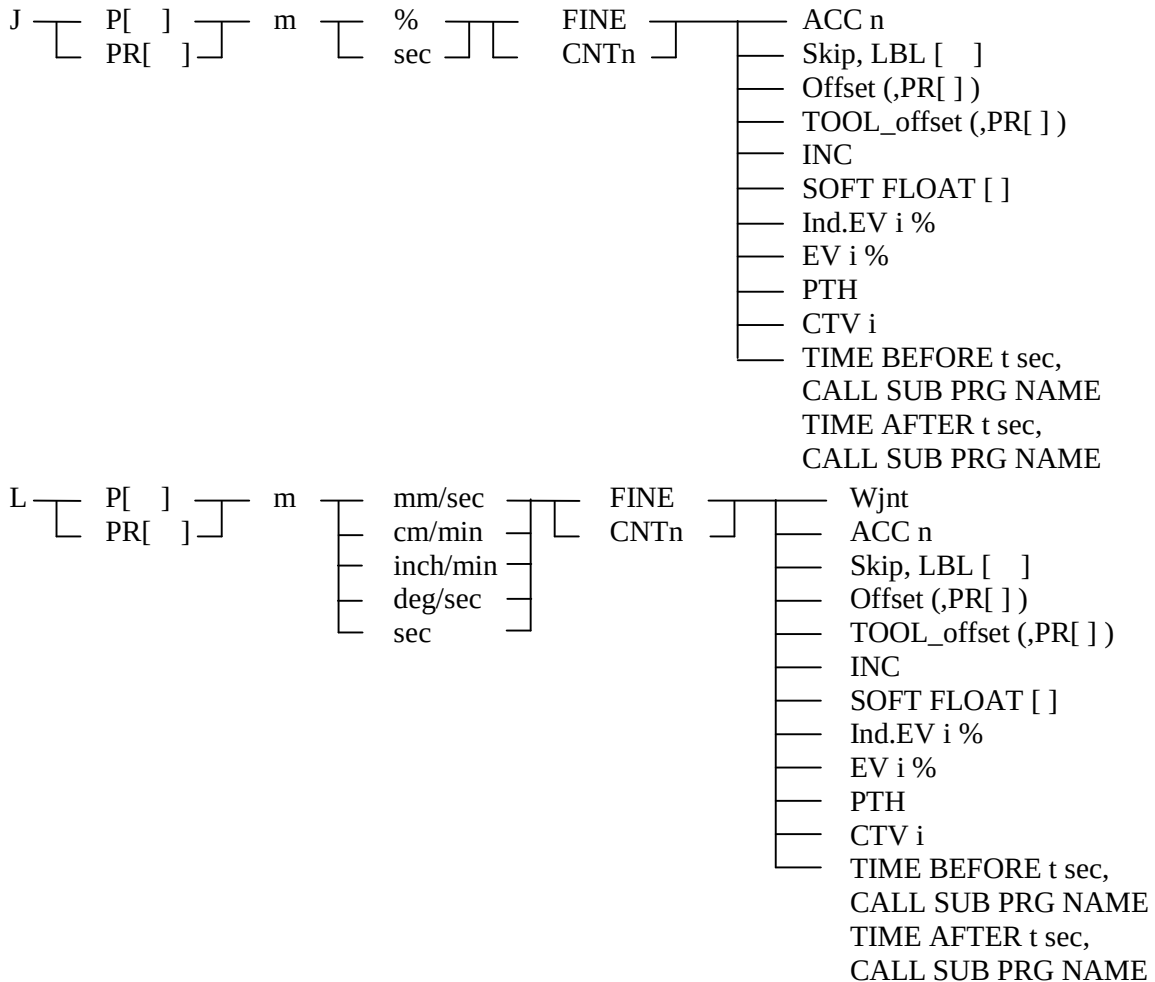
Motion format	J L C	Enables robot operation for each joint with interpolation. Moves the robot tool linearly. Enables the tool tip of the robot to make a circular motion.
Position variable	P[i : Comment] PR[i : Comment]	Standard variable for storing position data. Register for storing position data.
Feedrate unit	% mm/sec, cm/min, inch/min, deg/sec sec	Specify the rate of a feedrate to the highest feedrate of the robot. Specifies the speed with which the tool tip makes a linear or circular motion. Specify the time required during a motion.
Positioning path	FINE CNTn n (0 - 100):	The robot stops at the specified position and starts the next motion. The robot moves gradually from the specified position to the position at which the next motion starts. Degree of gradual motion. The higher the specified number, the more gradual the robot moves.

A.4.2 Additional Motion Instructions

Table A.4.2 Additional motion instructions

Wrist joint motion	Wjnt	On a linear or arc motion, the wrist axis moves with a joint motion, and the joint coordinates vary.
Acceleration/deceleration override	ACC a a=0 to 500(%)	Sets the rate of acceleration/deceleration when moving.
Skip	Skip, LBL[]	Causes a branch to the specified label when the condition specified in a skip condition instruction is not satisfied. When the condition is satisfied, cancels the motion and executes the next line.
Positional offset	Offset Offset,PR[(GPK:)]i	Makes the robot move to the position where the value specified by the offset condition instruction is added to the positional variable. Makes the robot move to the position where the value specified by the offset condition instruction and the value of position register are added to the positional variable.
Tool offset	Tool_offset Tool_offset,PR[(GPK:)]i	Moves the robot to the position corresponding to the value specified by the tool offset instruction, added to the position variable. Moves the robot to the position corresponding to the position register value, added to the position variable.
Incremental	INC	Makes the robot move to the position where the value of the position variable is added to the current position.
Soft float	SOFT FLOAT[i]	Enables the soft float function.
Independent EV	Ind.EV(i)% i = 1 to 100 (%)	Moves the extended axis, independently of the robot motion.
Simultaneous EV	EV(value)% i = 1 to 100 (%)	Moves the extended axis, synchronized with the robot.
Path	PTH	Creates a motion plan, using the rate attainable in continuous operation.
Continuous turn	CTV i i = -100 to 100(%)	Start the execution of continuous turn.

Before execution	TIME BEFORE t CALL <prog> TIME AFTER t CALL <prog>	Before or after the specified end time, call a sub program and execute one. t=Execution start time. <prog>=Name of sub program.



Example

```

1: J P[1] 100% FINE
2: L P[2:LINE] 500mm/sec CNT100 Wjnt
   : Offset, PR[1]
3: L P[3] 3.5sec CNT100 INC
4: L P[4] 100cm/min FINE Wjnt
   : Skip, LBL[100]
5: C P[5]
   : P[6] 300mm/sec CNT50

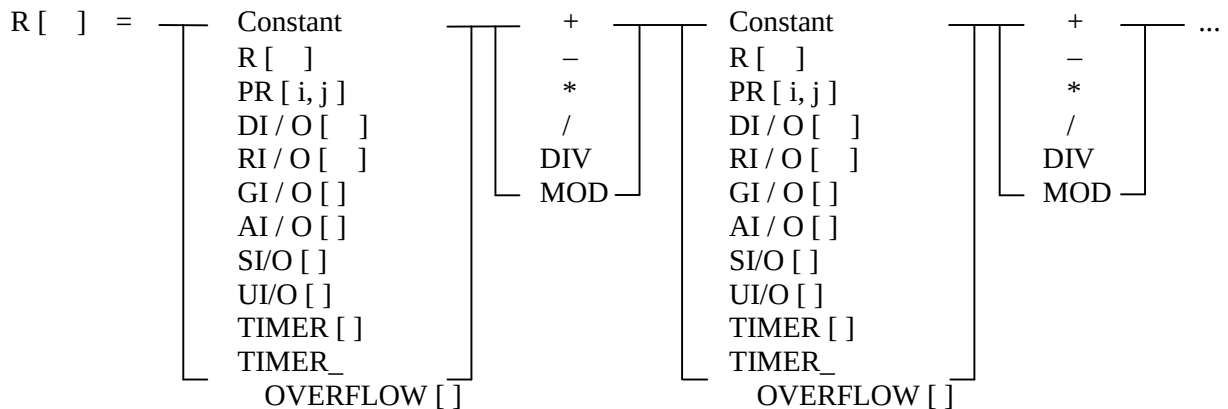
```

A.4.3 Register and I/O Instructions

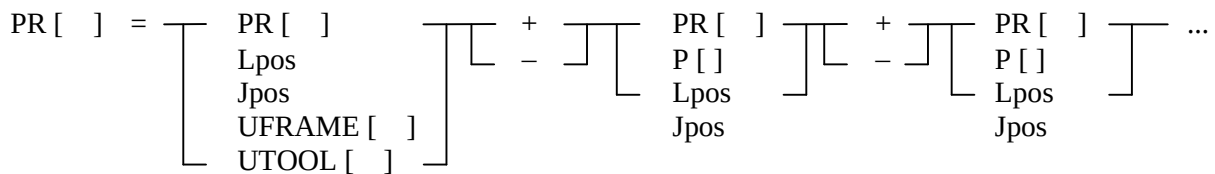
Table A.4.3 Register and I/O instructions

Register	R[i] i: 1 to 200	i: Register number.
Position register	PR[(GPK:) i] PR[(GPK:) i, j]	Fetches a position data element. GPK: Group number. k: 1 to 8 i: Position register number. i: 1 to 100 j: Number of an element in a position register. j: 1 to 9.

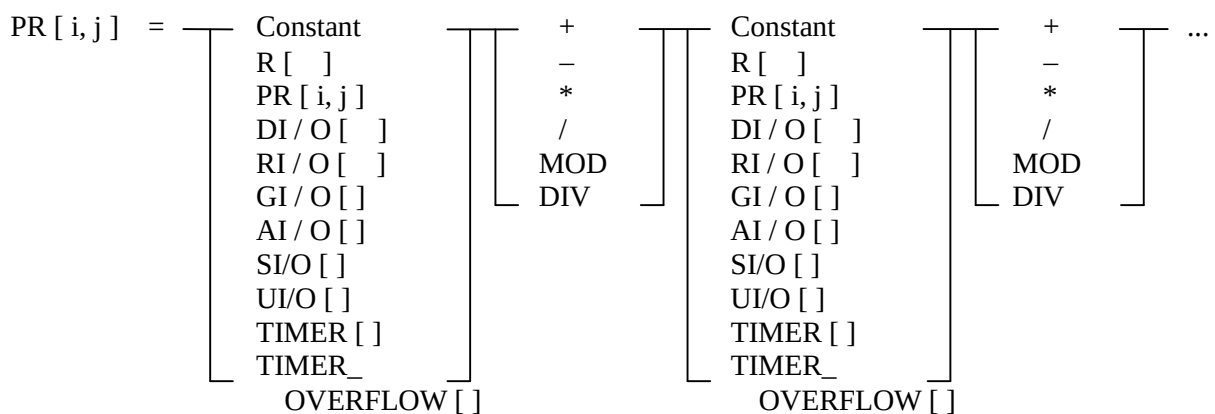
Position data	P[i :comment] Lpos Jpos UFRAME [i] UTOOL [i]	i: Position number. i: 1 to memory limit Cartesian coordinates of the current position Joint coordinates of the current position User coordinate system Tool coordinate system
Input/output signal	DI[i], DO[i] RI[i], RO[i] GI[i], GO[i] AI[i], AO[i]	System digital signals Robot digital signals Gourp signals Analog signals

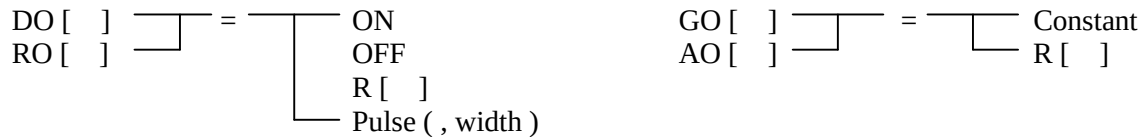


Example 1: R[1] = RI[3]
 2: R[3:] = DI[4]*PR[1,2]



Example 1: PR[1] = PR[6]
 2: PR[3] = PR[4]+Lpos
 3: PR[8] = UFRAME[1]
 4: PR[9] = UTOOL[2]





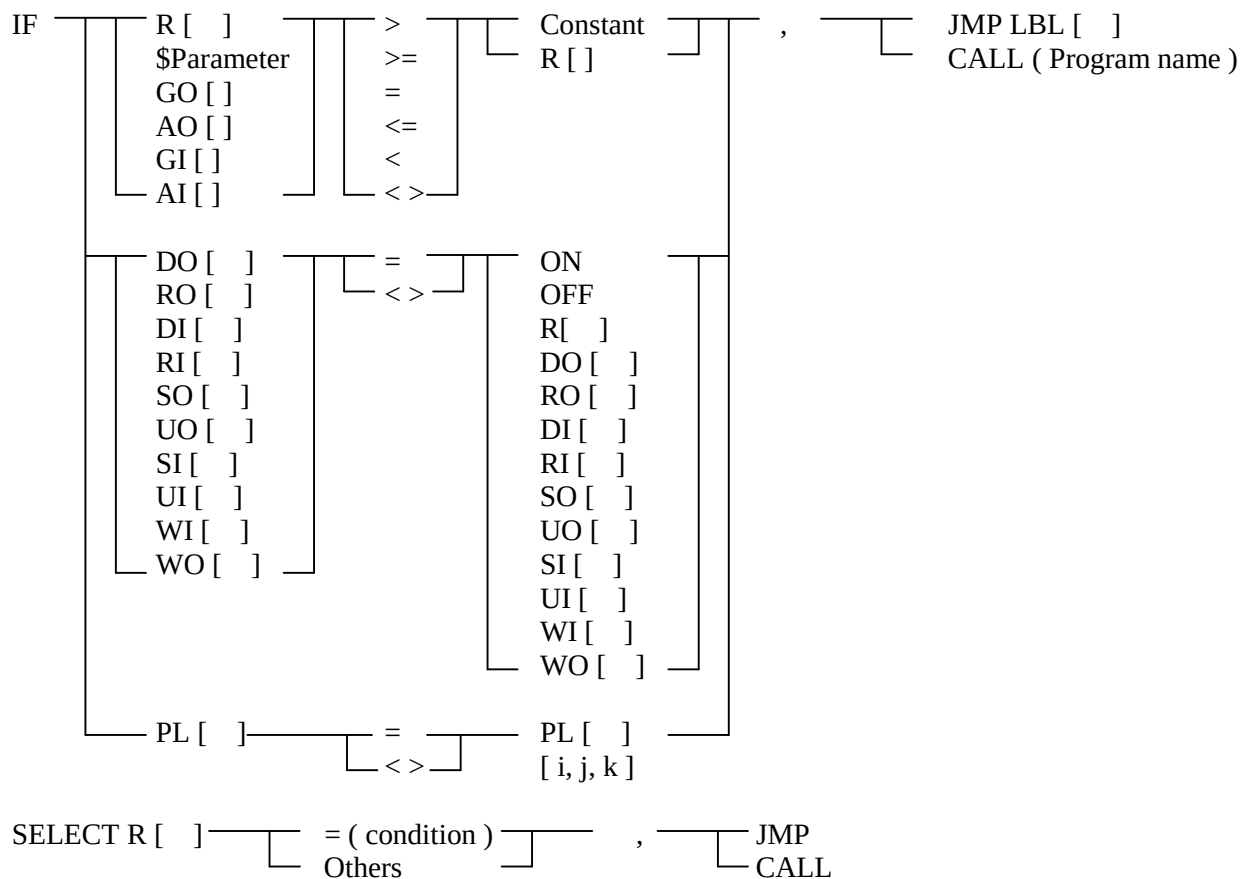
Example

- 1: DO[1] = ON
- 2: RO[3:] = PULSE,1.0sec
- 3: RO[4] = R[1]
- 4: GO[9] = R[2:control]
- 5: AO[10] = 12.5

A.4.4 Conditional Branch Instructions

Table A.4.4 Conditional branch instructions

Comparison condition	IF (condition) (branch)	Specifies a comparison condition and an instruction or program to which the program branches to. You can link (Conditions) by using operators.
Selection condition	SELECT R[i] = (value) (branch)	Specifies a selection condition and an instruction or program to which the program branches to.



Example

- 1: IF R[2] >= R[3],LBL[1:HANDOPEN]
- 2: IF DI[2] = ON,CALL SUBPROGRAM

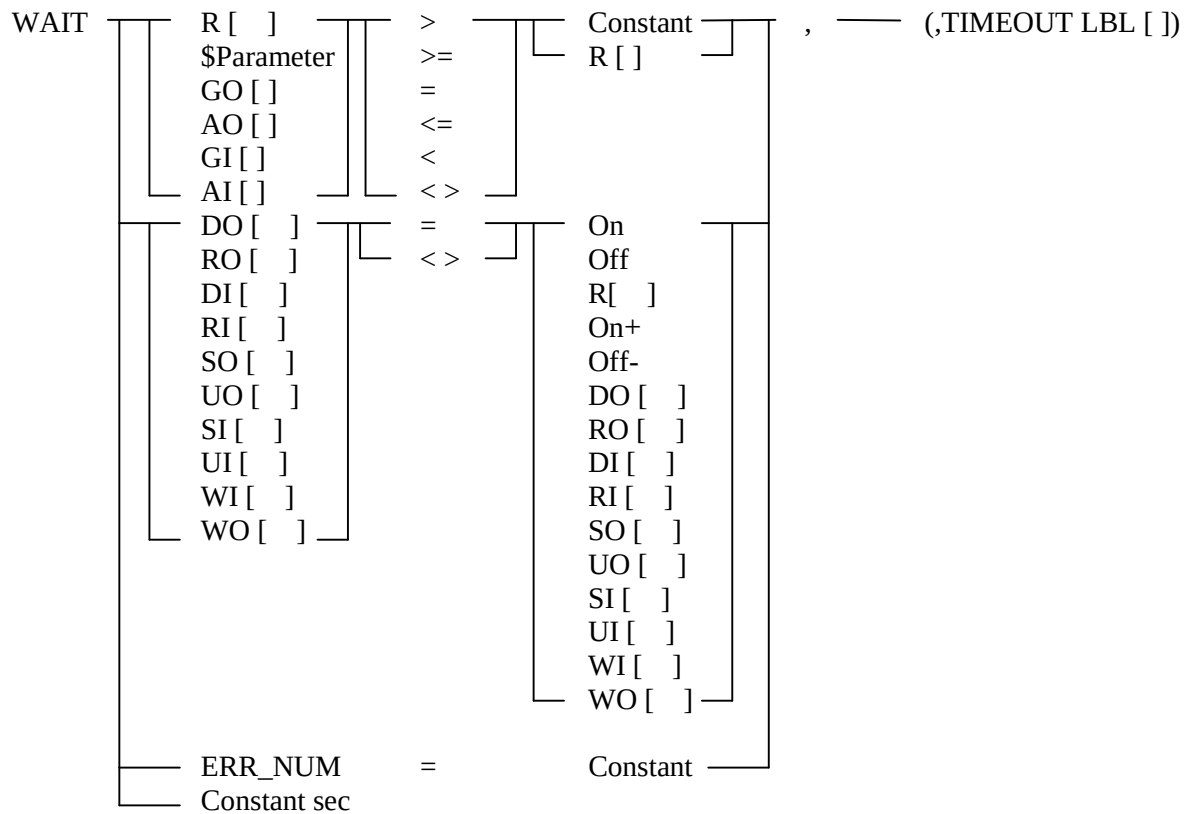
Example

- 3: SELECT R[2] = 1,JMP LBL[1]
- 4: = 2,JMP LBL[2]
- 5: = 3,JMP LBL[3]
-

A.4.5 Wait Instruction

Table A.4.5 Wait instruction

Wait	WAIT < condition > WAIT < time >	Waits until the specified condition is satisfied or until the specified time has elapsed. You can link (Conditions) by using operators.



Example

- 1: WAIT RI[1] = ON
- 2: WAIT 10.5sec
- 3: WAIT R[2],TIMEOUT,LBL[1]

A.4.6 Unconditional Branch Instructions

Table A.4.6 Unconditional branch instructions

Label	LBL [i : COMMENT] JMP LBL[i]	Specifies the instruction which the program branches to. Causes a branch to the specified label.
Program call	CALL (program-name)	Causes a branch to the specified program.
Program end	END	Ends the program and returns control to the calling program.

JMP LBL[]

CALL (Program name) LBL[]

- Example
- 1: LBL[1: HANDCLOSE]
 - 2: JMP LBL[2]
 - 3: CALL SAMPLE1
 - 4: CALL PRG2: LBL[1]

A.4.7 Program Control Instructions

Table A.4.7 Program control instructions

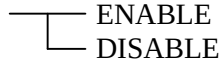
Halt	PAUSE	Halts a program.
Abort	ABORT	Aborts a program.

Example 1: PAUSE
2: ABORT

A.4.8 Other Instructions

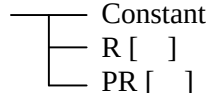
Table A.4.8 Other instructions

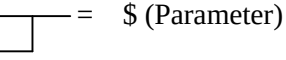
RSR	RSR[i]	Enables or disables RSR signals (i = 1 to 4).
User alarm	UALM[i]	Displays a user alarm on the alarm line.
Timer	TIMER[i]	Sets the timer.
Override	OVERRIDE	Sets override.
Comment	!(comment)	Inserts a comment in a program.
Message	MESSAGE [message-text]	Displays a user message on a user screen.
Parameter	\$(system variable name)	Changes the value of a system variable.
Maximum speed	JOINT_MAX_SPEED [] LINEAR_MAX_SPEED	Sets the maximum speed for operation statements in the program.

RSR [] = 

TIMER [] = 

OVERRIDE = m%

\$(system variable name) = 

R [] 
PR [] 

JOINT_MAX_SPEED [] = 

LINEAR_MAX_SPEED [] = 

Example

```

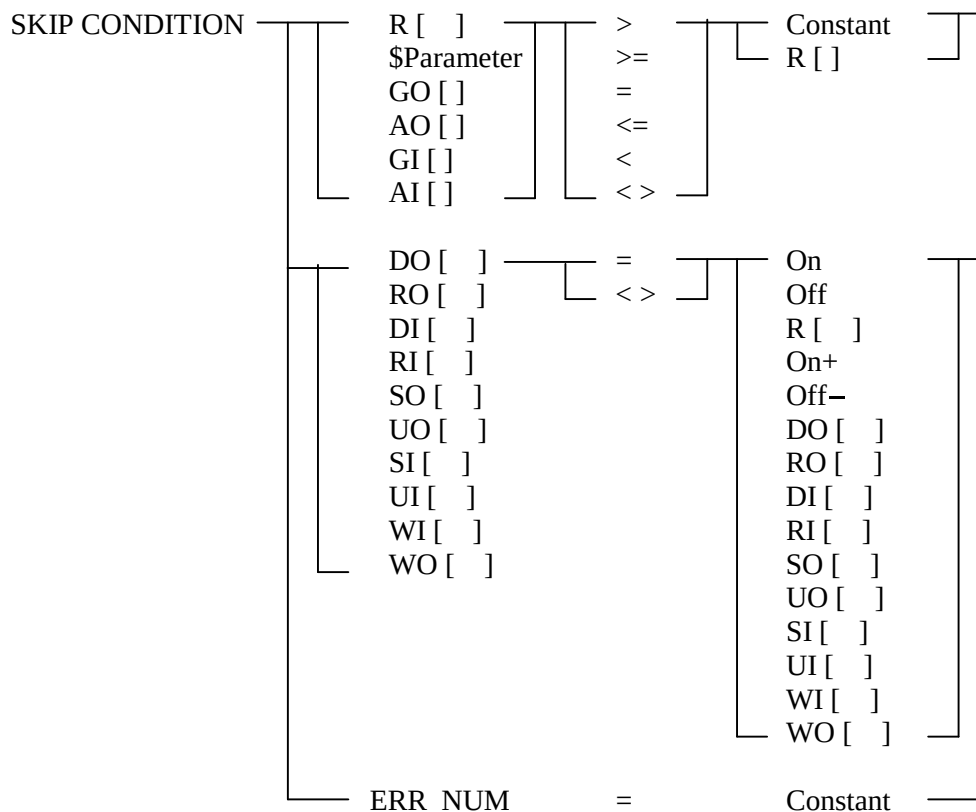
3: OVERRIDE = 50%
4: $NRM_TURN = 1
5: $NRM_TURN = 1
6: RSR[2] = ENABLE
7: UALM [3:no work]
8: ! STEP 2 START
9: TIMER[1] = START
10: MESSAGE [STEP1 EXECUTION]
11: $NRM_TURN = 1

```

A.4.9 Skip and Offset Condition Instruction

Table A.4.9 Skip and Offset condition instruction

Skip condition	SKIP CONDITION (condition)	Specifies the skip execution condition for an additional motion instruction. You can link (Conditions) by using operators.
Offset condition	OFFSET CONDITION (offset amount)	Specifies the amount of offset used by the motion instruction.
Tool offset condition	TOOL_OFFSET CONDITION (offset amount)	Specifies the amount of tool offset used by the motion instruction.



OFFSET CONDITION PR[](, UFRAME[])

TOOL OFFSET CONDITION PR[](, UTOOL[])

- Example
- 1: SKIP CONDITION DI[1] = ON
 - 2: SKIP CONDITION RI[2] <> DI[3]
 - 3: OFFSET CONDITION PR[1],UFRAME[1]
 - 4: TOOL OFFSET CONDITION PR[2],UTOOL[1]

A.4.10 Frame Setup Instruction

Table A.4.10 Frame setup instruction

User frame	UFRAME[i]	User frame i=1 to 9
User frame selection	UFRAME_NUM	The number of current user frame.
Tool frame	UTOOL[i]	Tool frame. i=1 to 9
Tool frame selection	UTOOL_NUM	The number of current tool frame

UFRAME[]  =  P []
 UTOOL[]  =  PR[]
 UFRAME_NUM  =  Constant
 UTOOL_NUM  =  R[]

Example 1: UFRAME[1] = P[12]
 2: UTOOL[3]=PR[1]
 3: UFRAME_NUM=3

A.4.11 Macro Instruction

Table A.4.11 Macro instruction

Macro	(macro-instruction)	Executes a program defined on the macro instruction setting screen.

Example 1: HAND1 OPEN
 2: HAND2 CLOSE

A.4.12 Multiaxis Control Instructions

Table A.4.12 Multiaxis Control Instructions

Program execution	RUN	Starts execution of a specified program in another motion group.

RUN (Program Name)

Example **PROGRAM 1** **PROGRAM 2**
 1: R[1] = 0 1: J P[3] 100% FINE
 2: RUN PRG2 2: J P[4] 100% FINE
 3: J P[1] 100% FINE 3: J P[5] 100% FINE
 4: J P[2] 100% FINE 4: J P[6] 100% FINE
 5: WAIT R[1] = 0 5: R[1] = 1

Group Mask[1, *, *, *, *] Group Mask[*, 1, *, *, *]

A.4.13 Position Register Look-ahead Execution Instruction

Table A.4.13 Position register look-ahead execution instruction

Position register lock	LOCK PREG	Locks a position register to prevent the register contents from being changed.
Position register unlock	UNLOCK PREG	Unlocks a position register.

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

A.4.14 Soft Float Instruction

Table A.4.14 Soft float instruction

Soft float start	SOFTFLOAT[i]	Enables the soft float function.
Soft float end	SOFTFLOAT END	Disables the soft float function.
Follow-up	FOLLOW UP	Assumes the current robot position to be the taught position (follow-up) when the soft float function is used.

Example

```

1: J P[1] 100% FINE
2: SOFTFLOAT[1]
3: L P[2] 100mm/sec FINE
4: FOLLOWUP
5: L P[3] 100mm/sec FINE
6: SOFTFLOAT END

```

A.4.15 Status Monitoring Instructions

Table A.4.15 Status Monitoring Instructions

Status monitoring start instruction	MONITOR<Condition program name>	Starts monitoring under the conditions specified in the condition program.
Status monitoring end instruction	MONITOR END <Condition program name>	Ends monitoring under the conditions specified in the condition program.

Example

```

1: MONITOR WRKFALL
2: J P[1] 100% FINE
.
.
.
8: J P[7] 100% FINE
9: MONITOR END WRKFALL
10: OPEN HAND

```

A.4.16 Motion Group Instructions

Table A.4.16 Motion Group Instructions

Independent motion group	Independent GP	Enables motion groups to operate independently of each other.
Simultaneous motion group	Simultaneous GP	Enables motion groups to operate simultaneously with the motion group that requires the longest travel time.

Example

```

1: Independent GP
: GP1 L P[1] 90mm/sec CNT100
: GP2 J P[1] 100% CNT50
2: Simultaneous GP
: GP1 L P[2] 150mm/sec CNT20
: CP2 J P[2] 70% CNT50

```

A.4.17 Palletizing Instructions

Table A.4.17 Frame setup instruction

Palletizing	PALLETIZING-B_i i: Palletizing number	Calculates palletizing.
Palletizing end	PALLETIZING-END_i i: 1 to 16	Increases or decreases the contents of a palletizing register.
Palletizing motion	L PAL_i[A-j] 300mm/s FINE	Specifies the palletizing position. i: Palletizing number (1 - 16). j: Number of the approach point.

```

PALLETIZING B_i
              Bx_i
              E_i
              Ex_i
              END_I

```

J	PAL_i[]	m	%	FINE
L			mm/sec	CNTn
			cm/min	
			deg/sec	
			sec	

Example 1: PALLETIZING-B_3

2:	J	PAL_3[A_2]	50%	CNT50
3:	L	PAL_3[A_1]	100mm/sec	CNT10
4:	L	PAL_3[BTM]	50mm/sec	FINE
5:		HAND1	OPEN	
6:	L	PAL_3[R_1]	100mm/sec	CNT10
7:	J	PAL_3[R_2]	50%	CNT50
8:		PALLETIZING-END_3		

$$\text{PL}[i, j, k] = \text{PL}[i, j, k] \text{ (with } i \text{ and } j \text{ swapped)} + \text{PL}[i, j, k] \text{ (with } i \text{ and } k \text{ swapped)} + \text{PL}[i, j, k] \text{ (with } j \text{ and } k \text{ swapped)} + \dots$$

Example 1: $PL[1] = PL[3] + [1, 2, 1]$
 2: $PL[2] = [1, 2, 1] + [1, R[1], *]$

B SPECIAL OPERATION

This appendix summaries items necessary for using this model. It may also be used as an index.

Contents of this appendix

- B.1 START MODE
- B.2 MASTERING
- B.3 SOFTWARE VERSION
- B.4 ROBOT AXIS STATUS
- B.5 DIAGNOSIS SCREEN
- B.6 WORLD FRAME ORIGIN
- B.7 FSSB LINE SETUP
- B.8 POSITIONER SETUP
- B.9 EXTENDED AXIS SETUP
- B.10 INDEPENDENT ADDITIONAL AXIS SETUP

B.1 START MODE

B.1.1 Start Up Methods

Robot controller has the following four start up methods(start mode):

Initial start

When the unit is started in the initial start mode, all programs are deleted, and all settings are reset to their standard values. Upon the completion of the initial start, a controlled start is performed automatically.

Controlled start

When the unit is started in the controlled start mode, a controlled start menu, which is a simple system, starts up. The controlled start menu cannot be used to operate the robot. The controlled start menu can, however, be used to change a system variable which normally cannot be changed, to read a system file, and to set the robot. From the menu displayed by pressing the Fctn key on the controlled start menu, a cold start can be made.

Cold start

The cold start mode is used to perform normal power-up while power restoration is disabled. The program is aborted, and all output signals are turned off. Once the cold start has been completed, the robot can be operated.

A cold start can be performed while power restoration is enabled, provided the necessary setting is made at power-up.

Hot start

The hot start mode is used to perform normal power-up while power restoration is enabled. When the unit starts, the program runs and output signals are restored to the state existing prior to the last power-down. Once the hot start has been completed, the robot can be operated.

The cold start or the hot start is started in usually operation. Which mode is used depends on whether the hot start is enable or disable.

The initial start and the controlled start will be used when maintaining it. These modes will not be used in usually operation.

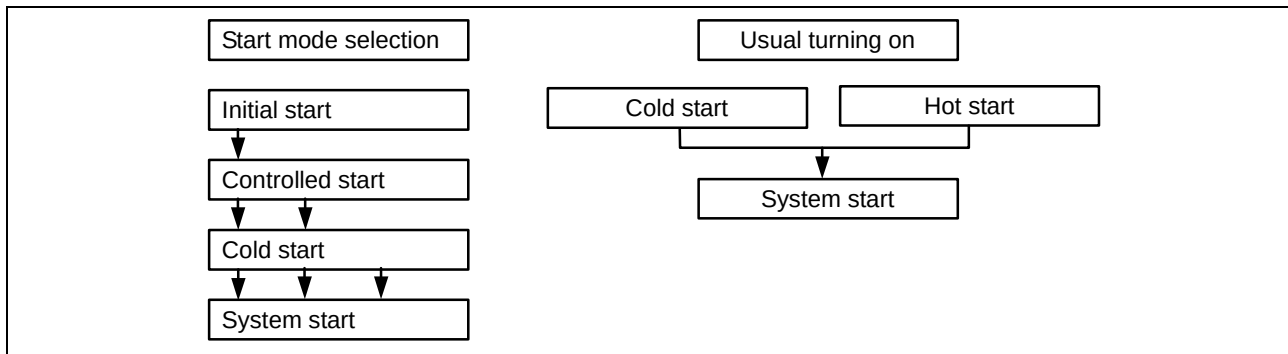


Fig. B.1.1 Start mode

B.1.2 Initial Start

When the unit is started in the initial start mode, all programs are deleted, and all the settings are reset to their standard values.

Once the initial start has been completed, a controlled start is performed automatically.

⚠ CAUTION

At an initial start, programs and all data including settings will be lost. The factory-set mastering data is also erased. The initial start should be made only when the main printed circuit board or software is replaced. Before performing an initial start, therefore, make a backup copy of the necessary programs and system files.

⚠ CAUTION

Before 7DA3, backup restore might be needed two times when backup is loaded after initial start. When max number of item is changed, this procedure is needed. (Example: max number of register or user alarm is changed by Program Limits screen.)

In this case, please load backup as following procedure.

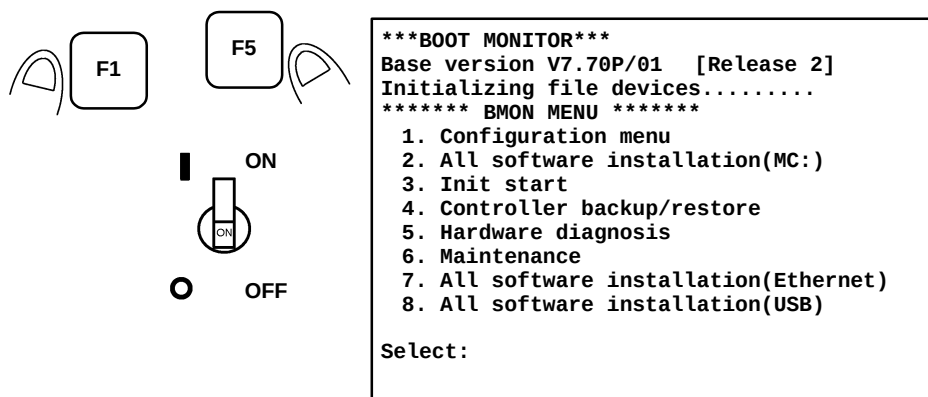
1. Perform control start and load backup files by file menu.
2. Perform cold start.
3. Perform control start and load backup files again.

In 7DA4 or later, backup files don't need to load two times.

Procedure B-1 Initial start

Step

- 1 With the **[F1]** key and the **[F5]** key held down on the teach pendant, set the power breaker on the controller to ON.



- 2 Select 3, Init start.



- 3 Enter 1, (YES) to the confirmation message of initial start.

CAUTION:INIT start is selected
 Are you SURE? [Y=1 / N=else]

An initial start is performed. Upon the completion of the initial start, a controlled start is performed automatically, and the controlled start menu appears.

B.1.3 Controlled Start

When the unit is started in the controlled start mode, a controlled start menu, which is a simple system, starts up. The controlled start menu cannot be used to operate the robot. The controlled start menu can, however, be used to change a system variable which normally cannot be changed, to read a system file, and to set the robot.

Press the Fctn key on the controlled start menu. A menu appears. From that menu, select 1 START (COLD). A cold start is performed.

The following screens can be displayed from the menu displayed by pressing the MENU key on the controlled start menu:

Setting screens

Settings can be made.

System variables screen

System variables can be set. Even a system variable which cannot normally be changed (R0) can be changed.

File screen

A program or system file can be saved and read. The system file can be read only from the controlled start menu.

On the file screen of the controlled start menu, F4 is displayed as [RESTORE]. When the F4 key is pressed, all files are read automatically. To switch F4 to [BACKUP] as on other file screens, press the Fctn key. A menu appears. From that menu, select RESTORE/BACKUP.

Version ID Screen

The software edition is displayed.

Alarm history screen

The alarm history is displayed.

Port init screen

A serial port is set.

This screen is used to read a file upon a controlled start.

Memory screen

The memory status is displayed.

MAINTENANCE

A robot setting can be changed. An additional axis can be set. A motion group can be added or deleted.

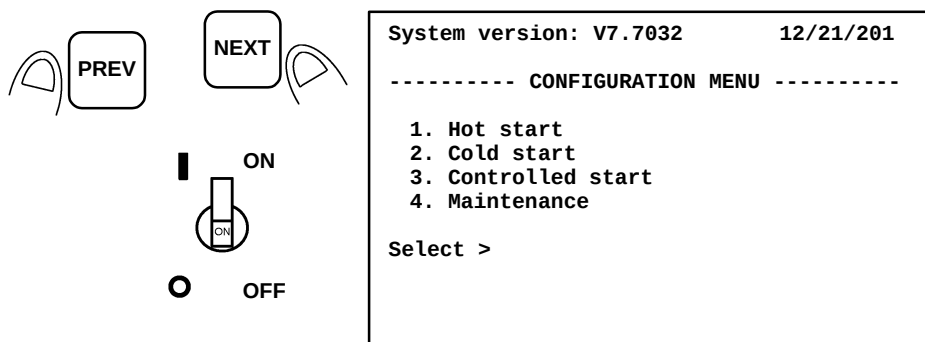
Max. number setting screen

The numbers of resistor, macro, and user alarm can be changed.

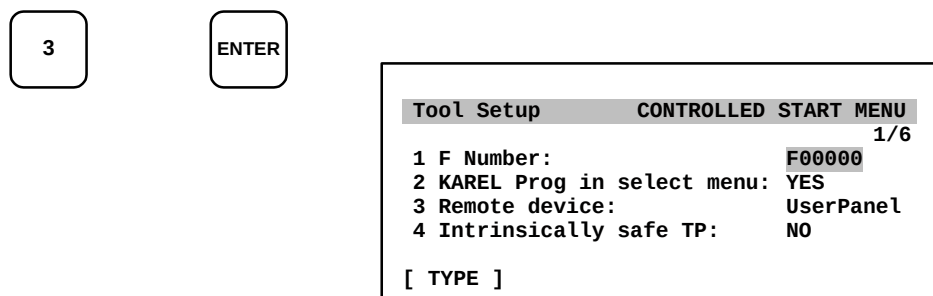
Procedure B-2 Controlled start

Step

- 1 With the **PREV** key and the **NEXT** key held down on the teach pendant, set the power breaker on the controller to ON.



- 2 Select 3 CONTROLLED START. The setting screen for the controlled start menu appears.



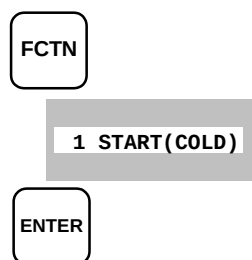
⚠ CAUTION

Please verify the rest of FROM is more than 600KB when you need to do the controlled start. (You can verify the available FROM in memory status menu [select menu →0.Next→4.status→memory])

If the rest of FROM is less than 600KB, please remove the unnecessary data from FR: or FRA: device to make room in FROM before you do the controlled start.

Also you cannot select controlled start in configuration menu if the rest of FROM is less than 600KB in case system software version is 7DA1/46 or later, 7DA3/37 or later, 7DA4/22 or later, 7DA5/16 or later, 7DA7/12 or later.

- 3 To operate the robot, a cold start must be performed. To do this, press the Fctn key. A menu appears. From that menu, select 1 START (COLD). A cold start is performed.



B.1.4 Cold Start

The cold start mode is used when normal power-up is performed while power restoration is disabled.

Upon a cold start, the following is performed:

- Each output signal of digital I/O, analog I/O, robot I/O, and group I/O is turned off or set to 0.
- The program is aborted, and the beginning of the program becomes the current line.
- The feedrate override is reset to the initial value.
- The manual feed coordinate system enters the JOINT state.
- The machine lock is released.

The cold start procedure depends on the power restoration setting.

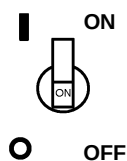
Procedure B-3 Cold start

Condition

- Hot start must be set to invalid.

Step

- 1 Turn on the power breaker on the controller to ON. The following screen is displayed after the system starts by cold start.



UTILITIES Hints

JOINT 10%

HandlingTool

V7.70P/13

7DA7/13

Copyright 2010, All Rights Reserved
 FANUC LTD, FANUC Robotics America, Inc.
 Licensed Software: Your use constitutes
 your acceptance. This product protected
 by several U.S. patents.

[TYPE] LICENSE PATENTS

HELP

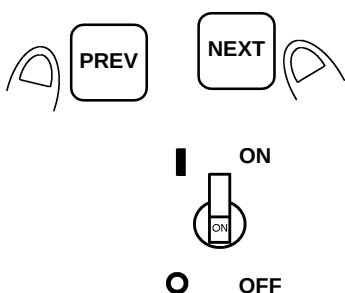
Procedure B-4 Cold start

Condition

- Hot start is set to disable

Step

- With the **PREV** key and the **NEXT** key held down on the teach pendant, set the power breaker on the controller to ON. The configuration menu is displayed.



System version: V7.7032

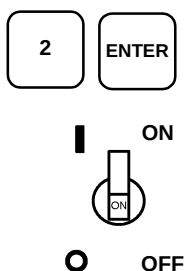
12/21/201

----- CONFIGURATION MENU -----

1. Hot start
2. Cold start
3. Controlled start
4. Maintenance

Select >

- Select "2. Cold start". Cold start is executed and following is displayed.



UTILITIES Hints

JOINT 10%

HandlingTool

V7.70P/13

7DA7/13

Copyright 2010, All Rights Reserved
 FANUC LTD, FANUC Robotics America, Inc.
 Licensed Software: Your use constitutes
 your acceptance. This product protected
 by several U.S. patents.

[TYPE] LICENSE PATENTS

HELP

B.1.5 Hot Start

The hot start mode is used when normal power-up is performed while power restoration is enabled. Upon a hot start, the following is performed:

- Each output signal of digital I/O, analog I/O, robot I/O, and group I/O is set in the same manner as it was prior to the last power-down.

- The program runs in the same way as it did prior to the last power-down. If the program was running up until the last power-down, the program enters the pause state.
- The feedrate override, manual feed coordinate system, and machine lock are set in the same manner as they were prior to the last power-down.

CAUTION

When a hot start is performed in the following state, each output signal of the digital I/O, analog I/O, robot I/O, and group I/O is turned off or set to 0:

- When the I/O allocation is changed
- When an I/O unit is mounted or removed
- When the number of signals is changed on the I/O Link screen

Procedure B-5 Hot start

Condition

- Hot start must be set to enable.

Step

- 1 Turn on the power breaker on the controller to ON. The screen which was being displayed at power off will be displayed on the screen of the teach pendant after a few seconds.

B.2 MASTERING

Mastering associates the angle of each axis of the robot with the pulse count of the absolute pulsecoder (APC) connected to the motor of each axis. More specifically, mastering is performed by obtaining the pulse count read at the zero-degree position.

The current position of the robot is determined by the pulse counts of the absolute pulsecoders (APCs) for the axes.

Since mastering data is factory-set, mastering is unnecessary in normal operation. If one of the following events occurs, however, mastering must be performed:

- Mastering data is lost for some reason such as a drop in the voltage of the backup battery for C-MOS in the controller or memory erasing with an initial start.
- The APC pulse counts are lost for some reason such as a drop in the voltage of the backup battery for pulsecoder in the mechanical unit or exchange of pulsecoder.
- The pulse counts do not indicate the angles of the axes because the mechanical unit was hit bumped, etc.

CAUTION

The robot data including mastering data and the pulsecoder data are maintained independently by backup batteries. If the batteries go empty, data is lost. To prevent this, replace both batteries periodically. When the battery voltage drops, an alarm `BLAL' notifies the user.

There are five types of mastering as listed below.

Table B.2 Mastering types

Type of mastering	Explanation
Jig mastering	Mastering is performed using a special jig. Jig mastering is performed at the factory.
Mastering at the zero-degree positions	Mastering is performed with each axis of the robot aligned with the zero-degree position. The zero-degree position mark attached to each axis of the robot is referenced.

Type of mastering	Explanation
Quick mastering	The mastering position can be set at any position. To do this, reference points must be set in advance.
Single axis mastering	Single axis mastering is that the mastering is performed every one axis.
Setting mastering data	Mastering data is set in mastering counters directly.

**CAUTION**

After the robot is installed, the quick mastering reference points should be stored in case the factory-adjusted settings are needed in mastering in the future.

After mastering, be sure to perform positioning (calibration). Positioning means that the controller reads the current pulse counts and recognizes the current position.

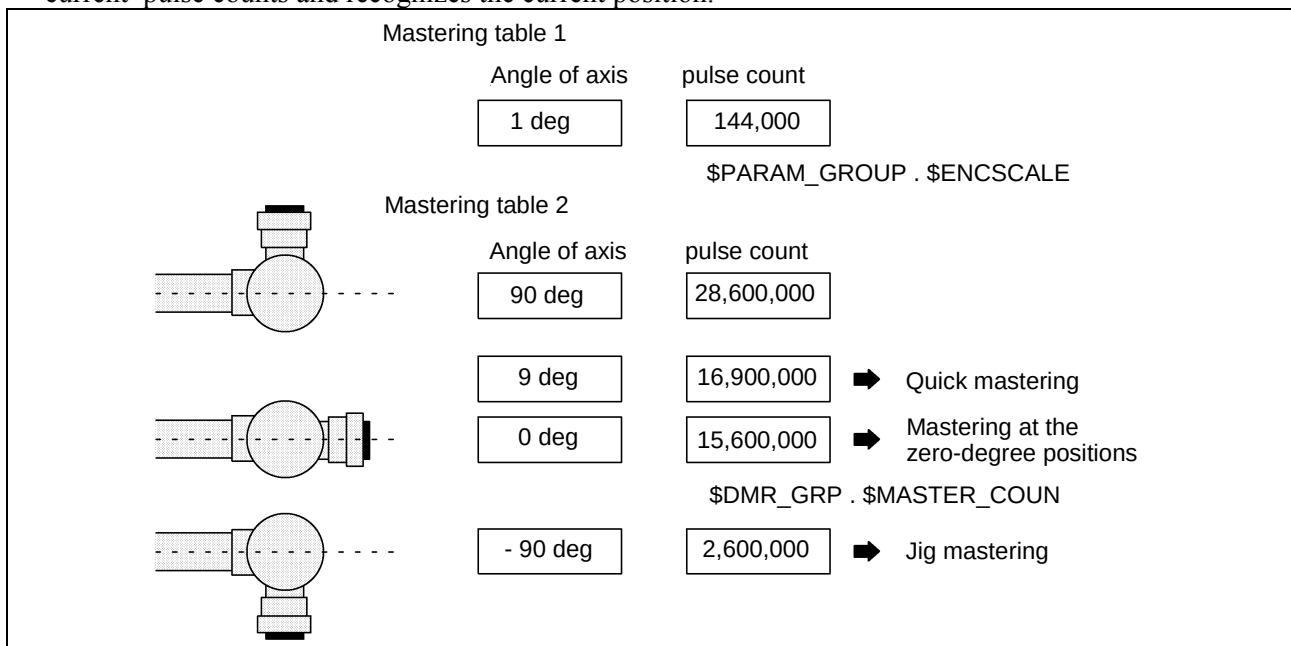


Fig. B.2 Mastering

The current position of the robot is determined by the following data:

- Pulse count per degree (See mastering table 1.) This value is defined in system variable $\$PARAM\_GROUP.\$ENCSCALE$.
- Pulse count at the zero-degree position (See mastering table 2.) This data is stored in $\$DMR\_GRP.\$MASTER\_COUN$ by mastering.
 - In jig mastering, the pulse count at the jig position is received and converted to mastering data.
 - In quick mastering, the pulse count at the quick mastering reference position defined by the user is received and converted to mastering data.
- Current pulse count. The current pulse count is received from the pulsecoder by calibration.

Mastering and calibration are performed on the Master/Cal screen [6 SYSTEM, Master/Cal].

NOTE

Mastering by mistake may cause the robot to move unexpectedly and it is very dangerous. Therefore, the Master/Cal screen will be displayed only when the system variable, $\$MASTER\_ENB$, is set to 1 or 2. Press F5,DONE, which is displayed in the Master/Cal screen after mastering. $\$MASTER\_ENG$ is automatically set to 0 and then the Master/Cal screen can not be displayed. If you want to display the Master/Cal screen again, set $\$MASTER\_ENB$ to 1 in the system variable screen again.

B.2.1 Jig Mastering

Jig mastering is performed at the factory using a special jig. This mastering is performed at the mastering position set beforehand.

With this mastering, the accurate mastering can be performed by using the special jig.

This mastering is usually unnecessary to perform it in normal operation because this is used at shipment.

For details of jig mastering, refer to the mechanical manual.

Procedure B-6 Jig mastering

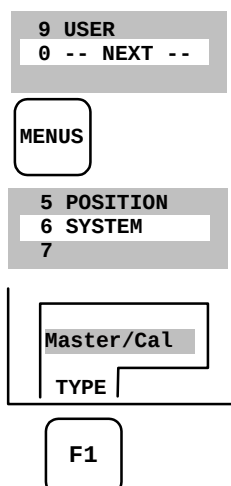
Condition

- System variable \$MASTER_ENB must be set to 1 or 2.

SYSTEM Variables		JOINT 10%
		57/136
57 \$MASTER_ENB	1	

Step

- 1 Press the MENUS key. The screen menu is displayed.
- 2 Select "0 -- NEXT --" and then select "6 SYSTEM".
- 3 Press F1, [TYPE]. The screen change menu is displayed.
- 4 Select "Master/Cal" on the screen change menu. The positioning screen appears.



SYSTEM Master/Cal		JOINT 30%
		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.		
[TYPE]	LOAD RES_PCA	DONE

- 5 Move the robot by jog feed to the mastering position. Release the brake on the manual brake control screen if necessary.

- 6 Select “1 FIXTURE POSITION MASTER” and press the F4 key (yes). Mastering data is set.

SYSTEM Mater/Cal	
1 FIXTURE POSITION MASTER	ENTER
2 ZERO POSITION MASTER	
Master at master position? [NO]	
Master at master position? [NO]	
[TYPE]	YES NO
F4	

SYSTEM Master/Cal		JOINT 30%
		TORQUE=[ON]
1 FIXTURE POSITION MASTER		
2 ZERO POSITION MASTER		
3 QUICK MASTER		
4 SINGLE AXIS MASTER		
5 SET QUICK MASTER REF		
6 CALIBRATE		
Robot Mastered! Mstering Data:		
<0> <11808249> <38767856>		
<9873638> <122000309> <2000319>		
[TYPE]	LOAD RES_PCA	DONE

- 7 Select “6 CALIBRATE” and press the F4 key (yes). Calibration is performed.

5 SET QUICK MASTER REF	
6 CARIBRATE	ENTER
Calibrate? [NO]	
Calibrate? [NO]	
[TYPE]	YES NO
F4	

SYSTEM Master/Cal		JOINT 30%
		TORQUE=[ON]
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):		
<10.000> <-25.000> <40.000>		
<5.000> <-15.000> <0.000>		
[TYPE]	LOAD RES_PCA	DONE

- 8 Press F5 “DONE”, after mastering.

DONE
F5

- 9 Alternatively, to perform positioning, turn the power off, then turn it on again. Calibration is performed whenever the power is turned on.

B.2.2 Mastering at the Zero-degree Positions

Mastering at the zero-degree positions is performed for the robot with its all axes at the zero-degree positions. On each axis of the robot, a zero-degree position mark is attached. Using these marks as a reference, move the robot by jog feed to the zero-degree positions for all axes.

Mastering at the zero-degree positions cannot be performed as accurate by as other types of mastering because it relies on visual adjustment. Perform mastering at the zero-degree positions only as an emergency measure.

For details of mastering at the zero-degree positions, refer to the mechanical manual.

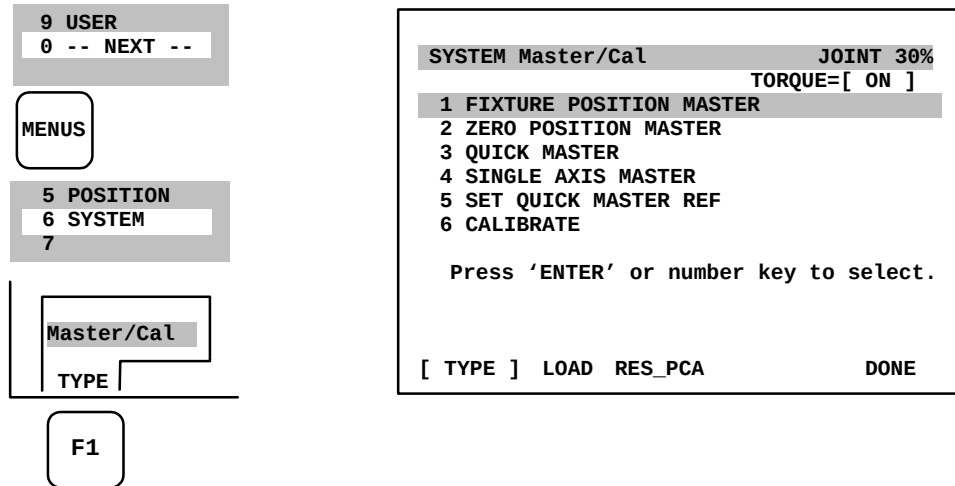
Procedure B-7 Mastering at the zero-degree positions

Condition

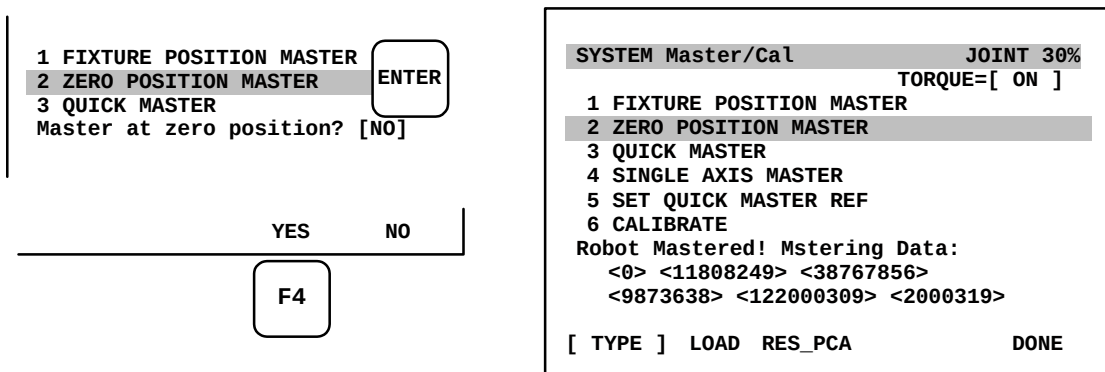
- System variable \$MASTER_ENB must be set to 1 or 2.

Step

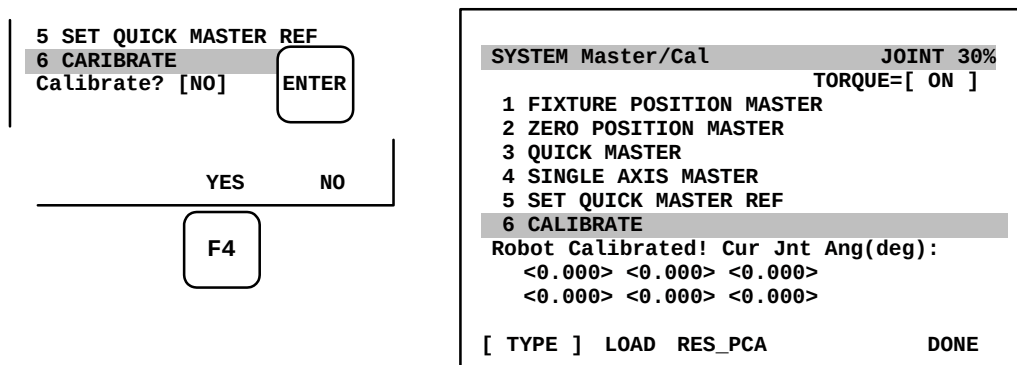
- 1 Press the MENUS key. The screen menu is displayed.
- 2 Select "0 — NEXT —" and then select "6 SYSTEM".
- 3 Press F1, [TYPE]. The screen change menu is displayed.
- 4 Select "Master/Cal" on the screen change menu. The Master/Cal screen appears.



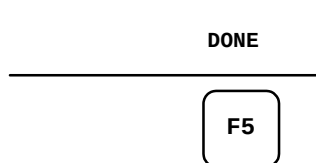
- 5 Move the robot by jog feed to the zero-degree positions for all axes. Set brake control to off, if necessary.
- 6 Select "2 ZERO POSITION MASTER" and press the F4 key (yes). Mastering data is set.



- 7 Select "6 CALIBRATE" and press the F4 key (yes). Calibration is performed.



- 8 Press F5 "DONE", after mastering.



- 9 Alternatively, to perform calibration, turn the power off, then turn it on again. Calibration is performed whenever the power is turned on.

B.2.3 Quick Mastering

Quick mastering allows mastering at any user-defined position. The pulse counts are calculated from the speed and angular displacement within one rotation of the APCs connected to the motors. Quick mastering uses the fact that the absolute angular displacement within one rotation is not lost.

- If mastering data is lost due to the empty of the backup battery for the pulsecoder, quick mastering can be used.
- When the pulsecoder is replaced or when mastering data in the robot controller is lost, quick mastering cannot be used.

To perform quick mastering, a reference point set after mastering is necessary (→ reference point setting). The reference point is factory-set to the zero position.

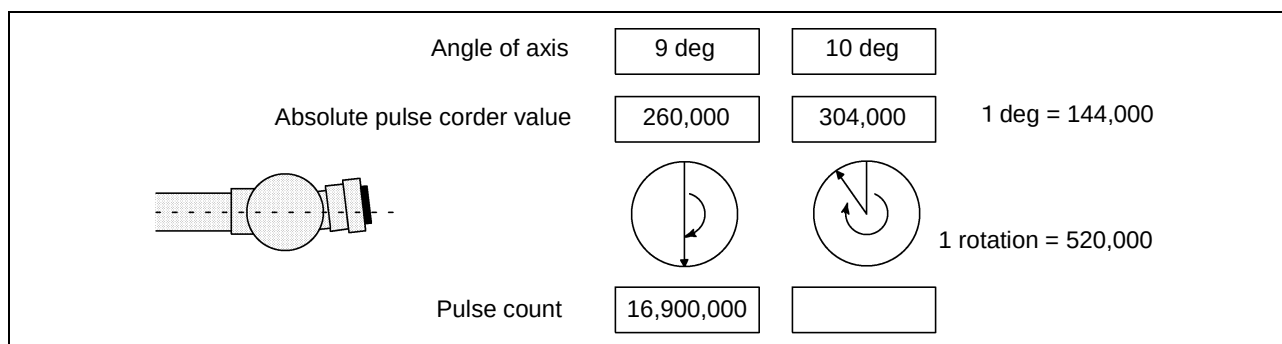


Fig. B.2.3 Quick Mastering

Quick mastering uses the fact that the deviation of the angle of the axis from the reference point can accurately be compensated when it is within one rotation of the APC. For details of quick mastering, refer to the mechanical manual.



CAUTION

If the robot is installed in such a way that the robot cannot be set to the zero degree position, which is the reference point of initial quick mastering, the reference point of quick mastering should be stored after the installation. This must be done to store the factory-set mastering setting, providing for future mastering.

Procedure B-8 Quick mastering

Condition

- System variable \$MASTER_ENB must be set to 1 or 2.
- Quick mastering reference position (reference position) must be set.

Step

- 1 Display the Master/Cal screen.

SYSTEM Master/Cal		JOINT 30%
		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.		
[TYPE]	LOAD RES_PCA	DONE

- 2 Jog the robot to the quick mastering position (reference position). If it is necessary, turn off the brake control.
- 3 Select "3 QUICK MASTER" and press the F4 key (yes). Mastering data is set.

2 ZERO POSITION MASTER	ENTER
3 QUICK MASTER	
4 SINGLE AXIS MASTER	
YES NO	
F4	

- 4 Select "6 CALIBRATE" and press the F4 key (yes). Calibration is performed.
- 5 Press F5 "DONE" after mastering.

DONE
F5

Procedure B-9 Setting reference points for quick mastering
(If the robot is installed in such a way that the robot cannot be set to the zero-degree position)

**CAUTION**

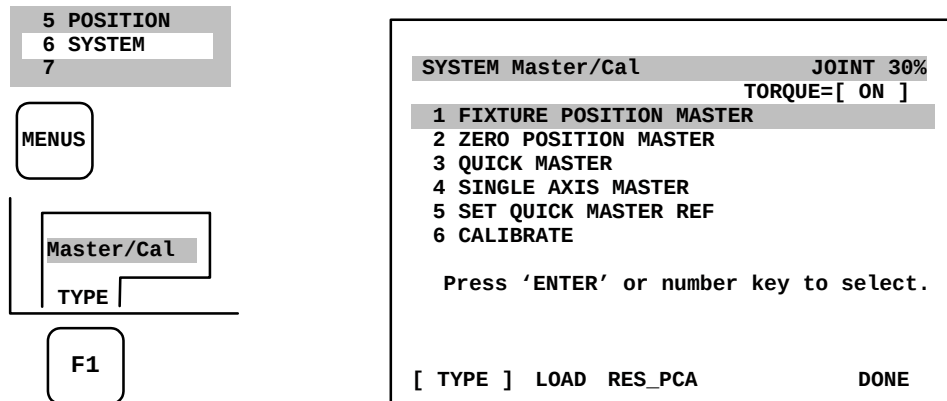
This operation cannot be executed if the mastering data is lost because of mechanical disassembly or maintenance. If that is the case, jig mastering or zero-degree positions mastering should be executed to restore the mastering data.

Condition

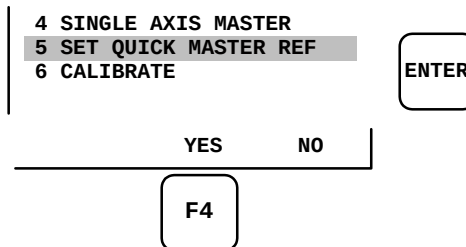
- System variable \$MASTER_ENB must be set to 1 or 2.

Step

- 1 Select "6 SYSTEM" on the screen menu.
- 2 Select "Master/Cal" on the screen change menu. The Master/Cal screen appears.



- 3 Move the robot by jog feed to the quick mastering reference position. Set brake control to off, if necessary.
- 4 Select “5 SET QUICK MASTER REF” and press the F4 key (yes). The reference points for quick mastering are stored in memory.



B.2.4 Single Axis Mastering

User can select the arbitrary position for the mastering of each axis.

Single axis mastering should be used when the mastering data of some axes is lost for some reason such as the drops of the voltage of the backup battery for pulsecoder or exchanging of the pulsecoder.

SINGLE AXIS MASTER				JOINT 30%
				1/9
ACTUAL POS	(MSTR POS)	(SEL)	[ST]	
J1 25.255	(0.000)	(0)	[2]	
J2 25.550	(0.000)	(0)	[2]	
J3 -50.000	(0.000)	(0)	[2]	
J4 12.500	(0.000)	(0)	[2]	
J5 31.250	(0.000)	(0)	[2]	
J6 43.382	(0.000)	(0)	[2]	
E1 0.000	(0.000)	(0)	[2]	
E2 0.000	(0.000)	(0)	[2]	
E3 0.000	(0.000)	(0)	[2]	
				GROUP EXEC

Table B.2.4 Settings for single axis mastering

Items	Descriptions
ACTUAL POS	The current position expressed by joint (degree) of the robot is displayed.
MSTR POS	Specifies the mastering position to the axis to be performed the single axis mastering. It is usually specified 0 degree.
SEL	For the axis to be performed mastering, set this item to 1. It is usually 0.

Items	Descriptions
ST	Display the state of completion of the single axis mastering. The value displayed at this item can not be directly changed. The values of \$EACHMST_DON[1 to 9] are displayed at this column. - 0 Specifies that the mastering data has been lost. The single axis mastering need to be performed. - 1 The mastering data has been lost.(Only other interactive axes is performed mastering.) This axis need to be performed mastering. - 2 The mastering has been completed.

Refer to the mechanical manual for an accurate method of single axis mastering.

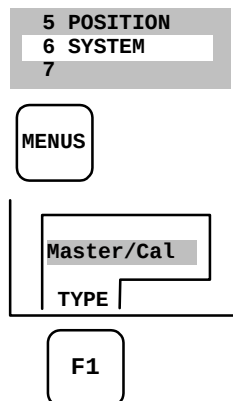
Procedure B-10 Single axis mastering

Condition

- System variable \$MASTER_ENBL must be set to 1.

Step

- Select "6 SYSTEM" on the screen menu.
- Select "Master/Cal" on the screen change menu. The Master/Cal screen appears.



```

SYSTEM Master/Cal          JOINT 30%
                           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.

[ TYPE ]  LOAD  RES_PCA          DONE
  
```

- Select "4 SINGLE AXIS MASTER". The single axis mastering screen is displayed.

In the example on the mastering of J5 and the J6 axis needs to be executed.

SINGLE AXIS MASTER				JOINT 30%
				1/9
	ACTUAL POS	(MSTR POS)	(SEL)	[ST]
J1	25.255	(0.000)	(0)	[2]
J2	25.550	(0.000)	(0)	[2]
J3	-50.000	(0.000)	(0)	[2]
J4	12.500	(0.000)	(0)	[2]
J5	31.250	(0.000)	(0)	[0]
J6	43.382	(0.000)	(0)	[0]
E1	0.000	(0.000)	(0)	[2]
E2	0.000	(0.000)	(0)	[2]
E3	0.000	(0.000)	(0)	[2]
				GROUP EXEC

- Enter 1 to SEL setting field of the axis that you want to perform the mastering. SEL can be specified every one axis or plural axes simultaneously.

ER	JOINT 30%
	2/9
(0.000)	(0) [0]
(0.000)	(0) [0]

SINGLE AXIS MASTER	JOINT 30%
	3/9
J5 31.250 (0.000)	(1) [0]
J6 43.382 (0.000)	(1) [0]
	GROUP EXEC

- 5 Jog the robot to the mastering position. Turn off the brake control if it is necessary.
- 6 Enter the axis data of the mastering position.

ER	JOINT 30%
	2/9
(0.000)	(1) [0]
(0.000)	(1) [0]

SINGLE AXIS MASTER	JOINT 30%
	3/9
J5 31.250 (0.000)	(1) [0]
J6 43.382 (90.000)	(1) [0]
	GROUP EXEC

- 7 Press F5 "EXEC." The mastering is performed. This operation causes that SEL is set to 0 and ST is set to 2 or 1.

GROUP	EXEC
	F5

SINGLE AXIS MASTER	JOINT 30%
	1/9
ACTUAL POS (MSTR POS) (SEL) [ST]	
J1 25.255 (0.000)	(0) [2]
J2 25.550 (0.000)	(0) [2]
J3 -50.000 (0.000)	(0) [2]
J4 12.500 (0.000)	(0) [2]
J5 0.000 (0.000)	(0) [2]
J6 90.000 (90.000)	(0) [2]
E1 0.000 (0.000)	(0) [2]
E2 0.000 (0.000)	(0) [2]
E3 0.000 (0.000)	(0) [2]
	GROUP EXEC

- 8 When the single axis mastering is completed, press the PREV key to display the Master/Cal screen.

PREV

SYSTEM Master/Cal	JOINT 30%
	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.	
[TYPE] LOAD RES_PCA	DONE

- 9 Select "6 CALIBRATE" and press F4 "YES." The calibration is performed.
- 10 Press F5 "DONE", after calibration.

DONE
F5

B.2.5 Setting Mastering Data

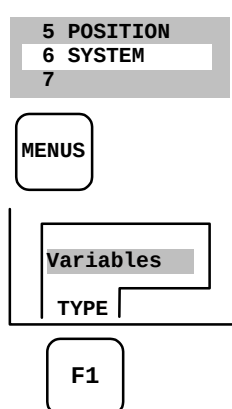
Mastering data can directly be set to the system variable. Setting mastering data can be performed when the pulse counts are not changed.

- If C-MOS mastering data is lost for some reason such as an initial start, set the recorded mastering data.
- Setting mastering data cannot be performed when pulse count data is lost.

Procedure B-11 Directly setting mastering data

Step

- 1 Select "6 SYSTEM" on the screen menu.
- 2 Select "Variables" on the screen change menu. The system variable screen appears.



SYSTEM Variables		JOINT 10%
		1/655
1	\$AAVM	AAVM_T
2	\$ABSPOS_GRP	ABSPOS_GRP_T
3	\$ACC_MAXLMT	150
4	\$ACC_MINLMT	0
5	\$ACC_PRE_EXE	0
6	\$ANGTOL	[9] of Real
7	\$APPLICATION	[3] of STRING[21]
8	\$AP_ACTIVE	2
9	\$AP_AUTOMODE	FALSE
10	\$AP_CHGAPONL	TRUE
[TYPE] DETAIL		

- 3 Change mastering data.
Mastering data is stored in system variable \$DMR_GRP.\$MASTER_COUN.

13	\$DMR_GRP	DMR_GRP_T
14	\$ENC_STAT	[2] of ENC_STAT_T
[TYPE]		

- 4 Select "\$DMR_GRP."

DMR_GRP_T
[2] of ENC_STAT_T

ENTER

SYSTEM Variables		JOINT 10%
\$DMR_GRP		1/1
1 [1]	DMR_GRP_T	

SYSTEM Variables		JOINT 10%
\$DMR_GRP[1]		4/28
1 \$MASTER_DONE	FALSE	
2 \$OT_MINUS	[9] of BOOLEAN	
3 \$OT_PLUS	[9] of BOOLEAN	
4 \$MASTER_COUN	[9] of INTEGER	
5 \$REF_DONE	FALSE	
6 \$REF_POS	[9] of REAL	
7 \$REF_COUNT	[9] of INTEGER	
8 \$BCKLSH_SIGN	[9] of BOOLEAN	
9 \$EACHMST_DON	[9] of INTEGER	
10 \$SPC_COUNT	[9] of INTEGER	
[TYPE] DETAIL		

- 5 Select "\$MASTER_COUN" and enter mastering data.

[9] of BOOLEAN
[9] of INTEGER
FALSE

ENTER

SYSTEM Variables		JOINT 10%
\$DMR_GRP[1]. \$MASTER_COUN		1/9
1 [1]	95678329	
2 [2]	10223045	
3 [3]	3020442	
4 [4]	304055030	
5 [5]	20497709	
6 [6]	2039490	
[TYPE] DETAIL		

- 6 Press the PREV key.
7 Set "\$MASTER_DONE" to "TRUE."

TRUE FALSE

F4

SYSTEM Variables		JOINT 10%
\$DMR_GRP[1]		1/28
1 \$MASTER_DONE	TRUE	
2 \$OT_MINUS	[9] of BOOLEAN	

- 8 Display the Master/Cal screen and select "6 CALIBRATE."
9 Press F5 "DONE", after calibration.

DONE

F5

B.3 SOFTWARE VERSION

Screens related to the software version display identification information of the controller. Please report this information to FANUC, if a failure occurs in the controller. The following are the screens related to the software version:

```
[ TYPE ] SOFTWARE CONFIG MOTOR SERVO >
```

- F2 “SOFTWARE” : Displays the software version screen.
- F3 “CONFIG” : Displays the software configuration screen.
- F4 “MOTOR” : Displays the motor information screen.
- F5 “SERVO” : Displays the servo parameter information screen.

Software version screen

The software version screen displays the following information.

STATUS Version ID		JOINT 30%
	SOFTWARE	ID 1/19
1	Handling Tool	7DA7/16
2	S/W Serial No.	DEV_SW
3	Controller ID	F00000
4	Default Personality (from FD)	
5	R-2000iB/165F	V7.70P/16
6	Servo Code	12345678
7	Cart. Mot. Parameter	V3.00
8	Joint Mot. Parameter	V3.00
9	DCS	V2.0.9
10	Stop pattern	A

```
[ TYPE ] SOFTWARE CONFIG MOTOR SERVO >
```

Software configuration

The software configuration screen displays the software installed.

STATUS Version ID		JOINT 30%
	FEATURE	ORD NO: 1/128
1	Handling Tool	H552
2	English Dictionary	H521
3	Multi Language (KANJI)	H530
4	Basic Software	H510
5	KAREL Run-Time Env	J539
6	Robot Servo Code	H930
7	R-2000iB/165F	H740
8	Analog I/O	H550
9	Automatic Backup	J545
10	Background Editing	J616

```
[ TYPE ] SOFTWARE CONFIG MOTOR SERVO >
```


Motor information screen

The motor information screen displays servo control information on each axis.

STATUS			Version ID	JOINT 30%	
GR: AX: MOTOR ID AND INFO:			1/56		
1	1	1	ACaiSR30/3000 80A H1	DSP1-L	
2	1	2	ACaiSR30/3000 80A H2	DSP1-M	
3	1	3	ACaiSR30/3000 80A H3	DSP1-J	
4	1	4	ACaiS12/4000 40A H4	DSP1-K	
5	1	5	ACaiS12/4000 40A H5	DSP1-L	
6	1	6	ACaiS12/4000 40A H6	DSP1-M	
7	**	**	*****		
8	**	**	*****		
9	**	**	*****		
10	**	**	*****		
[TYPE] SOFTWARE CONFIG MOTOR SERVO >					

Servo parameter information screen

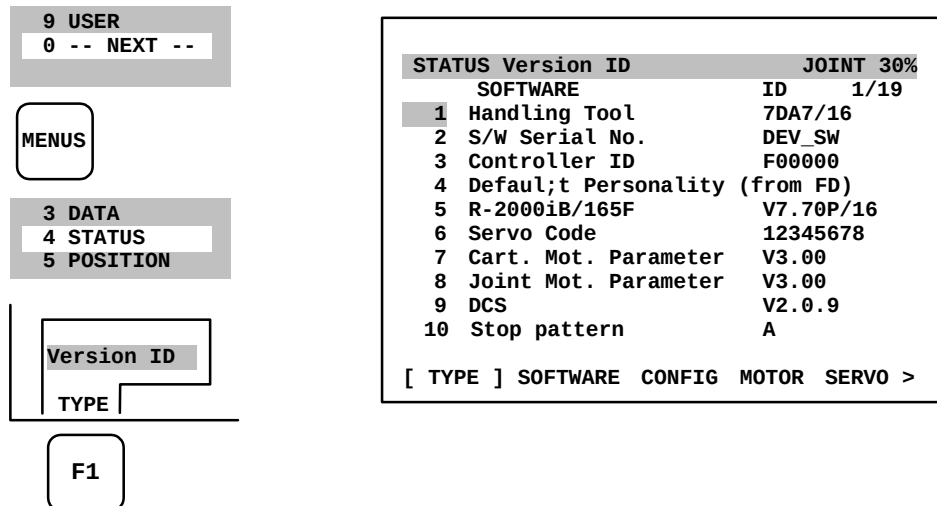
The servo parameter information screen displays the ID of the servo parameter for each axis.

STATUS Version ID			JOINT 30%
GROUP: AXIS: SERVO PARAM ID: 1/56			
1	1	1	P01.02
2	1	2	P01.02
3	1	3	P01.02
4	1	4	P01.02
5	1	5	P01.02
6	1	6	P01.02
7	**	**	*****
8	**	**	*****
9	**	**	*****
10	**	**	*****
[TYPE] SOFTWARE CONFIG MOTOR SERVO >			

Procedure B-12 Software version screen

Step

- 1 Press MENUS key to display the screen menu.
- 2 Select "0 — NEXT —" and then select "4 STATUS" on the next page.
- 3 Press F1 "TYPE" to display the screen change menu.
- 4 Select "Version ID" . Software version screen is displayed.



- F2 “SOFTWARE” : Displays the software version screen.
- F3 “CONFIG” : Displays the software configuration screen.
- F4 “MOTOR” : Displays the motor information screen.
- F5 “SERVO” : Displays the servo parameter information screen.

B.4 ROBOT AXIS STATUS

The robot axis status screens displays the status of each axis motor of the robot. The status of each axis is updated in real time. This status information is used during maintenance.

Status 1 screen

The status 1 screen displays the alarm status of the servo system. The status information consists of servo alarm status 1 (16 bits) and servo alarm status 2 (16 bits).

STATUS Axis		JOINT 30%
		GRP [1]
	FLAG Bits 1/2	History
J1:	0000000000000000	(0000000000000000)
	0000000000000000	(0000000000000000)
J2:	0000000000000000	(0000000000000000)
	0000000000000000	(0000000000000000)
J3:	0000000000000000	(0000000000000000)
	0000000000000000	(0000000000000000)
[TYPE] STATUS1 STATUS2 PULSE [UTIL]>		

Flag 1	Servo alarm status 1
Flag 2	Servo alarm status 2

Table B.4 (a) Servo Alarm Status 1

Address: FC80h (L-axis), FCC0h (M-axis)

MSB	B14	B13	B12	B11	B10	B9	B8
OHAL	LVAL	OVC	HCAL	HVAL	DCAL	FBAL	ALDF
B7	B6	B5	B4	B3	B2	B1	LSB
MCAL	MOFAL	EROFL	CUER	SSTB	PAWT	SRDY	SCRDY

OHAL	Amplifier overheat alarm
LVAL	It indicates a low voltage alarm.
OVC	It indicates an overcurrent (OVC) alarm.
HCAL	It indicates an high current alarm.
HVAL	It indicates an high voltage alarm.
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 1ITP 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.

OVL	FBAL	ALDF	Alarm
1	0	1	Motor overload alarm (not used for a serial pulsecoder)
1	0	0	Amplifier overload alarm
0	1	1	Pulse coder disconnection alarm (not used for a serial pulsecoder)

Table B.4 (b) Servo Alarm Status 2

Address: FC81h (L-axis), FCC1h (M-axis)

MSB	B14	B13	B12	B11	B10	B9	B8
SRCMF	CLALM	FSAL	<u>DCLVAL</u>	<u>BRAKE</u>	IPMAL	SFVEL	GUNSET
B7	B6	B5	B4	B3	B2	B1	LSB
<u>FSSBDC</u>	<u>SVUCAL</u>	<u>AMUCAL</u>	<u>CHGAL</u>	<u>NOAMP</u>			

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
<u>DCLVAL</u>	Low DC Link voltage alarm
<u>BRAKE</u>	Brake alarm of 6-axis amplifier
<u>IPMAL</u>	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.

<u>SFVEL</u>	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.
<u>GUNSET</u>	Servo gun switch completion signal Once the resetting (initialization) of the pulsecoder has been completed after the servo gun is switched, the signal is set to 1 only for 1ITP.
<u>FSSBDC</u>	FSSB disconnection alarm When a disconnection of FSSB is detected, this bit is set to 1. (Hardware detection by FSSBC)
<u>SVUCAL</u>	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)
<u>AMUCAL</u>	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)
<u>CHGAL</u>	Amplifier charge alarm
<u>NOAMP</u>	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).

Status 2 screen

The status 2 screen indicates the pulsecoder alarm status (12 bits).

STATUS Axis		JOINT 30%
		GRP[1]
	Alarm Status	History
J1:	0000000000000000	(0000000000000000)
J2:	0000000000000000	(0000000000000000)
J3:	0000000000000000	(0000000000000000)
J4:	0000000000000000	(0000000000000000)
J5:	0000000000000000	(0000000000000000)
J6:	0000000000000000	(0000000000000000)
[TYPE] STATUS1 STATUS2 PULSE [UTIL]>		

Alarm Status	Pulse coder alarm status

Table B.4 (c) Pulse Coder Alarm Status

				MSB	B10	B9	B8
				SPHAL	STBERR	CRCERR	DTERR
B7	B6	B5	B4	B3	B2	B1	LSB
OHAL	CSAL	BLAL	PHAL	RCAL	BZAL	CKAL	

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.
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.

Pulse screen

The pulse screen displays the servo delay, machine position, and status of the motion command.

STATUS Axis		JOINT 30%	
	Position Error	Machine Pulse	GRP[1] Motion 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
[TYPE] STATUS1 STATUS2 PULSE [UTIL]>			

Position Error	Servo delay (pulses). Delay of the actual pulse to the command pulse
Machine Pulse	Machine position (pulses). Actual absolute pulses
Motion Command	Relative command pulses from the host (pulses)

Monitor screen

The 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.

STATUS Axis		JOINT 30%	
	Torque Monitor		GRP[1]
	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
[TYPE] MONITOR TRACKING DISTURB[UTIL]>			

Ave.	Average of the root-mean-square current values (A)
Max.	Maximum of the root-mean-square current values (A)
Inpos	Position status (0 or 1)
OT	Overtravel status (0 or 1)
VRDY	Servo amplifier ready status (on or off)

Tracking screen

The tracking screen displays the status of the tracking servo system.

STATUS Axis		JOINT 30%	
Tracking Status			
	FLAG Bits 1	Flag Bits 2	
P1:	0000000000000000	0000000000000000	
P2:	0000000000000000	0000000000000000	
	Alarm Status	Counter Value	
P1:	000000000000	0	
P2:	000000000000	0	
[TYPE] MONITOR TRACKING DISTURB			>

Flag Bits 1	Servo alarm status 1
Flag Bits 2	Servo alarm status 2
Alarm Status	Pulse coder alarm status
Counter Value	Line tracking counter

For the servo and pulsecoder alarm statuses, see Table B.4 (a), Table B.4 (b), and Table B.4 (c)

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 pulsecoder. 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 the servo power off.

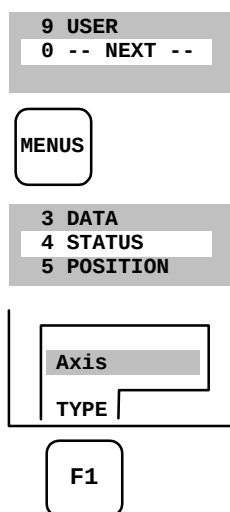
STATUS Axis		JOINT 30%	
Disturbance Torque (A)		GRP[1]	
Curr./	Max.(Allowed)/Min.(Allowed)		
J1 :	0.0 0.0(90.9)	0.0(	-90.9)
J2 :	0.0 0.0(84.3)	0.0(	-84.3)
J3 :	0.0 0.0(97.4)	0.0(	-97.4)
J4 :	0.0 0.0(30.2)	0.0(	-30.2)
J5 :	0.0 0.0(34.3)	0.0(	-34.3)
J6 :	0.0 0.0(21.7)	0.0(	-21.7)
[TYPE] MONITO TRACKING DISTURB [UTIL] >			

Current	Estimated disturbance torque to the servo motor (A)
Max.	Maximum value of the above estimated disturbance torque (A)
Min.	Minimum value of the above estimated disturbance torque (A)

Procedure B-13 Robot axis status screens

Step

- 1 Press the MENUS key to display the screen menu.
- 2 Select "0 — NEXT —" to display the next page, then select "4 STATUS."
- 3 Press the F1 key, "[TYPE]" to display the screen change menu.
- 4 Select "Axis." The robot axis status screens can be displayed.



STATUS Axis		JOINT 30%
		GRP[1]
	FLAG Bits 1/2	History
J1:	0000000000000000	(0000000000000000)
	0000000000000000	(0000000000000000)
J2:	0000000000000000	(0000000000000000)
	0000000000000000	(0000000000000000)
J3:	0000000000000000	(0000000000000000)
	0000000000000000	(0000000000000000)
[TYPE] STATUS1 STATUS2 PULSE [UTIL]>		
[TYPE] MONITOR TRACKING DISTURB [UTIL]>		
[TYPE] REG.DIS DUTY DIAG. [UTIL]>		

- F2 “STATUS 1”: Displays the status 1 screen.
 - F3 “STATUS 2”: Displays the status 2 screen.
 - F4 “PULSE”: Displays the pulse screen.
 - F2 “MONITOR” on the next page: Displays the monitor screen.
 - F3 “TRACKING” on the next page: Displays the tracking screen.
 - F4 “DISTURB” on the next page: Displays the disturbance torque screen.
 - When F2 REG.DIS is selected on the next page, the regenerative discharge screen appears.
- 5 To change the group number, press F5 [UTIL]. A menu appears. On that menu, select 1 GROUP, then enter a desired group number.

B.5 DIAGNOSIS SCREEN

B.5.1 Outline

This function is a function to show users very useful information at maintenance of the robot. Each information has help that shows the description and the recommended action. You can use the robot long time without trouble. The following items are shown.

- Main (List)
- Reducer diagnosis
- Overheat diagnosis
- Torque diagnosis
- Disturbance diagnosis
- OVC diagnosis
- collision diagnosis
- Help

B.5.2 About Reducer Diagnosis

Servo diagnosis function includes reducer’s recommended overhaul time diagnosis. The overhaul time depends on the future motion of reducer. This function indicates the overhaul time in the motion of recent 50 hours. When the exchange or overhaul of the reducer is done, you have to reset parameters. The procedure to reset is as follows.

- Please push E-Stop. The calculation of reducer diagnosis pauses.

- Please set \$FMR_GRP[reducer exchanged group].\$T_LIFE[reducer exchanged axis]=0 from system variable screen.
Parameters of reducer diagnosis are reset.

NOTE

DON'T SET \$FMR_GRP[other group].\$T_LIFE[other axis].

- Please reset E-stop. The calculation of reducer diagnosis restarts.

B.5.3 Procedure

Procedure B-14 Diagnosis screen

Step

- Press the MENUS key to bring up the screen menu.
- Select "4 STATUS" on the next page.
- Press F1 "[TYPE]" to display the pull-up menu.
- Select "Axis".
- Press [next] key until "diag" is shown above function key.
- Press F4 "diag". Diagnosis main screen is shown first.

Diagnosis		JOINT 10%
		1/8
group[1]		
reducer	55542.2 hours	
overheat(motor)	11.31 %	
overheat(trans)	12.12 %	
current	76.39 %	
disturbance	21.75 %	
OVC	9.824 %	
collision detection	19 times	
discharge	134 W	
[TYPE] main reducer ov.heat HELP >		

Procedure B-15 Change diagnosis screen

Step

- Each item is allocated to the function key. Press function key to show the item.
For example, by pressing F3 key reducer diagnosis screen is shown.

Diagnosis/reducer			JOINT 10%
			1/2
group[1]	used	until 100% (estmtd)	
J1	50.0%	46020.2hours	
J2	17.0%	67005.8hours	
J3	18.0%	42501.3hours	
J4	5.0%	73050.0hours	
J5	40.0%	50004.7hours	
J6	3.0%	98050.6hours	
100%(estimated) means L10 life achieved			
Estimation is based on the duty of:			
2008/08/30 19:45 - 2008/09/10 21:45			
[TYPE] main reducer ov.heat HELP >			

- 2 You can change the allocation of function keys by pressing the [next] key.

Diagnosis/reducer		JOINT 10%
		1/2
group[1]	used	until 100% (estmtd)
J1	50.0%	46020.2hours
J2	17.0%	67005.8hours
J3	18.0%	42501.3hours
J4	5.0%	73050.0hours
J5	40.0%	50004.7hours
J6	3.0%	98050.6hours
100%(estimated) means L10 life achieved		
Estimation is based on the duty of:		
torque disturb OVC cl.det.		

- 3 To show the Axis screen again, press [PREV] key.

B.5.4 Each Item

Main: Each item shows the value of the worst axis.

Diagnosis		JOINT 10%
		1/8
group[1]		
reducer		55542.2 hours
overheat(motor)		11.31 %
overheat(trans)		12.12 %
current		76.39 %
disturbance		21.75 %
OVC		9.824 %
collision detection		19 times
discharge		134 W
[TYPE] main reducer ov.heat HELP >		

Reducer: The time until the recommended overhaul of reducers

Diagnosis/reducer		JOINT 10%
		1/2
group[1]	used	until 100% (estmtd)
J1	50.0%	76863.3hours
J2	17.0%	57686.3hours
J3	18.0%	85768.6hours
J4	5.0%	93217.6hours
J5	40.0%	76876.8hours
J6	3.0%	65337.8hours
100%(estimated) means L10 life achieved		
Estimation is based on the duty of:		
2008/08/30 19:45 - 2008/09/10 21:45		
[TYPE] main reducer ov.heat HELP >		

In several editions, only the time until recommended overhaul are shown.

Over heat: The ratio of root mean square current to the rated current

Diagnosis		JOINT 10%
		1/2
over heat		
trans	32.14 %	
motor group[1]		
J1	46.21 %	
J2	63.58 %	
J3	50.13 %	
J4	30.50 %	
J5	15.09 %	
J6	56.97 %	
[TYPE] main reducer ov.heat HELP >		

Torque: The ratio of the current torque to the maximum.

Diagnosis		JOINT 10%
		1/1
torque		
group[1]		
J1	46.21 %	
J2	63.58 %	
J3	50.13 %	
J4	30.50 %	
J5	15.09 %	
J6	56.97 %	
[TYPE] main reducer ov.heat HELP >		

Disturbance: The ratio of the force observed by the servo software to the alarm threshold.

Diagnosis		JOINT 10%
		1/1
disturbance		
group[1]	current	max(%) min(%)
J1	23.63 %	32.33 / -23.52
J2	26.37 %	45.21 / -23.43
J3	-17.74 %	21.56 / -35.21
J4	12.65 %	15.41 / -23.65
J5	-16.55 %	19.37 / -16.55
J6	19.32 %	21.65 / -10.54
[TYPE] main reducer ov.heat HELP >		

OVC: The: The ratio of the temperature simulated by the software to the alarm threshold.

Diagnosis		JOINT 10%
		1/1
OVC		
group[1]		
J1	17.33 %	
J2	23.63 %	
J3	14.74 %	
J4	21.65 %	
J5	14.55 %	
J6	15.32 %	
[TYPE] main reducer ov.heat HELP >		

Collision detection: The count of the collision and the data of the last collision detection.

Diagnosis		JOINT 10%
		1/2
last detection		
2010/ 4/ 6, 16: 55: 26		
group[1]		
count / position		
J1	***** times	-17.33 deg
J2	***** times	23.63 deg
J3	***** times	-14.74 deg
J4	***** times	121.65 deg
J5	14 times	114.55 deg
J6	***** times	115.32 deg
[TYPE] main reducer ov.heat HELP >		

Help: Information of the last shown item

Diagnosis		JOINT 10%
HELP		Arrows to scroll, PREV to exit
INFORMATION		
The count of the collision and the data of the last collision detection.		
REMEDY:		
If many collision detection occur, execute the overhaul more frequently.		
DETAIL:		
You can watch the count of the collision detection ever occurred. You can also watch the information of the last detection(time, position)		
[TYPE] main reducer ov.heat HELP >		

B.6 WORLD FRAME ORIGIN

This section describes the world frame origin of the each robot model (See Section 3.9, "SETTING COORDINATE SYSTEMS" for the world frame). When the user frame or tool frame is set, refer to this.

R series / M series/ ARC Mate / LR Mate

(Other than Top Mount Robot and M-710iC/70W)

A crossing point between J1 axis and level plane which includes the J2 axis.

(Top Mount Robot and M-710iC/70W)

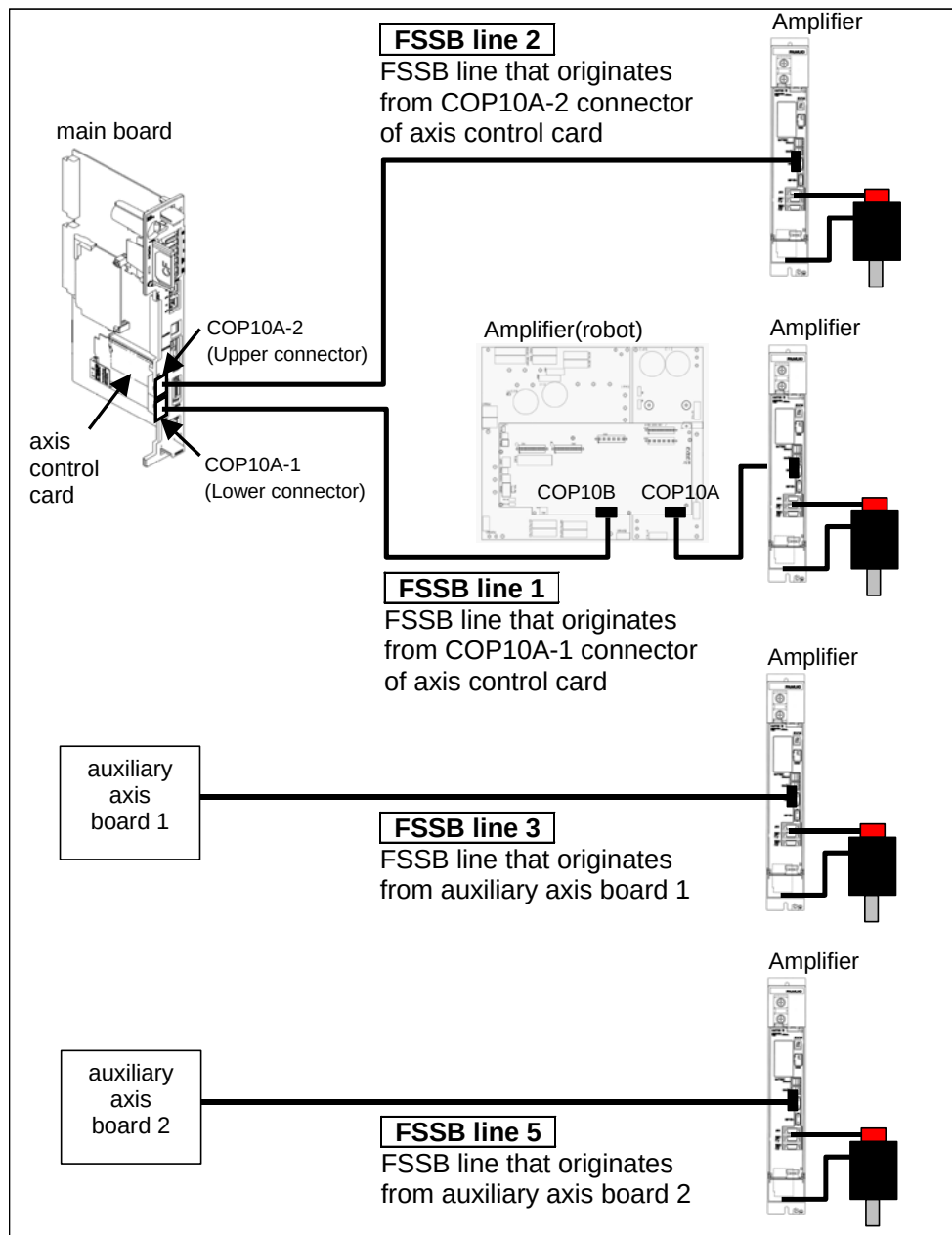
The position on J1 axis that is closest to J4 axis when J1 axis is at the zero position.

B.7 FSSB LINE SETUP

This document describes the FSSB LINE and HARDWARE START AXIS setting that is necessary to set up when system has auxiliary axes.

B.7.1 Definition of FSSB line

FSSB(FANUC Serial Servo Bus) line is defined as follows.



The main board has two slots, “JGP1” and “JGP2” for auxiliary axis boards.

In one auxiliary axis board system, either JGP1 or JGP2 can be used for auxiliary axis board 1. By ordinary, please install the auxiliary axis board to JGP2.

In two auxiliary axis boards system, JGP2 is for auxiliary axis board 1 and JGP1 is for auxiliary axis board 2.

B.7.2 Setting 1 (FSSB line)

Please enter the number of FSSB line that added axis is connected to.

```
**** GROUP 2 INDEPENDENT AXES SETUP ****
```

```
-- FSSB configuration setting --
```

```
1: FSSB line 1 (main axis card)
```

```
2: FSSB line 2 (main axis card)
```

```
3: FSSB line 3 (auxiliary axis board 1)
```

```
5: FSSB line 5 (auxiliary axis board 2)
```

```
Select FSSB line >
```

```
Default value = 1
```

Please enter the value
1, 2, 3 or 5

There are four FSSB lines, 1, 2, 3 and 5. Usually, the first FSSB line is used unless the system has many auxiliary axes or multi robots. This setting is necessary for extended axis and each motion groups, except for group 1.

B.7.3 Setting 2 (Number of total axes on FSSB line 1)

Please enter the number of total axes on FSSB line 1.

This setting is necessary only when added axis is on FSSB line 2.

Total number of axes on FSSB 1 >

Please enter the number of total axes on FSSB line 1.
The number includes robot axes.

B.7.4 Setting 3 (Hardware start axis)

Please enter the start axis number of the first axis in the total hardware configuration of the system. This setting is necessary only when you set up the first axis in a motion group. It is not necessary when you add 2nd or later auxiliary axis in the group.

```
**** GROUP 2 INDEPENDENT AXES SETUP ****
```

```
-- Hardware start axis setting --
```

```
Enter hardware start axis
```

```
(Valid range: 1 - 16)
```

Please enter the number
within valid range

Valid range of hardware start axis differs according to the FSSB line to which auxiliary axis is connected.

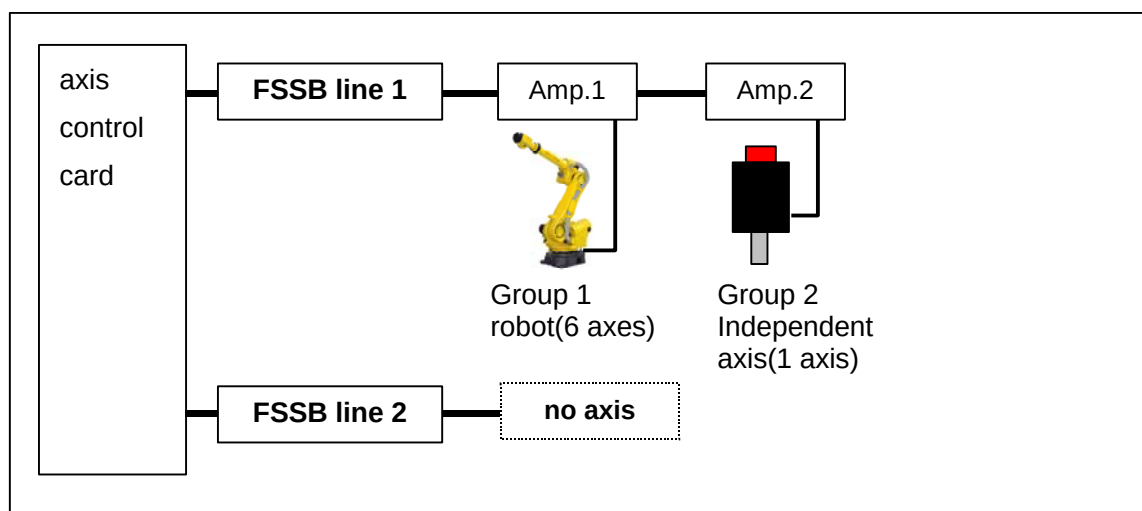
FSSB line	Valid hardware start axis
1	7~16 (*1)
2	*~24 (*2)
3	25~40 (*3)

FSSB line	Valid hardware start axis
5	41~56 (*4)

- (*1) It is possible to use the number less than 7 when the robot does not have six axes.
- (*2) The lower limit of hardware start axis for 2nd FSSB line differs according to the total axis number that is connected to the 1st FSSB line.
- In case that total axis number on the 1st FSSB line is a multiple of 4:
 $\ast = (\text{Total axis number on the 1st FSSB line}) + 1$
- In case that total axis number on the 1st FSSB line is not a multiple of 4:
 $\ast = (\text{Multiple of 4 that is greater than and nearest to the total axis number on the 1st FSSB line}) + 1$
- (*3) The lower limit of hardware start axis for 3rd FSSB line is 25 irrespective of the number of axes on 1st and 2nd FSSB line.
- (*4) The lower limit of hardware start axis for 4th FSSB line is 41 irrespective of the number of axes on 1st, 2nd and 3rd FSSB line.

B.7.5 Setup Examples

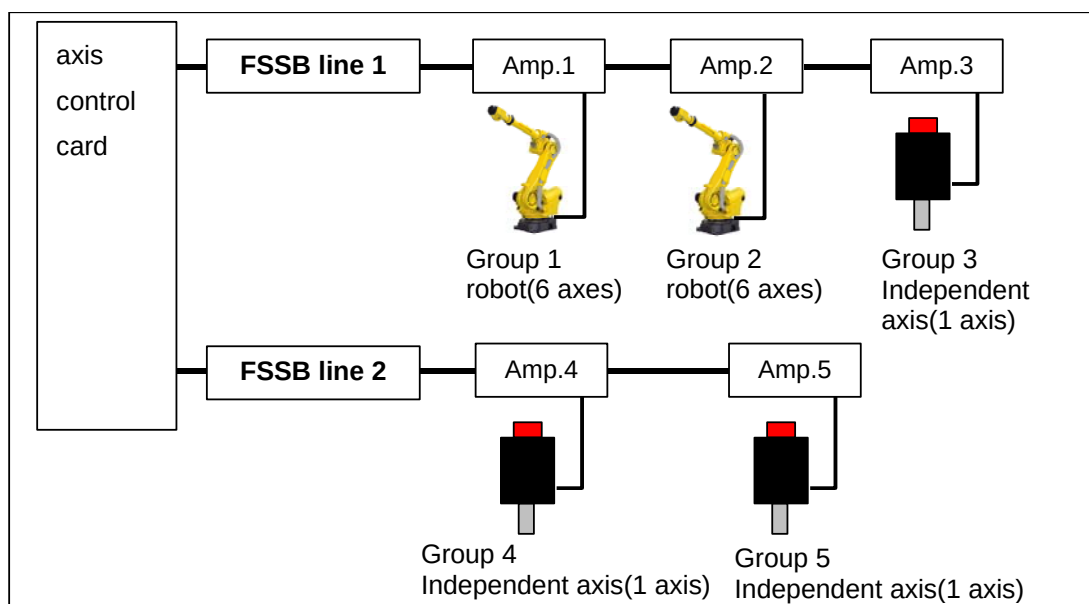
B.7.5.1 Example 1



When system configuration is like above figure, please set up as follows.

Motion group	FSSB line	Total number of axes on FSSB line 1	Hardware start axis
1	1	No need to set up	1
2	1	No need to set up	7

B.7.5.2 Example 2

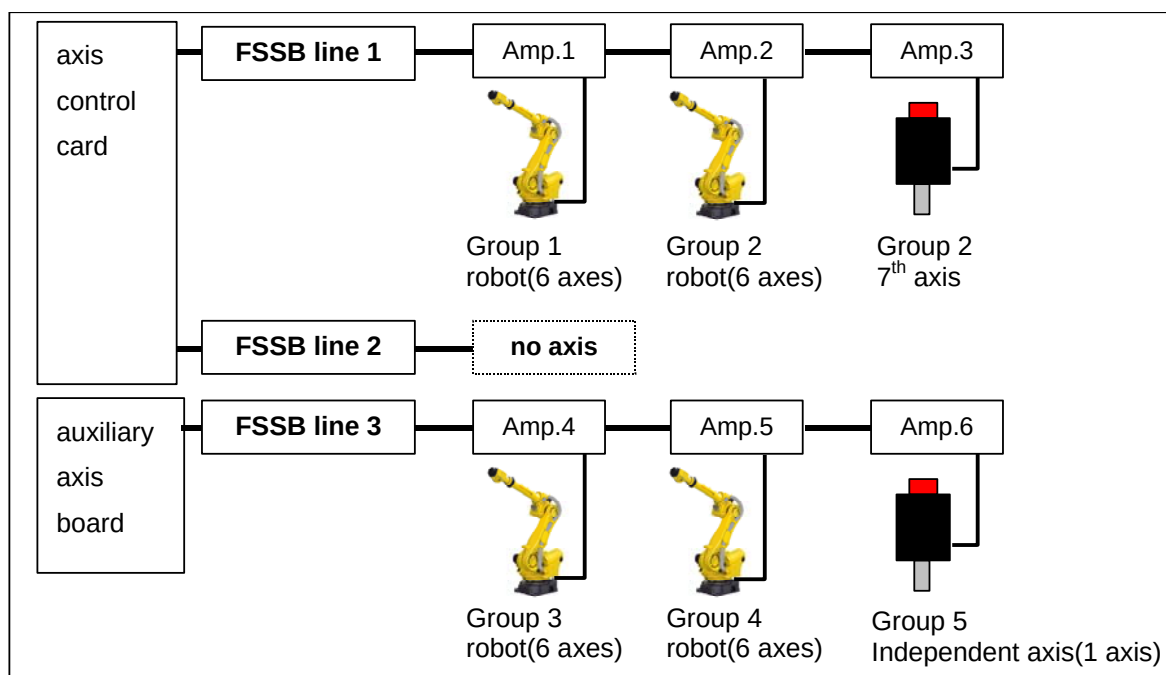


When system configuration is like above figure, please set up as follows.

Motion group	FSSB line	Total number of axes on FSSB line 1	Hardware start axis
1	1	No need to set up	1
2	1	No need to set up	7
3	1	No need to set up	13
4	2	13	17 (*)
5	2	No need to change(13)	18

(*) Hardware start axis of the 1st axis on the 2nd FSSB line must be the (multiple of 4) + 1. Hardware start axis of group 4 is not 14 but 17 for the above hardware configuration.

B.7.5.3 Example 3



When system configuration is like above figure, please set up as follows.

Motion group	FSSB line	Total number of axes on FSSB line 1	Hardware start axis
1	1	No need to set up	1
2	1	No need to set up	7
3	3	No need to set up	25(*)
4	3	No need to set up	31
5	3	No need to set up	37

(*) The number of hardware axis starts from 25 for 3rd FSSB line irrespective of the number of axes on 1st and 2nd FSSB line.

Hardware start axis of group 3 is not 14 but 25 for the above hardware configuration.

B.8 POSITIONER SETUP

Step

- 1 Turn ON the controller with “PREV” key and “NEXT” key pressed. Then select “3. Controlled start”.
- 2 Press MENUS key and select “9. MAINTENANCE”.
- 3 You will see a screen similar to the following.

Setup Robot System Variables			
Group	Robot Library/Option	Ext	Axes
1	R-2000iB/165F		0
2	Basic Positioner		0
[TYPE] ORD_NO AUTO MANUAL			

Press arrow (↑,↓) keys and move the cursor to “Basic Positioner”. Then press F4, ”MANUAL”.

- 4 You will see “FSSB configuration setting” screen. Please setup FSSB line and hardware start axis according to the instruction in section B.7. After adding at least one axis, you will no longer see this screen.
- 5 Select Kinematics Type.

-- Kinematics Type Setting --	
1: Known Kinematics	
2: Unknown Kinematics	
Select Kinematics Type?	
Default value = 1	

Choose “1: Known Kinematics” if the measurements of offset values between POSITIONER axes are accurately known. If not, choose “2: Unknown Kinematics”. After adding at least one axis, you will no longer see this screen.

- 6 You will see a screen similar to the following.
 “?” is the group number.
 “#” is the total number of axes installed. Initial value of # is 0.

***** Group ? Initialization *****	
***** Basic Positioner *****	
Total Axes Installed = #	
1: Display/Modify Axis 1~4	
2: Add Axis	
3: Delete Axis	
4: Exit	
Select item?	

To add a POSITIONER axis, choose “2. Add Axis”. Then setup procedure starts.

To delete a POSITIONER axis, choose “3. Delete Axis”. Then the following message is displayed. “?” is the deleted axis number.


```

POSITIONER Axis ? Was Deleted
Press ENTER to Continue.

```

NOTE

Please refer to the mechanical specification sheet for the following procedure.

- 7 Select the motor of the axis.

```

-- MOTOR SELECTION --

1: Standard Method
2: Enhanced Method
3: Direct Entry Method
Select ==>

```

If you choose “1: Standard Method”, then the following screen is displayed.
Select Motor Size.

```

-- MOTOR SIZE (alpha iS) --

60. aiS2      64. aiS22
61. aiS4      65. aiS30
62. aiS8      66. aiS40
63. aiS12
0. Next page
Select ==>

```

- 8 Select Motor Type.

```

-- MOTOR TYPE --

1. /2000      11. /4000
2. /3000      12. /5000
              13. /6000
Select ==>

```

- 9 Select Current Limit for Motor.

```

-- CURRENT LIMIT FOR MOTOR --

2.  4A      10.  20A
5.  40A     12. 160A
7.  80A
Select ==>

```

- 10 Enter Amplifier Number.

```

-- Amplifier Number Setting --

Enter Amplifier Number (1~56)?

```

- 11 Select Amplifier Type.

```

-- Amplifier Type Setting --

1. A06B-6107 series 6 axes amplifier
2. A06B-6117 series Alpha i amp. or
   A06B-6130 series Beta i amp.
Select Amplifier Type?

```

- 12 Select Axis Type.

```
-- Axis Type Setting --
1: Linear Axis
2: Rotary Axis
Select Axis Type?
```

If the axis does linear motion, choose “1: Linear Axis”.

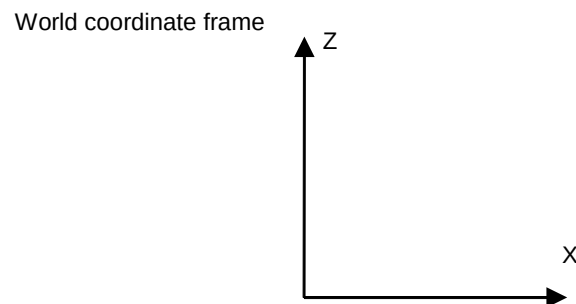
If the axis does rotary motion, choose “2: Rotary Axis”.

13 Select Direction of the axis.

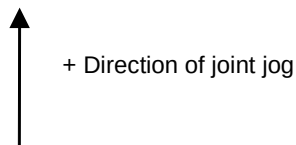
```
-- Direction Setting --
1: +X    3: +Y    5: +Z
2: -X    4: -Y    6: -Z
Select Direction?
```

“Direction” is the axis direction in the world coordinate system. The direction has a sign (+/-).

Example)

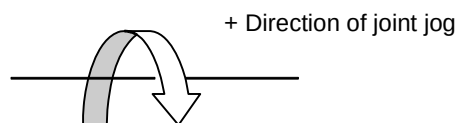


· Linear axis



In this case, the direction should be set to “+Z”.

· Rotary axis



In this case, the direction should be set to “+X”.

14 If you have select “Known Kinematics” in step 5, you will see the following screens. If the Kinematics is “Unknown Kinematics”, this procedure will be skipped.

Enter Offset value in X direction.

```
-- Offset Setting --
Enter Offset X (mm)?
```

Enter Offset value in Y direction.

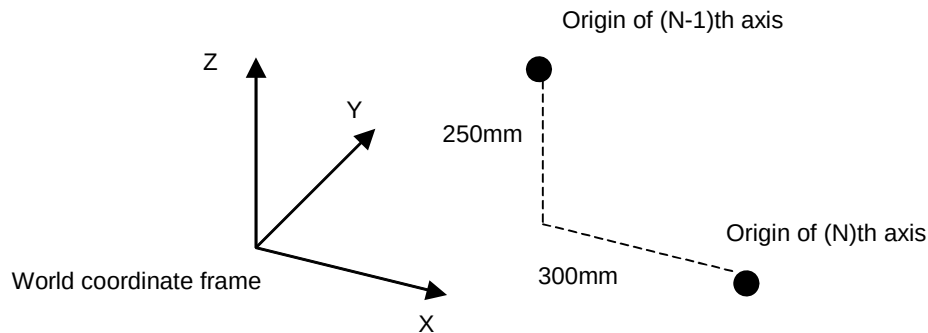
```
Enter Offset Y (mm)?
```

Enter Offset value in Z direction.

```
Enter Offset Z (mm)?
```

For the 1st axis, enter offset values between the origin of the world coordinate and that of the axis.
For the 2nd and later axes, enter offset values between the origin of the axis and that of the previous axis.

Example)



In this case, enter offset values for (N)th axis as follows.

Offset X: 300mm
Offset Y: 0mm
Offset Z: -250mm

15 Enter Gear Ratio.

For a linear axis, enter the distance of the motion which corresponds to one revolution of the motor.
(UNIT: mm/mot-rev)

```
-- Gear Ratio Setting --
Enter Gear Ratio (mm/mot-rev)?
```

For a rotary axis, enter the number of revolution of the motor which corresponds to one revolution of the axis. (UNIT: motor-rev/axis-rev)

```
-- Gear Ratio Setting --
Enter Gear Ratio (mot-rev/axis-rev)?
```

16 Enter Maximum Speed of the axis.

```
-- Maximum Speed Setting --
Suggested Speed = 150.000 (mm/sec)
(Calculated with Max Motor Speed)
Enter (1: Change, 2: No Change)?
```

“Suggested Speed” is the default value.

If you want to change the value, choose “1: Change”.

Then, enter Max Speed.

```
Enter Max Speed (mm/sec)?
```

17 Select Motion Sign.

Choose “1: TRUE” if the joint coordinate position of this axis increases when the motor rotates in the plus direction.

Choose “2: FALSE” if the joint coordinate position of this axis decreases when the motor rotates in the plus direction.

```
-- Motion Sign Setting --
Current value = TRUE
Enter (1: TRUE, 2: FALSE)?
```

- 18 Enter Upper Limit of the axis.
Unit for linear axis is “mm”.
Unit for rotary axis is “degree”.

```
-- Upper Limit Setting --  
Enter Upper Limit (deg)?
```

NOTE

You can determine Upper Limit of the POSITIONER axis by yourself. Here, there is a constraint on Lower Limit of this axis:

$$(\text{Upper Limit}) - (\text{Lower Limit}) = (\text{Stroke Range})$$

Example)

Stroke Range = 100mm

Upper Limit = 50mm

Lower Limit = (Stroke Range) – (Upper Limit) = 50mm

- 19 Enter Lower Limit of the axis.

```
-- Lower Limit Setting --  
Enter Lower Limit (deg)?
```

- 20 Enter the mastering position (Master Position) of the axis.

```
-- Master Position Setting --  
Enter Master Position (deg)?
```

- 21 Enter the acceleration/deceleration time constants (Accel Time).
First, enter the 1st accel time (Accel Time 1).

```
-- Accel Time 1 Setting --  
  
Current value = 384 msec  
Enter (1: Change, 2: No Change)?
```

“Current value” is the default value.

If you want to change the value, choose “1: Change”, then enter the new value.

Next, enter the 2nd accel time (Accel Time 2).

```
-- Accel Time 2 Setting --  
  
Current value = 192 msec  
Enter (1: Change, 2: No Change)?
```

“Current value” is the default value.

If you want to change the value, choose “1: Change”, then enter the new value.

- 22 Select use or not use the exponential acceleration/deceleration time filter.

```
-- Exponential Filter Setting --  
  
Exp Filter Valid = TRUE  
Enter (1: TRUE, 2: FALSE)?
```

If you use Exponential Filter, then enter the exponential acceleration/deceleration time constant.

```
-- Exponential Filter Setting --
Current Value = 20msec
Enter (1: Change, 2: No Change)?
```

“Current value” is the default value.

If you want to change the value, choose “1: Change”, then enter the new value.

- 23 Enter Minimum Accel Time. When doing motion, if the original acceleration/deceleration time is smaller than the specified time, the acceleration/deceleration time will be clamped to the specified time.

```
-- Minimum Accel Time Setting --
Current value = 384(ms)
Enter (1: Change, 2: No Change)?
```

“Current value” is the default value.

If you want to change the value, choose “1: Change”, then enter the new value.

- 24 Enter Load Ratio. This value is the ratio of all load inertia to the rotor inertia. The valid range of Load Ratio is from 1.0 to 5.0 . If you don't set this value, enter “0”.

```
-- Load Ratio Setting --
LoadRatio = (LoadInertia + MotorInertia) / MotorInertia
Enter Load Ratio (0:None, 1~5:Valid)?
```

- 25 Enter Brake Number.

```
-- Brake Number Setting --
Enter Brake Number (0~16)?
```

- 26 Select the type of brake control (Servo Timeout).

```
-- Servo Off Setting --
Servo Off Valid = FALSE
Enter (1: TRUE, 2: FALSE)?
```

If you choose “1: Enable”, then enter the delay time of brake control (Servo Off Time).

```
-- Servo Off Time Setting --
Enter Servo Off Time (0~30 sec)?
```

- 27 Come back to the screen of step 6.

```
***** Group ? Initialization *****
***** Basic Positioner *****

Total Axes Installed = #
1: Display/Modify Axis 1~4
2: Add Axis
3: Delete Axis
4: Exit
Select item?
```

- To display/modify the POSITIONER axis setting, choose “1: Display/Modify Axis”.
- To add the next POSITIONER axis, choose “2: Add Axis”, then start the procedure from step 7.
- To delete the POSITIONER axis, choose “3: Delete Axis”.

- To finish POSITIONER axis setup, choose “4: Exit”.

B.9 EXTENDED AXIS SETUP

Step

- 1 Turn ON the controller with “PREV” key and “NEXT” key pressed. Then select “3. Controlled start”.
- 2 Press MENUS key and select “9. MAINTENANCE”.
- 3 You will see a screen similar to the following.

```

Setup Robot System Variables

  Group  Robot Library/Option  Ext Axes
   1      R-2000iB/165F         0
      Extended Axis Control

[ TYPE ] ORD_NO  AUTO  MANUAL

```

Press arrow (↑,↓) keys and move the cursor to “Extended Axis Control”. Then press F4, ”MANUAL”.

- 4 Choose the group of the extended axis.

```

**** EXTENDED AXIS SETTING PROGRAM ****

-- SELECT GROUP --

0. EXIT
1. Group 1

```

- 5 You will see “Hardware start axis setting” screen. Please setup hardware start axis according to the instruction in section B.7. After adding at least one axis, you will no longer see this screen.
- 6 You will see a screen similar to the following.

```

**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 Initialization *****

                                     E1 E2 E3
*** Group 1 Total Ext Axis = * * *
  1: Display/Modify Ext axis 1~3
  2: Add Ext axes
  3: Delete Ext axes
  4: EXIT
Select?

```

To add new extended axis, choose “2. Add Ext axis”. Then setup procedure starts.

- 7 Enter the number of extended axis. For the first extended axis of the group, input “1”, for the second extended axis of the group, input ”2”, for the third axis of the group, input “3”. Please add axes in order of “1”, “2”, “3”.

```

Enter axis to add (1~3)?

```

- 8 Select the motor of the axis.

```

-- MOTOR SELECTION --

1: Standard Method
2: Enhanced Method
3: Direct Entry Method
Select ==>

```

If you choose “1: Standard Method”, then the following screen is displayed.
Select Motor Size.

```
-- MOTOR SIZE (alpha is) --
60. aiS2      64. aiS22
61. aiS4      65. aiS30
62. aiS8      66. aiS40
63. aiS12
0. Next page
Select ==>
```

9 Select Motor Type.

```
-- MOTOR TYPE --
1. /2000      11. /4000
2. /3000      12. /5000
              13. /6000
Select ==>
```

10 Select Current Limit for Motor.

```
-- CURRENT LIMIT FOR MOTOR --
2. 4A        10. 20A
5. 40A       12. 160A
7. 80A
Select ==>
```

11 Select the type of extended axis from among the four types.

```
-- EXTENDED AXIS TYPE --
1. Integrated Rail (Linear axis)
2. Integrated Arm (Rotary axis)
3. Auxiliary Linear Axis
4. Auxiliary Rotary Axis
Select?
```

NOTE

Integrated Rail / Arm:

World coordinate frame is fixed to the ground. Cartesian position of robot TCP varies if this extended axis moves.

Auxiliary Linear / Rotary Axis:

World coordinate frame is fixed to the robot. Cartesian position of robot TCP does not vary if this extended axis moves.

Integrated Rail (Linear axis):

Enter the axis direction in the world coordinate system.

```
Direction 1:X 2:Y 3:Z
Enter Direction (1-3)?
Select?
```

Integrated Arm (Rotary axis):

Enter the offset length in Z direction from the extended axis frame origin to the robot world frame origin.

```
Enter Offset Length (mm)?
```

Next, enter the arm length of extended axis.

- [Case 1] Direction of the extended axis is parallel to X-axis of the robot world frame:
Enter the offset length in Y direction from the extended axis frame origin to the robot world frame origin.

- [Case 2] Direction of the extended axis is parallel to Y-axis of the robot world frame:
Enter the offset length in X direction from the extended axis frame origin to the robot world frame origin.
- [Case 3] Direction of the extended axis is parallel to Z-axis of the robot world frame:
Enter the offset length in X direction from the extended axis frame origin to the robot world frame origin.

Enter Arm Length (mm)?

Then, select the direction of the extended axis. The extended axis should be parallel to one of the robot world frame (X,Y,Z). Choose one.

Direction 1:X 2:Y 3:Z
Enter Direction (1-3)?
Select?

12 Enter Gear Ratio.

For a linear axis, enter the distance of the motion which corresponds to one revolution of the motor.
(UNIT: mm/mot-rev)

-- GEAR RATIO --

For a linear axis it is the number of
mm's traveled for one rotation of the
motor.
Enter Gear Ratio?

For a rotary axis, enter the number of revolution of the motor which corresponds to one revolution of the axis. (UNIT: motor-rev/axis-rev)

-- GEAR RATIO --

Enter Gear Ratio

13 Enter Max Joint Speed of the axis.

-- MAX JOINT SPEED SETTING --

Suggested Speed = 666.667(mm/s)
(Calculated with Max motor Speed)
Enter (1: Change, 2: No Change)?

“Suggested Speed” is the default value.

If you want to change the value, choose “1: Change”.

Then, enter Max Joint Speed.

Enter Max Speed (mm/s)?

14 Select Motor Direction.

Choose “1: TRUE” if the joint coordinate position of this axis increases when the motor rotates in the plus direction.

Choose “2: FALSE” if the joint coordinate position of this axis decreases when the motor rotates in the plus direction.

-- MOTOR DIRECTION --

Ext_axis 1 Motion Sign = FALSE
Enter (1: TRUE, 2: FALSE)?

15 Enter Upper Limit of the axis.

Unit for linear axis is “mm”.

Unit for rotary axis is “degree”.


```
-- UPPER LIMIT --
Enter Upper Limit (deg)?
```

NOTE

You can determine Upper Limit of the extended axis by yourself. Here, there is a constraint on Lower Limit of this axis:

$$(\text{Upper Limit}) - (\text{Lower Limit}) = (\text{Stroke Range})$$

Example)

Stroke Range = 100mm

Upper Limit = 50mm

Lower Limit = (Stroke Range) – (Upper Limit) = 50mm

- 16 Enter Lower Limit of the axis.

```
-- LOWER LIMIT --
Enter Lower Limit (deg)?
```

- 17 Enter the mastering position (Master Position) of the axis.

```
-- MASTER POSITION --
Enter Master Position (deg)?
```

- 18 Enter the acceleration/deceleration time constants (ACC/DEC Time).
First, enter the 1st ACC/DEC Time (acc_time1).

```
-- ACC/DEC TIME --
Default acc_time1 = 256(ms)
Enter (1: Change, 2: No Change)?
```

“Default acc_time1” is the default value.

If you want to change the value, choose “1: Change”, then enter the new value.

Next, enter the 2nd ACC/DEC Time (acc_time2).

```
Default acc_time2 = 128(ms)
Enter (1: Change, 2: No Change)?
```

“Default acc_time2” is the default value.

If you want to change the value, choose “1: Change”, then enter the new value.

- 19 Enter Minimum Accel Time. When doing motion, if the calculated acceleration/deceleration time is smaller than the specified time, the acceleration/deceleration time will be clamped to the specified time.

```
-- MIN_ACCEL TIME --
Default min_acctime = 128(ms)
Enter (1: Change, 2: No Change)?
```

“Default min_acctime” is the default value.

If you want to change the value, choose “1: Change”, then enter the new value.

- 20 Enter Load Ratio. This value is the ratio of all load inertia to the rotor inertia. The valid range of Load Ratio is from 1.0 to 5.0 . If you don't set this value, enter "0".

```
-- LOAD RATIO --

      LoadInertia + MotorInertia
LoadRatio = -----
      MotorInertia

Enter Load Ratio (0:None, 1~5:Valid)?
```

- 21 Enter Amplifier Number.

```
-- SELECT AMP NUMBER --

Enter amplifier number (1~56)?
```

- 22 Select Amplifier Type.

```
-- SELECT AMP TYPE --

1. A06B-6107 series 6 axes amplifier
2. A06B-6117 series Alpha i amp. or
   A06B-6130 series Beta i amp.

Select?
```

- 23 Enter Brake Number.

```
-- BRAKE SETTING --

Enter Brake Number (0~16)?
```

- 24 Select the type of brake control (Servo Timeout).

```
-- SERVO TIMEOUT --

Servo Off is Disabled
Enter (1: ENABLE, 2: DISABLE)?
```

If you choose "1: Enable", then enter the delay time of brake control (Servo Off Time).

```
-- SERVO TIMEOUT VALUE --

Enter Servo Off Time? (0.0~30.0 sec)
```

- 25 Come back to the screen of step 6.

```
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 Initialization ****

                                E1 E2 E3
*** Group 1 Total Ext Axis = * * *
  1: Display/Modify Ext axis 1~3
  2: Add Ext axes
  3: Delete Ext axes
  4: EXIT
Select?
```

- To display/modify the extended axis setting, choose "1: Display/Modify Ext axis".
- To add the next extended axis, choose "2: Add Ext Axes", then start the procedure from step 7.
- To delete the extended axis, choose "3: Delete Ext axes".
- To finish extended axis setup, choose "4: EXIT".

B.10 INDEPENDENT ADDITIONAL AXIS SETUP

Step

- 1 Turn ON the controller with “PREV” key and “NEXT” key pressed.
Then select “3. Controlled start”.
- 2 Press MENUS key and select “9. MAINTENANCE”.
- 3 You will see a screen similar to the following.

Setup Robot System Variables			
Group	Robot Library/Option	Ext Axes	
1	R-2000iB/165F	0	
2	Independent Axes	0	
[TYPE]	ORD_NO	AUTO	MANUAL

Press arrow (↑,↓) keys and move the cursor to “Independent Axes”. Then press F4, ”MANUAL”.

- 4 You will see “FSSB configuration setting” screen. Please setup FSSB line and hardware start axis according to the instruction in section B.7. After adding at least one axis, you will no longer see this screen.
- 5 You will see a screen similar to the following.
“?” is the group number.
“#” is the total number of axes installed. Initial value of # is 0.

```

***** Group ? Initialization *****
***** Independent Axes *****

Total Axes Installed = #
1: Display/Modify Axis 1~4
2: Add Axis
3: Delete Axis
4: Exit
Select item?

```

To add an independent axis, choose “2. Add Axis”. Then setup procedure starts.

To delete an independent axis, choose “3. Delete Axis”.

NOTE

Please refer to the mechanical specification sheet for the following procedure.

- 6 Select the motor of the axis.

```

-- MOTOR SELECTION --

1: Standard Method
2: Enhanced Method
3: Direct Entry Method
Select ==>

```

If you choose “1: Standard Method”, then the following screen is displayed.
Select Motor Size.

```

-- MOTOR SIZE (alpha iS) --

60. aiS2      64. aiS22
61. aiS4      65. aiS30
62. aiS8      66. aiS40
63. aiS12
0. Next page
Select ==>

```

- 7 Select Motor Type.

```
-- MOTOR TYPE --
1. /2000    11. /4000
2. /3000    12. /5000
            13. /6000
Select ==>
```

- 8 Select Current Limit for Motor.

```
-- CURRENT LIMIT FOR MOTOR --
2.  4A      10. 20A
5.  40A     12. 160A
7.  80A
Select ==>
```

- 9 Select Axis Type.

```
-- INDEPENDENT AXES TYPE --
1: Linear Axis
2: Rotary Axis
Select?
```

If the axis does linear motion, choose “1: Linear Axis”.

If the axis does rotary motion, choose “2: Rotary Axis”.

- 10 Enter Gear Ratio.

For a linear axis, enter the distance of the motion which corresponds to one revolution of the motor.
(UNIT: mm/mot-rev)

```
-- GEAR RATIO --
For a linear axis it is the number of
mm's traveled for one rotation of the
motor.
Enter Gear Ratio?
```

For a rotary axis, enter the number of revolution of the motor which corresponds to one revolution of the axis. (UNIT: motor-rev/axis-rev)

```
-- GEAR RATIO --
Enter Gear Ratio?
```

- 11 Enter Max Joint Speed of the axis.

```
-- MAX JOINT SPEED SETTING --
Suggested Speed = 666.667(mm/s)
(Calculated with Max motor Speed)
Enter (1: Change, 2: No Change)?
```

“Suggested Speed” is the default value.

If you want to change the value, choose “1: Change”.

Then, enter Max Joint Speed.

```
Enter Max Speed (mm/s)?
```

- 12 Select Motor Direction.

Choose “1: TRUE” if the joint coordinate position of this axis increases when the motor rotates in the plus direction.

Choose “2: FALSE” if the joint coordinate position of this axis decreases when the motor rotates in the plus direction.

```
-- MOTOR DIRECTION --
INDEPENDENT AXES 1 Motion Sign = TRUE
Enter (1: TRUE, 2: FALSE)?
```

- 13 Enter Upper Limit of the axis.
Unit for linear axis is “mm”.
Unit for rotary axis is “degree”.

```
-- UPPER LIMIT --
Enter Upper Limit (deg)?
```

NOTE

You can determine Upper Limit of the independent axis by yourself. Here, there is a constraint on Lower Limit of this axis:

$$(\text{Upper Limit}) - (\text{Lower Limit}) = (\text{Stroke Range})$$

Example)

Stroke Range = 100mm

Upper Limit = 50mm

Lower Limit = (Stroke Range) – (Upper Limit) = 50mm

- 14 Enter Lower Limit of the axis.

```
-- LOWER LIMIT --
Enter Lower Limit (deg)?
```

- 15 Enter the mastering position (Master Position) of the axis.

```
-- MASTER POSITION --
Enter Master Position (deg)?
```

- 16 Enter the acceleration/deceleration time constants (ACC/DEC Time).
First, enter the 1st ACC/DEC Time (acc_time1).

```
-- ACC/DEC TIME --
Default Value of acc_time1 = 256(ms)
Enter (1: Change, 2: No Change)?
```

“Default acc_time1” is the default value.

If you want to change the value, choose “1: Change”, then enter the new value.

Next, enter the 2nd ACC/DEC Time (acc_time2).

```
Default Value of acc_time2 = 128(ms)
Enter (1: Change, 2: No Change)?
```

“Default acc_time2” is the default value.

If you want to change the value, choose “1: Change”, then enter the new value.

- 17 Enter Minimum Accel Time. When doing motion, if the calculated acceleration/deceleration time is smaller than the specified time, the acceleration/deceleration time will be clamped to the specified time.

```
-- MIN_ACCEL TIME --
Default Value of min_acctime = 128(ms)
Enter (1: Change, 2: No Change)?
```

“Default min_acctime” is the default value.

If you want to change the value, choose “1: Change”, then enter the new value.

- 18 Enter Load Ratio. This value is the ratio of all load inertia to the rotor inertia. The valid range of Load Ratio is from 1.0 to 5.0 . If you don’t set this value, enter “0”.

```
-- LOAD RATIO --

          LoadInertia + MotorInertia
LoadRatio = -----
          MotorInertia

Enter Load Ratio (0:None, 1~5:Valid)?
```

- 19 Enter Amplifier Number.

```
-- SELECT AMP NUMBER --

Enter amplifier number (1~56)?
```

- 20 Select Amplifier Type.

```
-- SELECT AMP TYPE --

1. A06B-6107 series 6 axes amplifier
2. A06B-6117 series Alpha i amp. or
   A06B-6130 series Beta i amp.

Select?
```

- 21 Enter Brake Number.

```
-- BRAKE SETTING --

Enter Brake Number (0~16)?
```

- 22 Select the type of brake control (Servo Timeout).

```
-- SERVO TIMEOUT --

Servo Off is Disabled
Enter (1: Enable, 2: Disable)?
```

If you choose “1: Enable”, then enter the delay time of brake control (Servo Off Time).

```
-- SERVO TIMEOUT VALUE --

Enter Servo Off Time? (0.0~30.0 sec)
```

- 23 Come back to the screen of step 5.

```
***** Group ? Initialization *****
***** Independent Axes *****
Total Axes Installed = #
1: Display/Modify Axis 1~4
2: Add Axis
3: Delete Axis
4: Exit
Select item?
```

- To display/modify the independent axis setting, choose “1: Display/Modify Axis”.
- To add the next independent axis, choose “2: Add Axis”, then start the procedure from step 6.
- To delete the independent axis, choose “3: Delete Axis”.
- To finish independent axis setup, choose “4: Exit”.

C SYSTEM VARIABLES

This part of this manual describes the names, functions, standard settings, and valid ranges of system variables.

Contents of this appendix

C.1 FORMAT OF A SYSTEM VARIABLE TABLE

C.2 SYSTEM VARIABLES

C.1 FORMAT OF A SYSTEM VARIABLE TABLE

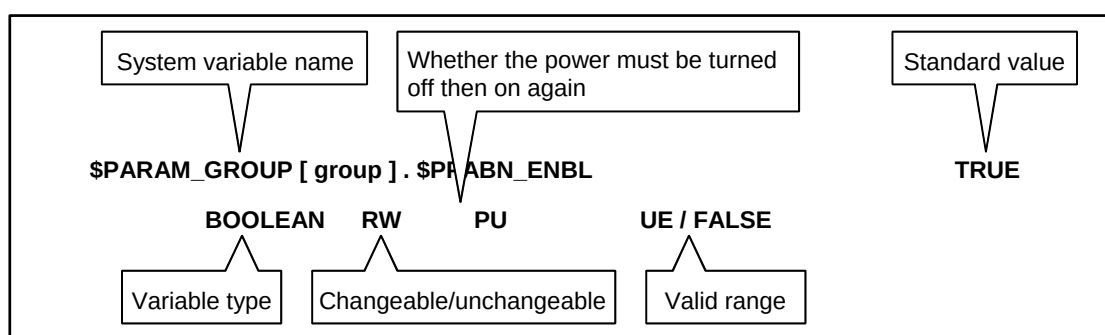


Table C.1 Format of a system variable table

System variable name		
Standard value	* Intrinsic value for each model	
Variable type	BOOLEAN	True/false type (TRUE/FALSE)
	BYTE	Integer (0 to 255)
	SHORT	Integer (-32768 to 32767)
	INTEGER	Integer (-1000000 to 1000000)
	REAL	Real number (-10000000000 to 10000000000)
	CHAR	Character string ("abcdefg")
	XYZWPR	Cartesian coordinates
Changeable/unchangeable	RW	Changeable
	RO	Unchangeable
Whether the power must be turned off then on again	PU	Indicates that the power must be turned on again.
Valid range (unit)		

Procedure C-1 Setting a system variable

Step

- 1 Press the MENUS key.
- 2 Select 0 (NEXT), then select 6 (SYSTEM).
- 3 Press the F1 (TYPE) key.

- 4 Select Variables. Then, the system variable screen is displayed.

9 USER
 0 -- NEXT --

MENUS

5 POSITION
6 SYSTEM
 7

Variables

TYPE

F1

SYSTEM Variables		JOINT 10%
		1/655
1	\$AAVM	AAVM_T
2	\$ABSPPOS_GRP	ABSPPOS_GRP_T
3	\$ACC_MAXLMT	150
4	\$ACC_MINLMT	0
5	\$ACC_PRE_EXE	0
6	\$ANGTOL	[9] of Real
7	\$APPLICATION	[3] of STRING[21]
8	\$AP_ACTIVE	2
9	\$AP_AUTOMODE	FALSE
10	\$AP_CHGAPONL	TRUE

[TYPE] DETAIL

- 5 To change the value of a system variable, move the cursor to a desired item, enter a new value, then press the ENTER key or select a desired item by pressing the corresponding function key.
- 6 When a system variable contains multiple system variables, move the cursor to a desired item and press the ENTER key. Then, the low-order system variables are displayed.

ENTER

SYSTEM Variables		JOINT 10%
		344/655
341	\$OVRD_RATE	5
342	\$PADJ_SCHNUM	10
343	\$PARAM2_GRP	MRR2_GRP_T
344	\$PARAM_GROUP	MRR_GRP_T
345	\$PARAM_MENU	[21] of STRING[21]

SYSTEM Variables		JOINT 10%
\$PARAM_GROUP[1]		1/236
1	\$BELT_ENABLE	FALSE
2	\$CART_ACCEL1	560
3	\$CART_ACCEL2	280
4	\$CIRC_RATE	1
5	\$CONTAXISNUM	0
6	\$EXP_ENBL	FALSE

[TYPE] TRUE FALSE

- 7 After changing the setting of the system variable for which PU is specified, turn off the power, then turn it on again. (PU is specified for all \$PARAM_GROUP system variables.)

NOTE

The setting of a system variable for which RO (unchangeable) is specified cannot be changed.

C.2 SYSTEM VARIABLES

Hot start

\$SEMIPOWERFL FALSE

BOOLEAN RW TRUE/FALSE

[Description] Specifies whether to perform a hot start when power is recovered. After a hot start, the robot is restored near the status immediately before a power failure.

TRUE: Performs a hot start after power recovery.

FALSE: Does not perform a hot start. Instead, performs a cold start.

Power failure recovery

\$PARAM_GROUP[group] . \$SV_OFF_ENB[1] *

\$PARAM_GROUP[group] . \$SV_OFF_ENB[2] *

\$PARAM_GROUP[group] . \$SV_OFF_ENB[3] *

\$PARAM_GROUP[group] . \$SV_OFF_ENB[4] *

\$PARAM_GROUP[group] . \$SV_OFF_ENB[5] *

\$PARAM_GROUP[group] . \$SV_OFF_ENB[6] *

\$PARAM_GROUP[group] . \$SV_OFF_ENB[7] *

\$PARAM_GROUP[group] . \$SV_OFF_ENB[8] *

\$PARAM_GROUP[group] . \$SV_OFF_ENB[9] *

BOOLEAN RW PU TRUE / FALSE

[Function] Enables or disables the break control function

[Description] Specifies how the brake is applied when the axis doesn't move for a given length of time. However the brake control function is disabled for all axes on the same break release DO if there is at least one axis that the brake control is disabled on the DO.

TRUE: It puts on brake when the axis doesn't move for a given length of time and it puts off the brake when the axis starts to move.

FALSE: It never put on brake even if the axis doesn't move for a given length of time.

\$PARAM_GROUP[group] . \$SV_OFF_TIME[1] *

\$PARAM_GROUP[group] . \$SV_OFF_TIME[2] *

\$PARAM_GROUP[group] . \$SV_OFF_TIME[3] *

\$PARAM_GROUP[group] . \$SV_OFF_TIME[4] *

\$PARAM_GROUP[group] . \$SV_OFF_TIME[5] *

\$PARAM_GROUP[group] . \$SV_OFF_TIME[6] *

\$PARAM_GROUP[group] . \$SV_OFF_TIME[7] *

\$PARAM_GROUP[group] . \$SV_OFF_TIME[8] *

\$PARAM_GROUP[group] . \$SV_OFF_TIME[9] *

INTEGER RW PU 0 to 10000000 (msec)

[Function] Time that elapses before the brake control function performs

[Description] Specifies the elapsed time [msec] before it puts on the brake by the brake control function.

\$PARAM_GROUP[group] . \$SV_OFF_ALL TRUE

BOOLEAN RW PU TRUE / FALSE

[Function] Enables or disables the synchronous brake control for all axes

[Description] Specifies how the brakes for all axes are applied at the same time.

TRUE: It puts on/off brakes for all axes at same time, i.e. it does not put on all brakes till all axes finish to move and it puts off all brakes when one axis starts to move.

FALSE: It puts on/off brakes independently, i.e. it puts on each brake release DO which all axes on the DO finish to move and it never put off each brake release DO which any axis on the DO doesn't start to move. However brakes on same brake release DO are not able to be put on/off independently because brakes for all axes on same brake release DO are controlled in a lump.

Mastering**\$MASTER_ENB 0****ULONG RW 1 / 0**

[Function] Displays positioning screen

[Description] When this variable is enabled, the positioning screen [6 (SYSTEM).Master/Cal] is displayed on the teach pendant.

0: Positioning screen not displayed.

1: Positioning screen displayed.

**\$DMR_GRP[group]. \$MASTER_DONE TRUE
BOOLEAN RW TRUE / FALSE**

[Function] Indicates if mastering is completed.

[Description] Indicates if mastering has been completed.

[Setting] On the positioning screen [6 (SYSTEM).Master/Cal]

\$DMR_GRP[group]. \$MASTER_COUN[1] *
\$DMR_GRP[group]. \$MASTER_COUN[2] *
\$DMR_GRP[group]. \$MASTER_COUN[3] *
\$DMR_GRP[group]. \$MASTER_COUN[4] *
\$DMR_GRP[group]. \$MASTER_COUN[5] *
\$DMR_GRP[group]. \$MASTER_COUN[6] *
\$DMR_GRP[group]. \$MASTER_COUN[7] *
\$DMR_GRP[group]. \$MASTER_COUN[8] *
\$DMR_GRP[group]. \$MASTER_COUN[9] *
INTEGER RW 0 to 100000000 (pulse)

[Function] Store mastering pulse counts

[Description] Pulse coder count at zero degree position is stored. This value is calculated from current count at mastering and current position.

\$PARAM_GROUP[group]. \$MASTER_POS[1] *
\$PARAM_GROUP[group]. \$MASTER_POS[2] *
\$PARAM_GROUP[group]. \$MASTER_POS[3] *
\$PARAM_GROUP[group]. \$MASTER_POS[4] *
\$PARAM_GROUP[group]. \$MASTER_POS[5] *
\$PARAM_GROUP[group]. \$MASTER_POS[6] *
\$PARAM_GROUP[group]. \$MASTER_POS[7] *
\$PARAM_GROUP[group]. \$MASTER_POS[8] *
\$PARAM_GROUP[group]. \$MASTER_POS[9] *
REAL RW PU -100000 to 100000 (deg)

[Function] Store jig position for jig mastering

[Description] Jig position for jig mastering is stored. Mastering pulse count is calculated from this data.

Quick mastering**\$DMR_GRP[group]. \$REF_DONE FALSE
BOOLEAN RW TRUE / FALSE**

[Function] Indicates if setting of the reference point for quick mastering is completed.

[Description] When the reference point of quick mastering is set, the pulse coder count and coordinate values of the reference position are stored.

[Setting] On the positioning screen [6 (SYSTEM).Master/Cal]

\$DMR_GRP[group]. \$REF_COUNT[1] 0
\$DMR_GRP[group]. \$REF_COUNT[2] 0
\$DMR_GRP[group]. \$REF_COUNT[3] 0
\$DMR_GRP[group]. \$REF_COUNT[4] 0
\$DMR_GRP[group]. \$REF_COUNT[5] 0
\$DMR_GRP[group]. \$REF_COUNT[6] 0
\$DMR_GRP[group]. \$REF_COUNT[7] 0
\$DMR_GRP[group]. \$REF_COUNT[8] 0
\$DMR_GRP[group]. \$REF_COUNT[9] 0
INTEGER RW 0 to 100000000 (pulse)

[Function] Store reference point mastering count

[Description] Store the count of the pulsecoder when the robot is positioned at the reference point.

\$DMR_GRP[group]. \$REF_POS[1] 0
\$DMR_GRP[group]. \$REF_POS[2] 0
\$DMR_GRP[group]. \$REF_POS[3] 0
\$DMR_GRP[group]. \$REF_POS[4] 0
\$DMR_GRP[group]. \$REF_POS[5] 0
\$DMR_GRP[group]. \$REF_POS[6] 0
\$DMR_GRP[group]. \$REF_POS[7] 0
\$DMR_GRP[group]. \$REF_POS[8] 0
\$DMR_GRP[group]. \$REF_POS[9] 0
REAL RW -100000 to 100000 (deg)

[Function] Store reference point to be set during quick mastering

[Description] Store the reference point to be set during quick mastering.

Positioning

\$MOR_GRP[group]. \$CAL_DONE TRUE
BOOLEAN RW TRUE / FALSE

[Function] Indicates if calibration is completed.

[Description] To check the current position of the robot, the count of the pulsecoder issued and the current position is calculated using mastering count. This check is usually performed when the power is turned on.

[Setting] On the positioning screen [6 (SYSTEM).Master/Cal]

Specifying coordinate systems

\$MNUFRAMENUM[group] 0
BYTE RW 0 to 9

[Function] Specifies user coordinate system number

[Description] Specifies the number of the user coordinate system currently used.

0: World coordinate system

1 to 9: User coordinate system

[Setting] On the user coordinate system setting & screen [6 SYSTEM, Coordinate, User]

\$MNUFRAME[group, 1] XYZWPR
\$MNUFRAME[group, 2] XYZWPR
\$MNUFRAME[group, 3] XYZWPR
\$MNUFRAME[group, 4] XYZWPR
\$MNUFRAME[group, 5] XYZWPR
\$MNUFRAME[group, 6] XYZWPR
\$MNUFRAME[group, 7] XYZWPR
\$MNUFRAME[group, 8] XYZWPR
\$MNUFRAME[group, 9] XYZWPR

POSITION RW XYZWPR

[Function] Specifies user coordinates system number

[Description] Specifies the Cartesian coordinates in the user coordinate system. Up to nine user coordinate systems can be registered.

\$MNUTOOLNUM[group] 0
BYTE RW 0 to 10

[Function] Specifies tool coordinate system number

[Description] Specifies the number of the tool coordinate system currently used.

0: Mechanical interface coordinate system

1 to 10: Tool coordinate system

[Setting] On the tool coordinate system setting screen [6 SYSTEM.Coordinate.Tool]

\$MNUTOOL[group, 1] XYZWPR
\$MNUTOOL[group, 2] XYZWPR
\$MNUTOOL[group, 3] XYZWPR
\$MNUTOOL[group, 4] XYZWPR
\$MNUTOOL[group, 5] XYZWPR
\$MNUTOOL[group, 6] XYZWPR
\$MNUTOOL[group, 7] XYZWPR
\$MNUTOOL[group, 8] XYZWPR
\$MNUTOOL[group, 9] XYZWPR
\$MNUTOOL[group,10] XYZWPR

POSITION RW XYZWPR

[Function] Specifies the tool coordinate system

[Description] Specify the Cartesian coordinates in the tool coordinate system. Ten tool coordinate systems can be registered.

\$JOG_GROUP[group]. \$JOG_FRAME XYZWPR
POSITION RW XYZWPR

[Function] Specifies the jog coordinate system

[Description] Specifies the Cartesian coordinates in the jog coordinate system.

[Setting] On the jog coordinate system setting screen [6 SYSTEM, Coordinate, jog]

Setting motors

\$SCR_GRP[group]. \$AXISORDER[1]	1
\$SCR_GRP[group]. \$AXISORDER[2]	2
\$SCR_GRP[group]. \$AXISORDER[3]	3
\$SCR_GRP[group]. \$AXISORDER[4]	4
\$SCR_GRP[group]. \$AXISORDER[5]	5
\$SCR_GRP[group]. \$AXISORDER[6]	6
\$SCR_GRP[group]. \$AXISORDER[7]	0
\$SCR_GRP[group]. \$AXISORDER[8]	0
\$SCR_GRP[group]. \$AXISORDER[9]	0

BYTE RW 0 to 16

[Function] Specify axis order

[Description] Specifies the order of axes by assigning the physical number of a servo motor controlled by the servo amplifier (servo register) to the logical number of a robot joint axis specified in software (Jx-axis). For instance, when \$AXISORDER[1] = 2, servo motor 2 is assigned to the J1-axis. When \$AXISORDER[1] = 0, no servo motor is assigned as the J1-axis.

\$SCR_GRP[group]. \$ROTARY_AXS[1]	*
\$SCR_GRP[group]. \$ROTARY_AXS[2]	*
\$SCR_GRP[group]. \$ROTARY_AXS[3]	*
\$SCR_GRP[group]. \$ROTARY_AXS[4]	*
\$SCR_GRP[group]. \$ROTARY_AXS[5]	*
\$SCR_GRP[group]. \$ROTARY_AXS[6]	*
\$SCR_GRP[group]. \$ROTARY_AXS[7]	*
\$SCR_GRP[group]. \$ROTARY_AXS[8]	*
\$SCR_GRP[group]. \$ROTARY_AXS[9]	*

BOOLEAN RO TRUE / FALSE

[Function] Specify axis type

[Description] Specifies whether joint axes of the robot are rotational or linear.

TRUE: Rotational

FALSE: Linear

\$PARAM_GROUP[group]. \$MOSIGN[1]	*
\$PARAM_GROUP[group]. \$MOSIGN[2]	*
\$PARAM_GROUP[group]. \$MOSIGN[3]	*
\$PARAM_GROUP[group]. \$MOSIGN[4]	*
\$PARAM_GROUP[group]. \$MOSIGN[5]	*
\$PARAM_GROUP[group]. \$MOSIGN[6]	*
\$PARAM_GROUP[group]. \$MOSIGN[7]	*
\$PARAM_GROUP[group]. \$MOSIGN[8]	*
\$PARAM_GROUP[group]. \$MOSIGN[9]	*

BOOLEAN RW PU TRUE / FALSE

[Function] Specify direction of rotation around axes

[Description] Specify whether the robot moves in the positive or negative direction when the motor rotates positively for each axis.

TRUE: The robot moves in a positive direction when the motor rotates positively.

FALSE: The robot moves in a negative direction when the motor rotates positively.

\$PARAM_GROUP[group]. \$ENCSCALES[1] *
\$PARAM_GROUP[group]. \$ENCSCALES[2] *
\$PARAM_GROUP[group]. \$ENCSCALES[3] *
\$PARAM_GROUP[group]. \$ENCSCALES[4] *
\$PARAM_GROUP[group]. \$ENCSCALES[5] *
\$PARAM_GROUP[group]. \$ENCSCALES[6] *
\$PARAM_GROUP[group]. \$ENCSCALES[7] *
\$PARAM_GROUP[group]. \$ENCSCALES[8] *
\$PARAM_GROUP[group]. \$ENCSCALES[9] *
REAL RW PU -10000000000 to 10000000000 (pulse/deg, pulse/mm)

[Function] Specify unit of pulsecoder count

[Description] Specify how many pulses are required for the pulsecoder when the robot moves around a joint axis one degree or the robot moves along a joint axis 1 mm.
 Rotation axis: \$ENCSCALES = 2E19 x deceleration ratio/360

\$PARAM_GROUP[group]. \$MOT_SPD_LIM[1] *
\$PARAM_GROUP[group]. \$MOT_SPD_LIM[2] *
\$PARAM_GROUP[group]. \$MOT_SPD_LIM[3] *
\$PARAM_GROUP[group]. \$MOT_SPD_LIM[4] *
\$PARAM_GROUP[group]. \$MOT_SPD_LIM[5] *
\$PARAM_GROUP[group]. \$MOT_SPD_LIM[6] *
\$PARAM_GROUP[group]. \$MOT_SPD_LIM[7] *
\$PARAM_GROUP[group]. \$MOT_SPD_LIM[8] *
\$PARAM_GROUP[group]. \$MOT_SPD_LIM[9] *
INTEGER RW PU 0 to 100000 (rpm)

[Function] Specify maximum motor speed

[Description] Specifies the maximum speed of each servo motor for the robot for each axis. When the robot moves around or along a certain axis at a speed exceeding the maximum speed, a warning is issued. Then, the robot decelerates and moves at a speed not exceeding the maximum speed. In this case, the robot may not trace the specified path.

Override

\$SHIFTOV_ENB 0
ULONG RW 0 / 1

[Function] Enables or disables shift override

[Description] The shift override function changes the feedrate override in five steps. To change the feedrate override, press and hold down the SHIFT key, then press the override key as many times as necessary to select the desired override.

1: Enables shift override.

0: Disables shift override.

Press and hold down the SHIFT key, then press the override key: The feedrate override changes in the order: VFINE " FINE → 5% → 50% → 100%.

\$MCR. \$GENOVERRIDE 10
INTEGER RW 0 to 100 (%)

[Function] Specifies the rate of change in feedrate override

[Description] Specifies the rate of changes in the robot feedrate in percentage.

The feedrate changes in this order:

FINE " VFINE → 0% → 50% → 100%. From 0% to 100% it changes in 5% increments.

[Setting] Use the override keys on the teach pendant.

\$MCR. \$PROGOVERRIDE 100
INTEGER RW 0 to 100 (%)

[Function] Specifies program override

[Description] Specifies the percentage of the robot feedrate while the program is being played back.

\$SCR_GRP . \$JOGLIM 12
INTEGER RO 0 to 100%

[Function] Maximum speed scale for coordinate jogging

[Description] Percentage of the maximum speed when jogging the robot in the x, y, or z directions using XYZ or TOOL frame. The maximum speed at linear motion is specified in \$PARAM_GROUP[group].\$SPEEDLIM.

\$SCR . \$JOGLIMROT 12
INTEGER RO 0 to 100%

[Function] Maximum speed scale for orientation jogging

[Description] Percentage of the maximum speed when jogging the robot about the x, y, or z axes using XYZ or TOOL frame. The maximum speed at orientation motion is specified in \$PARAM_GROUP[group].\$ROTSPEEDLIM.

\$SCR_GRP[group]. \$JOGLIM_JNT[1] *
\$SCR_GRP[group]. \$JOGLIM_JNT[2] *
\$SCR_GRP[group]. \$JOGLIM_JNT[3] *
\$SCR_GRP[group]. \$JOGLIM_JNT[4] *
\$SCR_GRP[group]. \$JOGLIM_JNT[5] *
\$SCR_GRP[group]. \$JOGLIM_JNT[6] *
\$SCR_GRP[group]. \$JOGLIM_JNT[7] *
\$SCR_GRP[group]. \$JOGLIM_JNT[8] *
\$SCR_GRP[group]. \$JOGLIM_JNT[9] *
INTEGER RO 0 to 100 (%)

[Function] Specify joint jog override

[Description] The joint jog override function specifies the percentage of the robot feedrate for each axis during jog feed. Specify a low jog override because it is generally unnecessary to move the robot at high speed, and because it is always prudent to avoid danger.

\$SCR. \$COLDVRD 10
INTEGER RO 0 to 100 (%)

[Function] Specifies maximum feedrate override after a cold start

[Description] The feedrate override is set to this value after a cold start.

\$SCR. \$COORDVRD 10
INTEGER RO 0 to 100 (%)

[Function] Specifies maximum feedrate override when the manual-feed coordinate system is changed

[Description] The feedrate override is set to this value or less when the manual-feed coordinate system is changed.

\$SCR. \$TPENBLEVRD 10
INTEGER RO 0 to 100 (%)

[Function] Specifies the maximum feedrate override when the teach pendant is enabled

[Description] The feedrate override is set to this value when the teach pendant is enabled.

\$SCR. \$JOGOVLIM 100
INTEGER RO 0 to 100 (%)

[Function] Specifies the maximum feedrate override during jog feed

[Description] The feedrate override is set to this value or less during jog feed.

\$SCR. \$RUNOVLIM 50
INTEGER RO 0 to 100 (%)

[Function] Specifies the maximum feedrate override when the program is executed

[Description] The feedrate override is set to this value or less when the program is executed.

\$SCR. \$FENCEOVRD 10
INTEGER RO 0 to 100 (%)

[Function] Maximum feedrate override when the safety fence is open

[Description] When the safety fence is opened (*SFSPD input is turned off), the feedrate override is set to this value or below.

\$SCR. \$SFJOGOVLIM 50
INTEGER RO 0 to 100 (%)

[Function] Maximum feedrate override of jog feed when the safety fence is open

[Description] If jog feed is performed while the safety fence is open, the feedrate override is set to this value or below.

\$SCR. \$SFRUNOVLIM 30
INTEGER RO 0 to 100 (%)

[Function] Maximum feedrate override of program execution while the safety fence is open

[Description] When a program is executed with the safety fence open (*SFSPD input set off), the feedrate override is set to this value or below.

\$SCR. \$RECOV_OVRD FALSE
BOOLEAN RW TRUE/FALSE

[Function] Function to restore feedrate override when the safety fence is closed

[Description] When the safety fence is closed (*SFSPD input set on), the previous feedrate override is restored. Then, automatic operation can be started immediately.

This function is enabled when the following conditions are satisfied:

- 1 \$SCR.\$RECOV_OVRD is set to TRUE.
- 2 The system is in the remote state.
- 3 The feedrate override is not changed while the safety fence is open.

If the safety fence is closed while the above conditions are not satisfied, the previous override cannot be restored.

[Setting] General item setting screen [6 SETTING, GENERAL]

Feedrate

\$PARAM_GROUP[group]. \$JNTVELLIM[1] *
\$PARAM_GROUP[group]. \$JNTVELLIM[2] *
\$PARAM_GROUP[group]. \$JNTVELLIM[3] *
\$PARAM_GROUP[group]. \$JNTVELLIM[4] *
\$PARAM_GROUP[group]. \$JNTVELLIM[5] *
\$PARAM_GROUP[group]. \$JNTVELLIM[6] *
\$PARAM_GROUP[group]. \$JNTVELLIM[7] *
\$PARAM_GROUP[group]. \$JNTVELLIM[8] *
\$PARAM_GROUP[group]. \$JNTVELLIM[9] *
REAL RW PU 0 to 100000 (deg/sec, mm/sec)

[Function] Specify the maximum joint speed

[Description] Specify the maximum joint speed for each axis. When the robot moves around or along a certain axis at a speed exceeding the maximum joint speed, a warning is issued. Then, the robot decelerates and moves at a speed not exceeding the maximum joint speed.

\$PARAM_GROUP[group]. \$SPEEDLIM 2000
REAL RW PU 0 to 3000 (mm/sec)

[Function] Specifies the maximum linear feedrate

[Description] Specifies the maximum feedrate during linear or circular motion under path control.

\$PARAM_GROUP[group]. \$ROTSPEEDLIM 90
REAL RW PU 0 to 1440 (deg/sec)

[Function] Specifies the maximum circular feedrate

[Description] Specifies the maximum feedrate during circular motion under attitude control.

Jog feedrate (joint feed) (deg/sec, mm/sec) =

$$\text{Maximum joint feedrate} \times \frac{\text{Joint jog override}}{100} \times \frac{\text{Feedrate override}}{100}$$

Jog feedrate (linear feed) (mm/sec) =

$$\text{Maximum linear feedrate} \times \frac{\text{Jog override}}{100} \times \frac{\text{Feedrate override}}{100}$$

Jog feedrate (circular feed) (deg/sec) =

$$\text{Maximum circular feedrate} \times \frac{\text{Orientation Jog override}}{100} \times \frac{\text{Feedrate override}}{100}$$

Joint jog override \$SCR_GRP[g].\$JOGLIM_JNT[i] (%)

Jog override \$SCR.\$JOGLIM (%)

Orientation jog override \$SCR.\$JOGLIMROT (%)

Maximum joint speed \$PARAM_GROUP[g]. \$JNTVELLIM[i] (deg/sec, mm/sec)

Maximum linear feedrate \$PARAM_GROUP[g]. \$SPEEDLIM (mm/sec)

Maximum circular feedrate \$PARAM_GROUP[g]. \$ROTSPEEDLIM (deg/sec)

Note: g is group number. i is axis number

Operation speed (joint motion) (deg/sec, mm/sec) =

$$\text{Maximum joint feedrate} \times \frac{\text{Programmed feedrate}}{100} \times \frac{\text{Feedrate override}}{100}$$

Operation speed (linear motion) (mm/sec) =

$$\text{Programmed feedrate} \times \frac{\text{Feedrate override}}{100}$$

Operation speed (circular motion) (deg/sec) =

$$\text{Programmed feedrate} \times \frac{\text{Feedrate override}}{100}$$

\$PARAM_GROUP[group]. \$LOWERLIMS[1] *
\$PARAM_GROUP[group]. \$LOWERLIMS[2] *
\$PARAM_GROUP[group]. \$LOWERLIMS[3] *
\$PARAM_GROUP[group]. \$LOWERLIMS[4] *
\$PARAM_GROUP[group]. \$LOWERLIMS[5] *
\$PARAM_GROUP[group]. \$LOWERLIMS[6] *
\$PARAM_GROUP[group]. \$LOWERLIMS[7] *
\$PARAM_GROUP[group]. \$LOWERLIMS[8] *
\$PARAM_GROUP[group]. \$LOWERLIMS[9] *

REAL RW PU -100000 to 100000 (deg, mm)

[Function] Specify the lower limit of the joint operating area

[Description] Specify the lower limit of the joint operating area which is the limit of the motion in the negative direction.

[Setting] Joint operating area screen [6 (SETTING).Joint Area]

```

$PARAM_GROUP[ group ]. $UPPERLIMS[ 1 ]      *
$PARAM_GROUP[ group ]. $UPPERLIMS[ 2 ]      *
$PARAM_GROUP[ group ]. $UPPERLIMS[ 3 ]      *
$PARAM_GROUP[ group ]. $UPPERLIMS[ 4 ]      *
$PARAM_GROUP[ group ]. $UPPERLIMS[ 5 ]      *
$PARAM_GROUP[ group ]. $UPPERLIMS[ 6 ]      *
$PARAM_GROUP[ group ]. $UPPERLIMS[ 7 ]      *
$PARAM_GROUP[ group ]. $UPPERLIMS[ 8 ]      *
$PARAM_GROUP[ group ]. $UPPERLIMS[ 9 ]      *
      REAL  RW    PU    -100000 to 100000 ( deg, mm )

```

[Function] Specify the upper limit of the joint operating area

[Description] Specify the upper limit of the joint operating area, which is the limit of the motion in the positive direction.

[Setting] Joint operating area screen [6 (SETTING).Joint Area]

Payload specification

If load information has not been set up on the load setting screen, it is necessary to enter the following information. If no load setting condition number has been selected, be sure to enter the information correctly, since the robot uses it when it runs. Therefore, be particularly careful when setting these values.

- \$GROUP[group].\$PAYLOAD
- \$PARAM_GROUP[group].\$PAYLOAD
- \$PARAM_GROUP[group].\$PAYLOAD_X
- \$PARAM_GROUP[group].\$PAYLOAD_Y
- \$PARAM_GROUP[group].\$PAYLOAD_Z
- \$PARAM_GROUP[group].\$PAYLOAD_IX
- \$PARAM_GROUP[group].\$PAYLOAD_IY
- \$PARAM_GROUP[group].\$PAYLOAD_IZ
- \$PARAM_GROUP[group].\$AXISINTERTIA[1 to 9]
- \$PARAM_GROUP[group].\$AXISMOMENT[1 to 9]
- \$PARAM_GROUP[group].\$AXIS_IM_SCL
- \$PARAM_GROUP[group].\$ARMLOAD[1 to 3]

```

$GROUP [ group ] . $PAYLOAD *
      REAL  RW    0 to 10000(kgf)

```

[Function] Payload

[Description] Specify a payload. If the load varies during an operation, specify the maximum value.

```

$PARAM_GROUP [ group ] . $PAYLOAD *
      REAL  RW    PU    0 to 10000(kgf)

```

[Function] Payload

[Description] Specify a payload. If the load varies during an operation, specify the maximum value.

```

$PARAM_GROUP[ group ]. $PAYLOAD_X      *
$PARAM_GROUP[ group ]. $PAYLOAD_Y      *
$PARAM_GROUP[ group ]. $PAYLOAD_Z      *
      REAL  RW    PU    -100000 to 10000(cm)

```

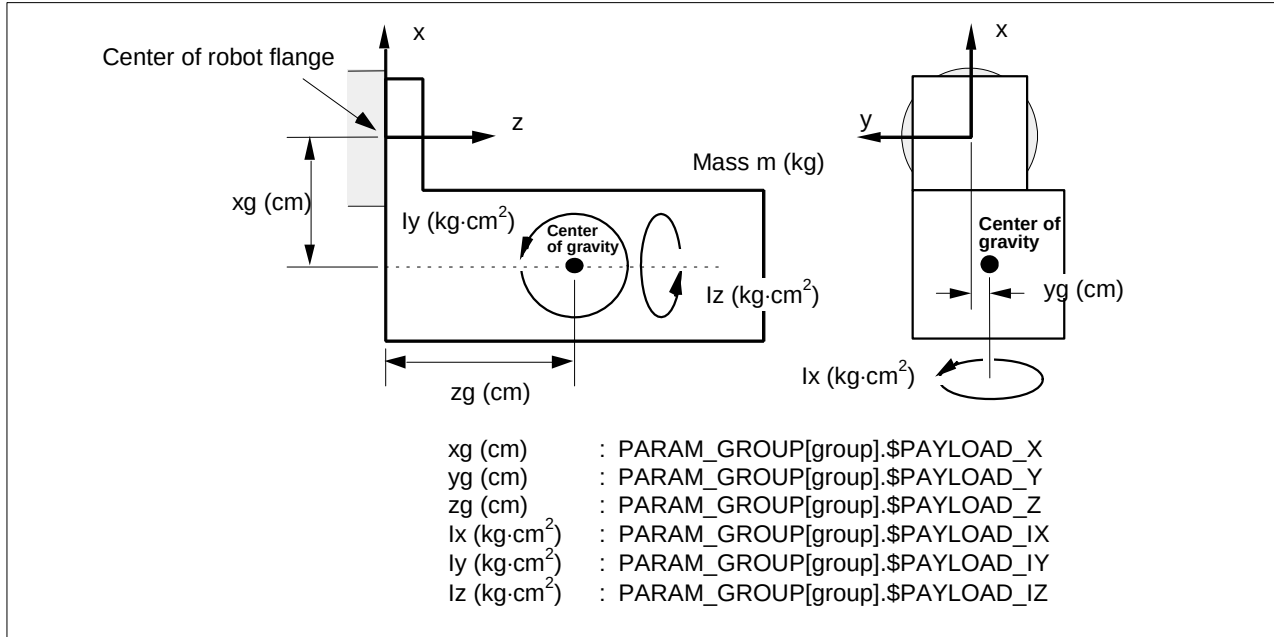
[Function] Load gravity center distance

[Description] Center of gravity of load viewed on the mechanical interface coordinate system (default tool coordinate system). The center of gravity of a load is measured along the X-axis, Y-axis, and Z-axis of the mechanical interface coordinate system.

\$PARAM_GROUP[group]. \$PAYLOAD_IX *
\$PARAM_GROUP[group]. \$PAYLOAD_IY *
\$PARAM_GROUP[group]. \$PAYLOAD_IZ *
REAL RW PU 0 to 10000(kg · cm²)

[Function] Load gravity center inertia

[Description] Inertia around the center of gravity of load. The inertia of a heavy load is calculated around the X-axis, Y-axis, and Z-axis of the mechanical interface coordinate system.
 The meaning of \$PARAM_GROUP[group].\$PAYLOAD_* is as illustrated below:



\$PARAM_GROUP [group]. \$AXISINERTIA[1] *
\$PARAM_GROUP [group]. \$AXISINERTIA[2] *
\$PARAM_GROUP [group]. \$AXISINERTIA[3] *
\$PARAM_GROUP [group]. \$AXISINERTIA[4] *
\$PARAM_GROUP [group]. \$AXISINERTIA[5] *
\$PARAM_GROUP [group]. \$AXISINERTIA[6] *
\$PARAM_GROUP [group]. \$AXISINERTIA[7] *
\$PARAM_GROUP [group]. \$AXISINERTIA[8] *
\$PARAM_GROUP [group]. \$AXISINERTIA[9] *

SHORT RW PU 0 to 32767 (kgf · cm · sec²)

[Function] Payload inertia

[Description] For each axis, specify an integer as the value of the inertia resulting from the applied payload.
 The values for the 1st to 3rd axes are calculated automatically; therefore, they need not be specified. (Set a value for each of the 4th, 5th, and 6th axes.)
 The inertia for each axis is calculated using the following expression:

$$\text{\$AXISINERTIA}[i] = \frac{\text{payload} \times (\text{l\_max}[i])^2}{g} \quad (\text{kgf} \cdot \text{cm} \cdot \text{sec}^2)$$

Payload : Payload [kgf]

l_max[i] : Maximum distance from the rotation center of the axis (axis i) to the mass center of the load on the robot [cm]

For the 4th and 5th axes, the distance may vary depending on the angle of the other axes. In such a case, set the maximum distance than can be achieved.

G : Gravity acceleration (= 980 [cm/sec²])

[NOTE] When specifying or changing this variable, refer to the explanation of \$PARAM_GROUP[.]\$AXIS_IM_SCL, below.

\$PARAM_GROUP [group]. \$AXISMOMENT[1] *
\$PARAM_GROUP [group]. \$AXISMOMENT[2] *
\$PARAM_GROUP [group]. \$AXISMOMENT[3] *
\$PARAM_GROUP [group]. \$AXISMOMENT[4] *
\$PARAM_GROUP [group]. \$AXISMOMENT[5] *
\$PARAM_GROUP [group]. \$AXISMOMENT[6] *
\$PARAM_GROUP [group]. \$AXISMOMENT[7] *
\$PARAM_GROUP [group]. \$AXISMOMENT[8] *
\$PARAM_GROUP [group]. \$AXISMOMENT[9] *
SHORT RW PU 0 to 32767 (kgf · m)

[Function] Axis moment

[Description] For each axis, specify an integer as the moment value resulting from the applied payload.
 The values for the 1st to 3rd axes are calculated automatically; therefore, they need not be specified. (Set a value for each of the each of 4th, 5th, and 6th axes.)
 The moment value for each axis is calculated using the following expression:

$\$AXISMOMENT[i] = \text{payload} \times l\_max[i] \text{ (kgf·m)}$

Payload : Payload [kgf]

$l\_max[i]$: Maximum distance from the rotation center of the axis (axis i) to the mass center of the load on the robot [m]

For the 4th and 5th axes, the distance may vary depending on the angle of the other axes. In such a case, set the maximum distance than can be achieved.

[NOTE] When specifying or changing this variable, refer to the explanation of \$PARAM_GROUP[.].\$AXIS_IM_SCL, below.

\$PARAM_GROUP [group]. \$AXIS_IM_SCL 1
SHORT RW PU 0 to 32767

[Function] Inertia and moment value adjustment scale

[Description] This scale is used to set up a number in decimal places for the inertia and moment values of each axis stated above.

[NOTE] It is usually unnecessary to re-set this variable.

Actually, the following inertia and moment values are used.

(Inertia value) =
$$\frac{\$PARAM\_GROUP[group].\$AXISINERTIA[i]}{\$PARAM\_GROUP[group].\$AXIS\_IM\_SCL}$$

(Moment value) =
$$\frac{\$PARAM\_GROUP[group].\$AXISMOMETN[i]}{\$PARAM\_GROUP[group].\$AXIS\_IM\_SCL}$$

It is therefore necessary to assign \$AXISINERTIA[i] and \$AXISMOMENT[i] with values that match the setting of this variable.

To enter the value "1.23," for example, as the inertia value for the fourth axis of the robot:

- Set up \$PARAM_GROUP[group].\$AXIS_IM_SCL = 100
- Set up \$PARAM_GROUP[group].\$AXISINERTIA[4] = 123
- Change these inertia and moment values for other axes according to the value of \$AXIS_IM_SCL.

\$PARAM_GROUP [group]. \$ARMLOAD[1] *
\$PARAM_GROUP [group]. \$ARMLOAD[2] *
\$PARAM_GROUP [group]. \$ARMLOAD[3] *
REAL RW PU 0 to 10000 (kgf)

[Function] Equipment weight

[Description] When equipment such as welding equipment is installed on a robot axis, specify the payload incurred by that equipment.

\$ARMLOAD[1]: Specify the weight of the equipment installed on the 3rd-axis arm.

\$ARMLOAD[2]: Specify the weight of the equipment installed on the 2nd-axis base.

\$ARMLOAD[3]: Not used.

Executing a program**\$DEFPULSE 4****SHORT RW 0 to 255 (100 msec)**

[Function] Specifies the standard DO output pulse width

[Description] This value is used when the pulse width is not specified for the output of a DO signal pulse.

Automatic operation**\$RMT_MASTER 0****INTEGER RW 0 to 3**

[Function] Specifies which remote unit is used

[Description] Specifies which remote unit is used. The specified remote unit has the right to start the robot.

0: Peripheral unit (remote controller)

1: CRT/keyboard

2: Host computer

3: No remote unit

Deleting the warning history**\$ER_NOHIS 0****BYTE RW 0 / 3**

[Function] Warning history delete function

[Description] WARN alarms, NONE alarms and resets can be deleted from the alarm history.

0: Disables the function. (All alarms and resets are recorded in the history.)

1: Does not record WARN and NONE alarms in the history.

2: Does not record resets.

3: Does not record resets, WARN alarms, and NONE alarms.

Disabling alarm output**\$ER_NO_ALM. \$NOALMENBL 0****BYTE RW 0 / 1**

[Function] Enables the no-alarm output function

[Description] When this function is enabled, the LEDs on the teach pendant and the machine operator's panel corresponding to the alarms specified with system variable \$NOALM_NUM do not light. In addition, the peripheral I/ O alarm signal (FAULT) is not output.

\$ER_NO_ALM. \$NOALM_NUM 5**BYTE RW 0 to 10**

[Function] Specifies the number of alarms not output

[Description] Specifies the number of alarms that are not output.

\$ER_NO_ALM. \$ER_CODE1 11001**\$ER_NO_ALM. \$ER_CODE2 11002****\$ER_NO_ALM. \$ER_CODE3 11003****\$ER_NO_ALM. \$ER_CODE4 11007****\$ER_NO_ALM. \$ER_CODE5 11037****\$ER_NO_ALM. \$ER_CODE6 0****\$ER_NO_ALM. \$ER_CODE7 0****\$ER_NO_ALM. \$ER_CODE8 0****\$ER_NO_ALM. \$ER_CODE9 0****\$ER_NO_ALM. \$ER_CODE10 0****INTEGER RW 0 to 100000**

[Function] Specify the alarms not output

[Description] Specify the alarms that are not output.

Setting : 11 002 (Meaning: SERVO-002 alarm)

Alarm ID Alarm number

Error code output**\$ER_OUT_PUT.\$OUT_NUM 0**
LONG RW 0 to 512

[Function] DO start number for error code output

[Description] Specify the start number for the DOs used for error code output. An error code is output, in binary format, using 33 DOs starting from that having the specified number. If 0 is specified, no error code is output.

\$ER_OUT_PUT.\$IN_NUM 0
LONG RW 0 to 512

[Function] DI number for error code output request

[Description] Every time the DI specified in this variable is set to ON, an error code is output to the DOs specified in \$ER_OUT_PUT.\$OUT_NUM, explained above.

User alarm**\$UALRM_SEV[] 6**
BYTE RW 0 to 255

[Function] User alarm severity

[Description] Sets the user alarm severity. \$UALRM_SEV[i] corresponds to the severity of user alarm [i].

0 WARN

6 STOP.L

38 STOP.G

11 ABORT.L

43 ABORT.G

The initial severity for each user alarm is 6 (STOP.L).

Jogging**\$JOG_GROUP.\$FINE_DIST 0.5**
REAL RW 0.0 to 1.0 (mm)

[Function] Move distance for linear step jogging

[Description] Specify an amount of travel in low-speed linear step feed by Cartesian/tool manual feed. The amount of travel in very low speed step feed is one tenth of the value specified here.

\$SCR.\$FINE_PCNT 10
INTEGER RO 1 to 100 %

[Function] Move distance for joint or orientation step jogging

[Description] Specify an amount of travel for step feed in attitude rotation by axial manual feed or Cartesian/tool manual feed. Specify manual feed with a percentage and an override of 1%.

I/O setting**\$OPWORK.\$UOP_DISABLE ***
BYTE RW 0 / 1

[Function] Enable/disable UOP I/O

[Description] Specify whether the peripheral equipment input signal is enabled or disabled. If the peripheral equipment input signal is enabled when the robot is operated without any peripheral equipment connected, an alarm cannot be cleared. By disabling the signal with this setting, the alarm can be cleared. When any peripheral equipment is connected, set this variable to 0 before using that equipment.

\$SCR . \$RESETINVERT FALSE**BOOLEAN RW TRUE / FALSE**

[Function] FAULT_RESET input signal detection.

[Description] When you set this value to "TRUE", an error is reset by rising edge of FAULT_RESET input signal. If "FALSE" is set, an error is reset by falling edge is detected.

TRUE: Check rising edge of reset input signal.

FALSE: Check falling edge of reset input signal.

\$PARAM_GROUP . \$PPABN_ENBL FALSE**BOOLEAN RW TRUE / FALSE**

[Function] Enable/disable pressure abnormal *PPABN input

[Description] Specifies if pressure abnormal signal is detected or not. If you want to use *PPABN input, you should set this variable to TRUE.

TRUE: Enable

FALSE: Disable

\$PARAM_GROUP. \$BELT_ENBLE FALSE**BOOLEAN RW TRUE / FALSE**

[Function] Belt rupture signal enabled/disabled

[Description] Specify whether the belt rupture signal (RI[7]) is detected. For a robot utilizing the belt rupture signal (A-510, L-1000), this value is automatically set to TRUE.

TRUE: Belt rupture signal enabled

FALSE: Belt rupture signal disabled

Software version**\$ODRDSP_ENB****0****ULONG RW 1 / 0**

[Function] Display of an order file

[Description] An order listing, showing the configuration of the software components installed in the controller can be displayed on the display (order file screen) of the teach pendant.

Soft float function**\$SFLT_ERRTYP****0****INTEGER RW 1 to 10**

[Function] Flag for specifying the alarm to be generated when time-out occurs during follow-up processing of the soft float function

[Description] This variable specifies the alarm (a servo alarm or program pause alarm) to be generated if a time-out occurs during follow-up processing of the soft float function.

0: Generates servo alarm "SRVO-111 Softfloat time out."

1: Generates program pause alarm "SRVO-112 Softfloat time out."

\$SFLT_DISFUP**FALSE****BOOLEAN RW TRUE / FALSE**

[Function] Specifies whether to perform follow-up processing at the start of each motion instruction.

[Description] Specify whether to perform follow-up processing of the soft float function at the start of each program motion instruction.

TRUE: Does not perform follow-up processing at the start of each program motion instruction.

FALSE: Performs follow-up processing at the start of each program motion instruction.

Saving files**\$FILE_APPBCK**

[Function] On the file screen, displays the name of a file to be saved as Application.

\$FILE_SYSBCK

[Description] On the file screen, displays the name of a file to be saved as System file.

Register speed specification function

\$RGSPD_PREXE **FALSE**
BOOLEAN **RO** **TRUE/FALSE**

[Function] Advanced register speed read enabled or disabled

[Description] Specify whether an advanced read of operation statement is performed (enabled) or not (disabled) when the movement speed specified by an operation statement is held in a register.

TRUE: Advanced read enabled

FALSE: Advanced read disabled

**CAUTION**

When an advanced register speed read is enabled with the setting indicated above, the timing at which the register value is changed is important. With some timings, a change in the register value may not be reflected in the operation speed, and the register value existing before the change may be applied to the movement. To enable advanced register speed read, some consideration is needed: The value of a register used for the movement speed during program execution should not be changed; An interlock should be provided.

Specifying an output signal of the BLAL/BZAL alarm

\$BLAL_OUT.\$DO_INDEX **0**
INTEGER **RW** **0 to 256**

[Description] When a non-zero number is specified, DO corresponding to that number is turned on at the occurrence of BLAL/BAZL. DO stays on until the voltage is restored by the replacement backup battery or some other means. (If a program or the I/O screen is used to turn off DO forcibly, DO is turned back on immediately.)

\$BLAL_OUT.\$BATL_M_OR **FALSE**
INTEGER **RW** **TRUE/FALSE**

[Description] Specifies whether to set BATL_M, a dedicated output signal, so that it has also the BZAL/BLAL function.

D SAVING RESEARCH DATA

Overview

This is the procedure manual of saving necessary data for investigating some abnormal status or issues. You can save the research data into the memory card to execute this manual in order. Please send the saving data to FANUC.

Needed Equipment

One flash ATA memory card over 128MB (it just calls memory card in this manual):

Usually research data can be saved when memory card has over 80MB vacant space. However in the case of a lot of files are existed in it, data cannot be saved normally. Then we recommend to use empty memory card.

Method of Saving Research Data

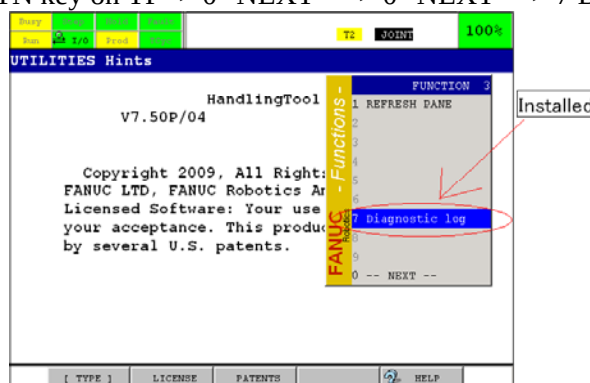


WARNING

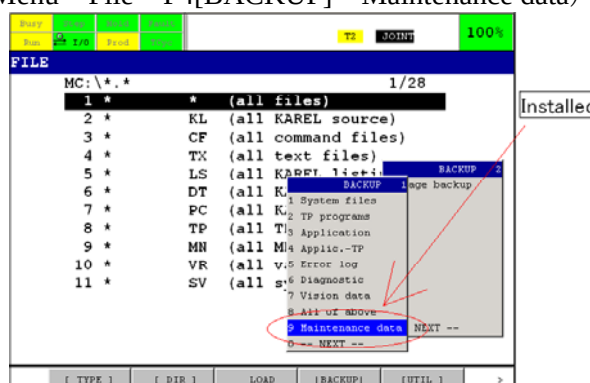
The figure in this manual may be different from your system.

You can save its data by doing next page procedure in order. Procedure is different from if following functions are installed or not. If yes, like following screen is displayed.

Diagnostic Log (Select FCTN key on TP -> 0--NEXT -- -> 0--NEXT-- -> 7 Diagnostic log) :

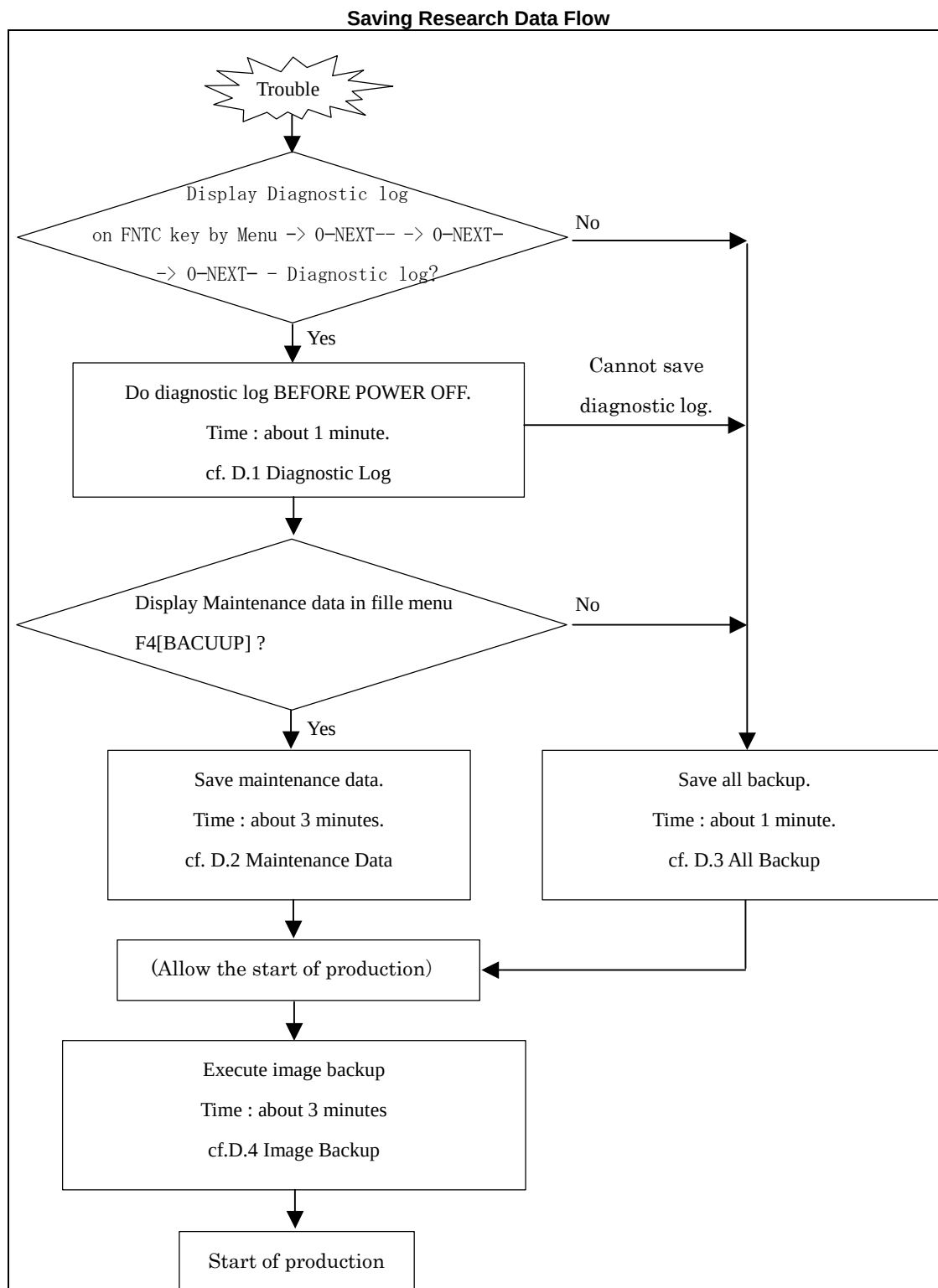


Maintenance Data(Select Menu – File – F4[BACKUP] – Maintenance data) :



WARNING

- 1 Diagnostic log can use after 7DA1P35, 7DA3P17 or 7DA4P01.
- 2 Maintenance data can use after 7DA3P30, 7DA4P12 or 7DA3P04.

**WARNING**

- 1 Time may differ largely in your system.
- 2 If you have time after doing maintenance data or all backup. please do image backup before starting production.

D.1 DIAGNOSTIC LOG

⚠ CAUTION

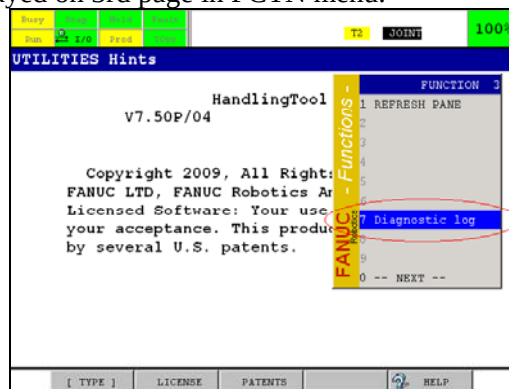
If you execute this function, verify that the status of robot is STOP. In some situations robot becomes jerky motion while saving log.

⚠ WARNING

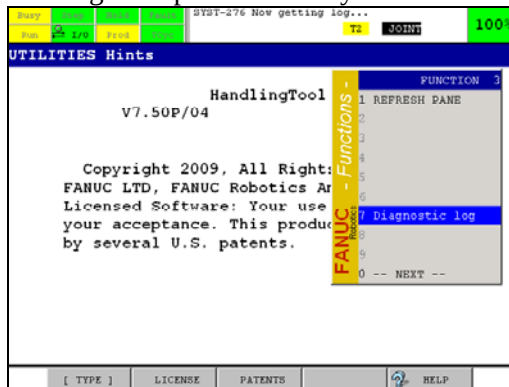
Do below operations BEFORE POWER OFF at the time of a trouble occurs. If you power off, available data for investigating is lost.

Data is save into FROM, therefore you do not have to insert the external memory such as memory card.

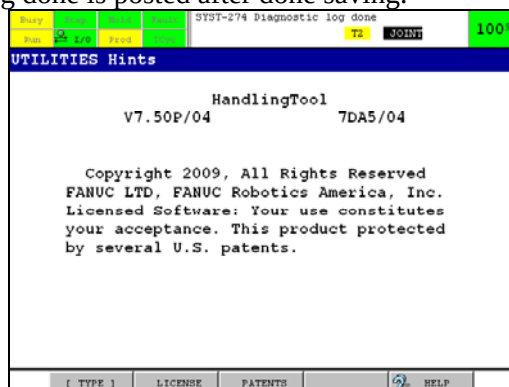
- 1 Select FCTN -> 0--NEXT-- -> 0--NEXT-- -> 7 Diagnostic log.
- 2 "Diagnostic log" is displayed on 3rd page in FCTN menu.



- 3 Set the cursor onto "Diagnostic log" and press Enter key. SYST – 276 Now getting log... is posted.



- 4 It takes about 1 minutes to save.
- 5 SYST – 274 Diagnostic log done is posted after done saving.



CAUTION

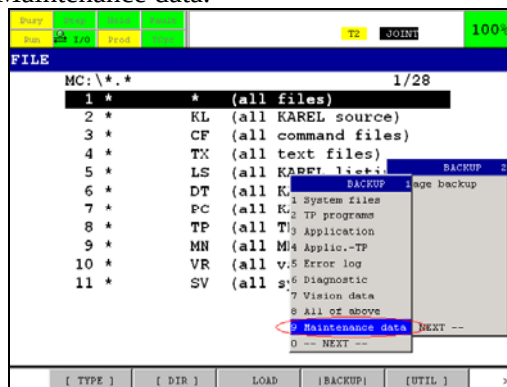
- 1 If you see “SYST –273 Lack of FRA for saving” or “SYST – 275 Diagnostic log failed”, please save all backup before power off, do image backup and send its data to FANUC. At this time please notify the procedure of duplication its problem too.
- 2 Depending on the condition of occurring problem, this function is not available. For example, in the case TP is frozen and any key operation are ignored.
- 3 If you execute this function again, data is overwritten. Diagnostic log is saved only 1 time.

D.2 MAINTENANCE DATA

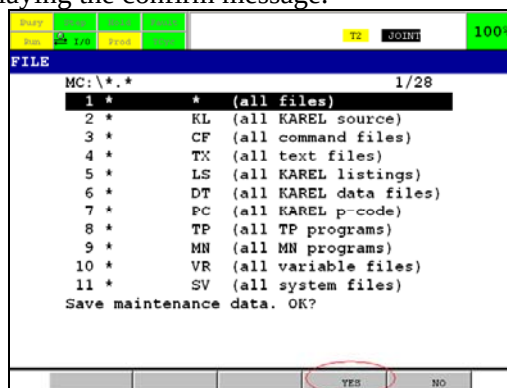
CAUTION

Saving time of maintenance data depends on the system. It takes about 3 minutes as a rough guide. In the case that a lot of TP program or many directories for auto backup are existed, saving time get longer.

1. Insert the memory card to the controller.
2. Select Menu – 7 FILE.
3. If MC: is not displayed in the upper left portion on the screen, select F5[UTIL] – Set Device – Mem Card(MC:).
4. Select – F4[BACKUP] – Maintenance data.



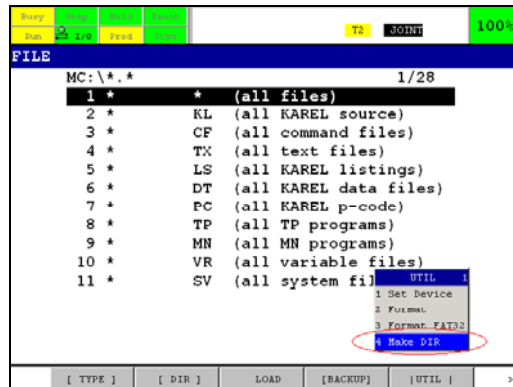
5. Select F4[YES] after displaying the confirm message.



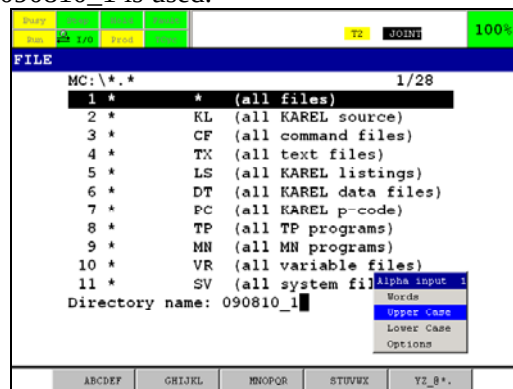
6. In above sample case, maintenance data is saved into MC:¥MNT_DATA¥. It is saved into MNT_DATA¥ under the current directory. MNT_DATA is newly created.

D.3 ALL BACKUP

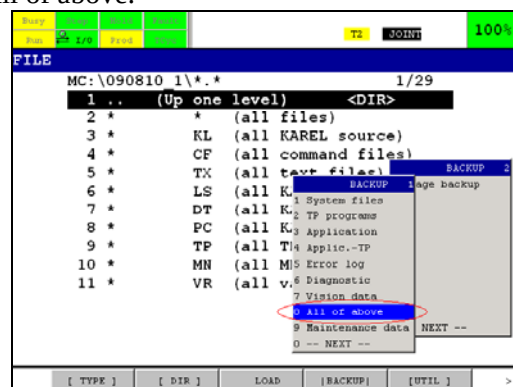
1. Insert the memory card to the controller.
2. Select Menu – 7 FILE.
3. For prevention of miss operation that deletes the data existed in the card, new directory is created. Select F5[UTIL] – Make DIR.



4. Input the directory name. For convenience of explanation, directory name is set “Year, Month, Day, Number”. In this sample 090810_1 is used.



5. Select F4[BACKUP] – All of above.



6. Select F4[YES] after confirm message for deleting is displayed.

MC:\090810_1*. * 1/29

1	..	(Up one level)	<DIR>
2	*	(all files)	
3	*	KL (all KAREL source)	
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)	
11	*	VR (all variable files)	

Delete MC:\090810_1\ before backup files

YES NO

7. Select F4[YES] after confirm message for saving is displayed.

MC:\090810_1*. * 1/29

1	..	(Up one level)	<DIR>
2	*	(all files)	
3	*	KL (all KAREL source)	
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)	
11	*	VR (all variable files)	

Delete MC:\090810_1\ and backup all file

YES NO

8. All backup is saved.

9. Select F2[DIR] - *. * after done saving.

MC:\090810_1*. * 1/29

1	..	(Up one level)	<DIR>
2	*	(all files)	
3	*	KL (all KAREL source)	
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)	
11	*	VR (all variable files)	

Press DIR to generate directory

[TYPE] [DIR] [LOAD] [BACKUP] [UTIL] >

10. Verify that all backup is saved normally.

MC:\090810_1*. * 1/200

1	..	(Up one level)	<DIR>
2	-BCKED8-	TP	74
3	-BCKED9-	TP	74
4	-BCKEDT-	TP	74
5	AMBERLED	GIF	1046
6	ATERRJOB	VR	46
7	BACKDATE	DT	274
8	C1	PC	587
9	C1	VR	40
10	C10	PC	589
11	C10	VR	41

[TYPE] [DIR] [LOAD] [BACKUP] [UTIL] >

See “Batched save” in the section “FILE INPUT/ OUTPUT”.

D.4 IMAGE BACKUP

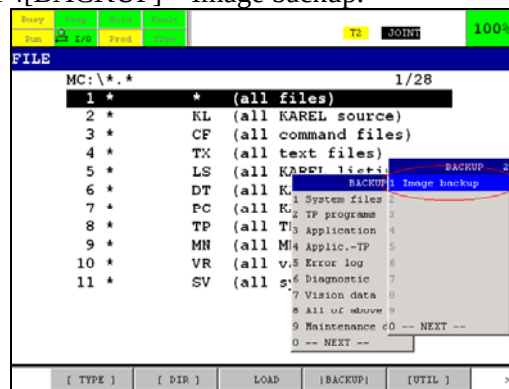
Please execute image backup after doing diagnostic log. Image backup allows the diagnostic log data to save external device such as memory card.



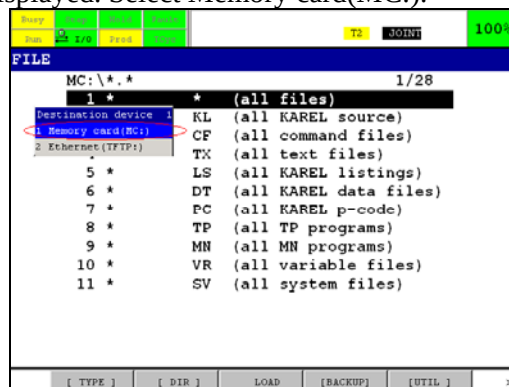
CAUTION

Executing image backup, cycle power is executed. Please verify whether cycle power is done or not.

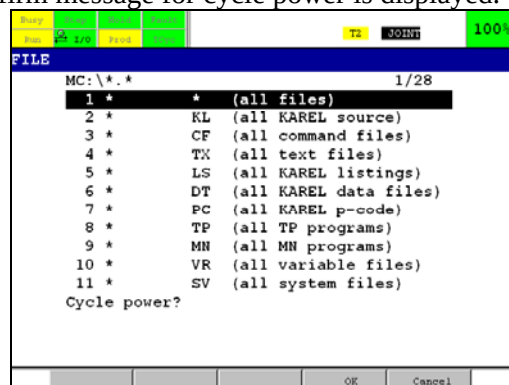
1. Insert the memory card to the controller.
2. Set the teach pendant enable switch to ON.
3. Select Menu – 7 FILE – F4[BACKUP] – Image backup.



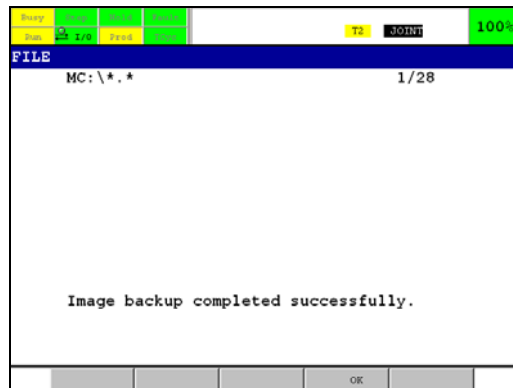
4. [Destination Device] is displayed. Select Memory card(MC:).



5. Select F4[YES] after confirm message for cycle power is displayed.



6. Please cycle power. Image backup is started automatically. Please do not push any key until done the image backup.
7. "Image backup completed successfully." is displayed after done the image backup. Select F4[OK].



8. Data are saved like FROM**.IMG, SRAM**.IMG (** is number) in the memory card. For example, FROM32MB, SRAM2MB system is saved 34 files like FROM00.IMG, FROM01.IMG...FROM31.IMG, SRAM00.IMG, SRAM01.IMG.
9. Select Menu – 7 FILE – F2[DIR] - *.*. Verify that FROM**.IMG, SRAM**.IMG are saved.

See the Subsection Image Backup Function under the Section FILE INPUT/OUTPUT.

INDEX

<A>

Abort Instruction	196
Abort program by UOP	660,676
Abort the program	742
About Reducer Diagnosis.....	1001
Absolute (ABS)	837
Addition of button change control.....	706
Addition of command button control	710
Addition of edit box control	711
Addition of label control	712
Addition of toggle button control	712
Addition of toggle lamp control	714
Additional Motion Instructions	149,966
Adjustment	641
ADVANCED CONSTANT PATH	567
ALARM	648
Alarm ID	682
Alarm Number	682
ALARM RESET	648
Alarms	829
All BACKUP	1047
Analog I/O.....	38
Analog I/O Instructions	174
Angle-input Shift Function.....	419
APPEARANCE AND OPERATIONS.....	878
Appearance and Switches.....	878
Appendix	643
Application.....	810
APPLICATION TOOL SOFTWARE	5
Arguments	180
Arithmetic Palletizing Register Instructions	167
ASCII File	358
ASCII File Example.....	768
ASCII file general event information	763
ASCII file specific event information	763
ASCII program with LINE REMARK.....	809
ASCII save	368
ASCII UPLOAD FUNCTION	394
Asynchronous Operation Group Instruction.....	204
ATPERCH	652,672
Attention	633
Auto Position Renumbering	286
AUTOMATIC BACKUP.....	381
AUTOMATIC ERROR RECOVERY FUNCTION.....	466,597
AUTOMATIC ERROR RECOVERY FUNCTION FOR MULTI-TASK	597
AUTOMATIC OPERATION	315
Automatic Operation (Operation Execution)	6
Automatic Operation by Robot Start Request (RSR).....	316
Automatic Operation with Program Number Selection (PNS).....	318
AUXILIARY AXIS SERVO OFF (LOCAL STOP) FUNCTION	626

Available iPendant Controls	701

BACKGROUND EDITING	273
Background operation of Math function	838
Backlight automatic blanking	893
Backup/restore	724
Backward execution.....	826
Basic operation.....	921
Basic tag.....	518
BATTERY ALARM.....	648
Behavior of main program and subprogram.....	588
BRAKE CHECK FUNCTION.....	695
BRANCH INSTRUCTIONS	175

<C>

Calibration Procedure (for 6-Axis Robots)	499
Cause and Remedy for alarm occurrence.....	785
Caution.....	503,697,773,934
Caution and Limitations.....	458
Caution for creation of KAREL program.....	722
Cautions	492
Cell Interface I/O	77
Change Display Format of Cycle Time Data	790
Changing a Control Instruction	252
Changing a Motion Instruction	243
CHANGING A PROGRAM.....	242
Changing a Standard Motion Instruction	228
Changing Conditions for Executing the Resume Program	480,619
Changing Program Information	269
Changing the Operation Target Screen	885
CMDENBL#1, #2	651
CMDENBL#1~#N	671
CMOS programs	778
COL DETECT ON / COL DETECT OFF	489
COL GUARD ADJUST.....	490
Cold Start.....	980
COLLISION DETECTION for AUXILIARY AXIS.....	503
Collision Guard Setup Screen	488
Color Display According to the Alarm Severity	892
Comment Instruction.....	199
Comment tag.....	519
Communication	19
Compatibilities and limitations	573
Conditional Branch Instructions.....	177,969
Conditional Wait Instructions	188
Configuration	448
CONNECT/ISOLATE Key	649
Consideration	809
Constant path with respect to program speed changes	573
Constraints	629
CONTINUOUS ROTATION FUNCTION.....	434
Continuous Test	308
Controlled Start.....	978

CONTROLLER	8	Edit REMARKED instructions	808
Controlling the Palletizing Function by a Palletizing Register.....	871	Effect of alarm occurrence	786
COORDINATE SYSTEM CHANGE SHIFT		Emergency Stop Devices	20
FUNCTIONS.....	424	Enable or disable all event logging	761
Copy programs	783	ENDFOR statement execution	821
Corner Region Termination Type (CRy)	571	ENHANCED MIRROR IMAGE	724
CREATING A PROGRAM	223	Entering Distance before.....	455
Creation of run button	722	Error code of torque limit multi-axis setup function	772
CRT/KB	19	Error Code Output function	679
CRy versus segment distance	572	ERROR SEVERITY TABLE	592
CSTOPI#1~#N	669	Error Severity Table Overview	592
Current edit line display	906	Event class selection screen	758
CURRENT POSITION	336	Event detail selection screen	760
Current position display	891	Example 1	1008
Cursor positioning	903	Example 2	1009
Custom Menu	748	Example 3	1009
Cut/Copy paste of control	718	Example ASCII File.....	397
Cycle Mode	788	Examples.....	767,826
Cycle Power	747	Exceptions and Restriction.....	840
Cycle Stop Signal (CSTOPI).....	591	EXECUTING A PROGRAM	290,296
Cycle Time Categories	786	Executing Macro Instructions	405
Cycle Time Screen Items	787	EXECUTING THE PALLETIZING FUNCTION.....	869
CYCLE TIME TRACKING	786	Execution example of Spot program	793
		Execution examples	826
<D>		EXECUTION HISTORY	342
Data File	358	Execution of the Resume Program from The Teach	
DATA MONITOR	683	Pendant and Test Mode	480,618
Data Monitor Chart	694	Exponent	836
Data Monitor Schedule	690	Extended Alarm Log.....	557
Data Monitor Setup	685	Extended Axis	20
Data Offset function.....	812	EXTENDED AXIS SETUP	1016
Default Logic File	357	External Override Selection Function	320
Default Menu Type tag	521		
Default Screen tag	522	<F>	
Defining a Resume Program	469	FANUC iPendant	877
Definition of FSSB line.....	1006	Fast Exit/entry Feature	598
Delete of control.....	717	FAULT#1, #2.....	653
Delete set.....	751	FAULT#1~#N.....	672
Detail of KAREL Config	745	FCTN/Display Menu Access tag.....	522
DIAGNOSIS SCREEN	1001	Features tag	521
DIAGNOSTIC LOG	596,1045	Feedrate.....	146
Digital I/O	30	FILE INPUT/OUTPUT.....	348
Digital I/O Instructions	172	FILE INPUT/OUTPUT UNITS	348
Disabling the Password Function	510	File Manipulation	366
Display Modes of Cycle Time	788	FILE MEMORY	398
Display of panel	723	FILE programs	779
Display Reference Cycle Time Line	792	FILES.....	357
Display Screen of the Teach Pendant.....	14	Filtered Program List	288
Display Target Cycle Time Line.....	791	Finish setting.....	930
DISTANCE BEFORE FUNCTION.....	447	FINISHING FUNCTION PACKAGE	809
DUAL MOTOR DRIVE	643	Fixed Program Name	287
Dump selections screen	754	Flow of Stitch process.....	639
		Flowchart for Resuming a Suspended Program	477
<E>		FOR statement execution	821
Each Item	1003	FOR/ENDFOR STATEMENT	820
EDCMD Access tag	523	FOR/ENDFOR statement Combination	824
[EDCMD] with LINE REMARK.....	808	FOR/ENDFOR statement specification	820

Forced Output	311	Input and Output signals	679
Form of ENDFOR statement.....	821	Input/Output.....	19
Form of FOR statement.....	820	Inputting Initial Data.....	850
FORMAT OF A SYSTEM VARIABLE TABLE.....	1025	Instruction	448,637
FRAME INSTRUCTIONS	194	Instruction format of assignment statements	832
Frame Setup Instruction	972	Instruction format of math function	832
FSSB LINE SETUP	1006	Instruction format of relational statements.....	832
Function	805	Instruction format of wait command statements	833
Function specification of Math functions.....	833	INTERFERENCE PREVENTION AREA FUNCTION	116
Functions.....	587	Internet Browser Screen.....	886
<G>		Interruption Disable	132
General.....	503	INTRODUCTION	1,626,643,645,661,663,679,770
GRAVITY COMPENSATION	505	Inverse trigonometric function (ACOS)	835
Group I/O	34	Inverse trigonometric function (ASIN)	834
Group I/O Instruction.....	175	Inverse trigonometric function (ATAN)	835
Group Mask.....	132	Inverse trigonometric function (ATAN2)	835
GROUP MASK EXCHANGE.....	841	Item select screen.....	926
<H>		Item setting	927
Halt and Forced Termination	590	<J>	
Halt by a Hold and Recovery	292	Jig Mastering.....	984
Halt by an Emergency Stop and Recovery.....	291	Jog Feed of the Robot	5
Halt Caused by an Alarm	293	JOINT OPERATING AREA	112
Halt Instruction	196	<K>	
Hardware	753	KAREL Config	739
Hardware and Software	753	KAREL program example	767
Hardware and software requirements.....	753	KAREL PROGRAM EXECUTION HISTORY	
Hardware Requirements.....	698	RECORD.....	752
HELD#1, #2.....	653	KAREL USE SUPPORT FUNCTION	738
HELD#1~#N	672	Key Switches	880
HIGH PERFORMANCE SOFTFLOAT FUNCTION	580	Knowhow of Cartesian Softfloat.....	584
HIGH-SENSITIVITY COLLISION DETECTION	487	Knowhow of Softfloat.....	583
HOLD	648	<L>	
HOLD#1, #2.....	650	Label Instruction	176
HOLD#1~#N.....	669	LEDS ON THE TEACH PENDANT	328
Hot Start.....	981	Level name tag.....	519
Hourly Mode	789	Limitation and Caution of KAREL Config.....	746
How it Works.....	568	Limitations	570,577,578,642,697,773,774
How to display alarm log.....	558	LINE NUMBER, PROGRAM END SYMBOL, AND	
How to start programs in multitasking	587	ARGUMENT	135
How to teach FOR/ENDFOR statement.....	821	LINE REMARK INSTRUCTION	805
How to use linear distance	570	Linear Distance	567
<I>		Line-by-Line Mode.....	789
I/O	21	List for CALL instruction edit	905
I/O CONNECTION FUNCTION.....	57	List for MACRO instruction edit	905
I/O INSTRUCTIONS.....	171	LIST OF MENUS	937
I/O Link List Screen.....	53	LIST OF PROGRAM INSTRUCTIONS	962
I/O Link SCREEN.....	53	List selected tasks screen	758
ICON editor.....	900	LOAD ESTIMATION	496
ICON menu	896	Load Estimation Procedure (for 6-Axis Robots).....	496
IMAGE BACKUP.....	1049	LOAD SETTING	493
IMAGE BACKUP FUNCTION.....	387	Loading a Specified Program File Using the File Screen	
INDEPENDENT ADDITIONAL AXIS SETUP	1021		371
Initial Setting.....	503	Loading an ASCII Teach Pendant Program from the	
Initial setting before use.....	695	Teach Pendant	394
Initial Start	977	LOADING FILES.....	369

Loading process in program execution.....	784
Loading Using the Program List Screen	370
Local Labels tag	520
Logging Events	762
Logging events to an ASCII file.....	763

<M>

MACRO INSTRUCTION.....	400,973
MAINT_PROG Instruction.....	602
MAINTENANCE DATA.....	1046
Make backup of programs	783
MANUAL I/O CONTROL	310
Manual Operation Screen of the Automatic Error Recovery Function	478
Manual Operation Screen of the Resume Program Function.....	616
MANUAL PLAN	1
Mastering	506,982
Mastering at the Zero-degree Positions.....	985
MATH FUNCTION INSTRUCTION.....	831
Max Speed.....	578
Maximum Speed Instructions.....	202
Meaning of Alarm Code.....	680
Memory Card.....	351
MEMORY USE STATUS DISPLAY.....	344
Message Instruction	199
Method of operating.....	662
Mirror Image of Extended Axes.....	730
Mirror Image using Existing Frames and Mirror Planes, with Controlled Orientation.....	729
Mirror Shift Function	416
MIXED LOGIC INSTRUCTION	528
Model B Unit List Screen.....	55
Modification and Addition of Signal.....	650,667
Modification of control	716
Modification of page	719
Modification of panel.....	715
Modifying Error Severity	592
MODIFYING THE PALLETIZING FUNCTION.....	872
Monitor	589
Motion Format	137
Motion Group Instructions	974
Motion Instruction Enhanced Editing (J560)	810
Motion Instruction Insert and Instruction delete function	810
MOTION INSTRUCTIONS	136,207,966
Motion of the Robot.....	19
Motion Performance Screens	493
MOTION Screen.....	505
Moving the Robot by Jog Feed	215
MULTI ROBOT CONTROL FUNCTION	645
Multi UOP Interface Function.....	663
MULTIAXIS CONTROL INSTRUCTIONS.....	203,973
Multi-language Comment Instruction	199
MULTITASKING FUNCTION.....	587

<N>

Natural Logarithm.....	836

Notes on main program and subprogram	589
Notes on program creation	587
Notes on Teaching the Palletizing Function.....	868
Number of ports Setting Screen	56

<O>

OFFSET CONDITION INSTRUCTION	193
ONLINE POSITION MODIFICATION.....	321
Operating Procedure	496
OPERATING THE HAND MANUALLY	314
OPERATION GROUP DO OUTPUT FUNCTION	441
OPERATION GROUP INSTRUCTIONS	204
OPERATION LOG BOOK.....	547
Operation panel	662,920
Operation panel setting screen	924
Operational improvement in 7DA5 or later series	895
Operational improvement in 7DA7 or later series	908
Operational Restrictions.....	894
Operations	553
Operator Panel	17
Operator panel status display	891
OPERATOR'S PANEL I/O	51
ORIGINAL PATH RESUME	560
Other	677
OTHER EDITING FUNCTION	286
OTHER INSTRUCTIONS.....	197,971
Other Related Matters	502
OTHER SETTINGS	127
Other specifications.....	640
Other Specifications and Restrictions	481,619
Outline	580,1001
Outline of the Automatic Error Recovery Function	467
Override Instruction	198
OVERVIEW 4,394,447,466,487,493,496,515,547,567, 587,597,637,695,698,738,752,767,773,786,805,820,877	
Overview of Automatic Backup.....	381
Overview of the Password Function	506

<P>

PALLETIZING ALL-POINT TEACHING	874
Palletizing End Instruction.....	162
PALLETIZING FUNCTION	845
PALLETIZING FUNCTION WITH EXTENDED AXES	873
Palletizing Instruction	161
PALLETIZING INSTRUCTIONS	161,847,975
Palletizing Motion Instruction.....	162
Palletizing Register	870
PALLETIZING REGISTERS.....	334
Panel enable condition setting.....	929
Panel Setup	924
PANEL WIZARD.....	698
Parallel Mirror Image.....	726
Parameter Instruction	200
Password Auto Login Function.....	527
Password Configuration File.....	515
PASSWORD FUNCTION.....	506
Password Log.....	524

Password Operations by Program Users and Setup Users	511	PROGRUN#1, #2, PAUSED#1, #2	652
Password Operations by the Install User	508	PROGRUN#1~#N, PAUSED#1~#N	671
PATH SWITCHING FUNCTION	621		
Pause program by Operator Panel	659	<Q>	
Pause program by Teach Pendant	659	Quick Mastering	987
Pause program by UOP	659, 676		
Perform Automatic Backup (Prior to V7.40)	383	<R>	
Perform Automatic Backup (V7.40 or later)	384	Recorded Events	549
Performance	754	Re-creation of panel	720
Peripheral I/O	19, 44, 665	Register and I/O Instructions	967
PMC	668	REGISTER INSTRUCTIONS	162, 163
PMC EDIT FUNCTION	545	Register panel	931
PMC MONITOR FUNCTION	543	Registering a Program	224
PNSTROBE#1, #2, PROD_START#1, #2	651	REGISTERS	329
PNSTROBE#1~#N, PROD_START#1~#N	671	Relation to Current Softfloat Function	580
Position Data	141	REMOTE	648
Position Register Axis Instructions	165	Remote Controller	18
Position Register Instructions	164	REMOTE TCP FUNCTION	482
POSITION REGISTER LOOK-AHEAD EXECUTION		REMOTE/LOCAL Key	648
FUNCTION	439	Restore the Backup	386
Position Register Look-ahead Execution Instruction	973	Restriction of teaching math function	840
POSITION REGISTERS	331	RESTRICTIONS	934
POSITIONER SETUP	1010	Results of brake check	697
Positioning Path	148	Resume Program Function	597
Precaution	785	RESUME_PROG Instruction	599
Predefined Position	208	Resuming a Program	299
PRINTING FILES	376	RETURN_PATH_DSBL Instruction	601
Procedure	1002	ROBOT	7
Procedure of operation	580	ROBOT AXIS STATUS	996
Process speed	575	ROBOT I/O	41, 653
Prog Select Screen	71	Robot I/O Instructions	173
Program	5	Robot Isolation function	661
Program Comment	131	Robot Motion	297
PROGRAM CONTROL INSTRUCTIONS	196, 971	Robot Service Request (RSR)	62, 658, 676
PROGRAM DETAIL INFORMATION	130	ROBOT TOOLS OF ROBOT HOMEPAGE	794
Program Edit Instructions	253	Rotational Mirror Image	728
Program edit screen display when operating motion		Round off (ROUND)	837
group is changed	675	RSR	651, 670
Program End Instruction	176	RSR Instruction	197
Program exchange function without enough memory		Run KAREL program	741
space	784	Run KAREL Program by Panel	721
Program Execution Instruction	203	Run program	656, 676
Program File	357	Run program by operator panel	657
PROGRAM HALT AND RECOVERY	290	Run program by Teach Pendant	656
Program Instructions	489, 494, 966	Run two programs at the same time	657
Program Look/Monitor	309		
Program Name	130	<S>	
Program Number Select (PNS)	658, 676	SAFETY PRECAUTIONS	s-1
Program Number Selection (PNS)	65	Safety signal status display	892
PROGRAM OPERATION	269	SAFETY SIGNALS	345
Program Shift Function	410	Save / Load LS files	783
PROGRAM STATUS	346	Save / Load programs	783
PROGRAM STRUCTURE	128	Save / Load TP files	783
PROGRAM TIMER	340	Save and restore setting value	930
PROGRAM TOOLBOX	564	Save Cycle Time Data	790
PROGRAMMING	206, 633, 693	Saving all the Program Files Using the File Screen	360
		SAVING FILES	358
		SAVING RESEARCH DATA	1043

Saving with a Function Menu	364	SINGULAR POINT CHECK FUNCTION	285
Saving with Program List Screen	359	Skip and Offset Condition Instruction.....	972
Screen Access tag.....	520	SKIP CONDITION INSTRUCTION.....	191
Screen maximization.....	907	SOFT FLOAT FUNCTION	428
Screen Restrictions According to Password Level.....	525	Soft Float Instruction	974
Screen Selection Menu and Screen Menus on the Edit		Soft Limit Setting.....	564
Screen.....	889	Software	753
SCREEN/PROGRAM INSTRUCTIONS	937	Software Options.....	647,664
Select program	654,672	Software parts ID, Screen ID	517
Select program at running or paused	658,674	Software Requirements and Explanation	698
Selecting a Palletizing Instruction.....	849	SOFTWARE VERSION	994
Selecting a Program	242	SOP	662
Set custom menu	749	Special Jog Sequence (J559).....	809
Setting 1 (FSSB line)	1007	SPECIAL OPERATION	976
Setting 2 (Number of total axes on FSSB line 1)	1007	Specification	447,487,627,637,646,664,679
Setting 3 (Hardware start axis).....	1007	Specifying Test Execution	304
SETTING A COMMUNICATION PORT.....	353	Splitting the Screen	883
Setting a Jog Coordinate System.....	103	Square root (SQRT)	833
SETTING A REFERENCE POSITION.....	108	Stack Size.....	133
Setting a Tool Coordinate System.....	83	Standard Operator Panel	647
Setting a User Coordinate System.....	93	START	648
SETTING AUTOMATIC OPERATION	61	Start brake check.....	696
SETTING COORDINATE SYSTEMS.....	81	START MODE	976
Setting Macro Instructions	401	Start mode config.....	744
Setting Mastering Data.....	992	Start the KAREL Config screen.....	739
Setting of Automatic Backup	382	Start Up Methods	976
Setting of motion groups intended by each UOP set....	673	START#1, #2	651
Setting of the Automatic Error Recovery Function.....	605	START#1~#N	670
Setting of the number of UOP set	666	Starting a Program	296
Setting Path Pattern Conditions.....	862	Starting custom menu.....	748
Setting the Automatic Error Recovery Function	470	STATE MONITORING FUNCTION.....	458
SETTING THE GENERAL ITEMS	126	STATUS DISPLAY	328
Setting up	698	Status Monitoring Instructions	974
Setting Up Book.....	553	Status Sub-window	890
Setting up events	762	Status Window	882
Setting up Fast Label.....	702	Step Test	305
Setting up fast lamp.....	703	Stitch condition	637
Setting up fast switch	705	STITCH FUNCTION.....	637
SETTING UP iPendant	910	Stop Logging Tasks Screen.....	757
Setting up TCP speed output.....	775	Stop program.....	659,676
Setting up the KAREL program execution history		Storage Configuration	779
record.....	754	Storages.....	778
SETTING UP THE ROBOT SYSTEM	21	String Register, String Instructions	169
Settings.....	487,629	STRING REGISTERS	335
Setup	485,557,644,810	STYLE	68
Setup and Operations	754	Sub program display	906
Setup Examples.....	1008	Subdirectories	379
Setup for ROBOT TOOLS.....	796	Subtype	131
Setup HTTP AUTHENTICATION.....	799	Summary	482,920
Setup TCP/IP.....	796	Synchronous Operation Group Instruction.....	204
Severity of Alarm.....	681	SYSRDY#1~#N	671
SHADOW ONDEMAND programs	779	SYRSY#1, #2.....	652
SHADOW programs	778	SYSTEM CONFIG MENU	120
SHIFT FUNCTIONS	409	System File/Application File.....	358
Simulated I/O	312	System Setting	5
SIMULATED INPUT SKIP FUNCTION.....	59	SYSTEM TIMER	342
Single Axis Mastering.....	989	System variable: \$ROBOT_ISOLC	662

SYSTEM VARIABLES.....	338,505,661,677,1025,1027
<T>	
Task selection screen.....	755
TCP SPEED OUTPUT.....	773
TCP speed output instruction.....	776
Teach FOR/ENDFOR statement.....	821
Teach LINE REMARK instruction.....	806
Teach Math function instruction.....	838
Teach Pendant.....	9
Teach Pendant Program Example.....	768
Teaching a Control Instruction.....	235
Teaching a Motion Instruction.....	230
Teaching a Path Pattern.....	865
Teaching a Stacking Pattern.....	856
Teaching an Supplementary Motion Instruction.....	232
Teaching and modifying position.....	656
Teaching techniques.....	573
TEACHING THE PALLETIZING FUNCTION.....	849
Teaching the RETURN_PATH_DSBL Instruction.....	470
Teaching to motion instruction.....	663
Test Operation (Test Execution).....	6
TESTING.....	303
The difference from Multi Robot Control function about	
UOP.....	664
The way to make original program name of program	
creation screen into initial strings.....	288
The way to register over six initial strings.....	288
Three-Mode Switch.....	211
TIME BEFORE FUNCTION.....	442
Timer Instruction.....	198
Time-specified Wait Instruction.....	188
TIPS ON EFFECTIVE PROGRAMMING.....	207
Toggling robot.....	654
TOOL OFFSET CONDITION INSTRUCTIONS.....	194
TORQUE LIMIT FUNCTION.....	770
Torque limit function feature.....	770
Torque Limit Multi-axis Setup Function.....	772
Touch Panel.....	881
TP DRAM/FILE STORAGE FUNCTION.....	778
TP ENABLED.....	648
TP Start Prohibition.....	240
Tracking Cycle Time.....	790
Trigonometric function (COS).....	834
Trigonometric function (SIN).....	833
Trigonometric function (TAN).....	834
Truncate (TRUNC).....	837
Tuning Procedure.....	503
TURNING ON THE POWER AND JOG FEED.....	209
Turning On the Power and Turning Off the Power.....	209
Type of Math functions.....	831
Types of Alarms.....	679
TYPES OF SCREENS.....	943
<U>	
Unconditional Branch Instructions.....	176,970
Update Cycle Time Data.....	791
Usable Memory Cards.....	382
Usage.....	723,809
Usage of ROBOT TOOLS.....	802
USB Port (Soft option, New iPendant only).....	894
Use KAREL Config screen.....	740
USER ALARM.....	113
User Alarm Instruction.....	197
User Authentication.....	923
User Operator Panel(UOP).....	649
USER SCREEN.....	329
USER1/USER2.....	649
UTILITY.....	399
<V>	
VARIABLE AXIS AREAS.....	115
Version Management.....	386
Viewing ASCII Upload Errors.....	395
<W>	
Wait Instruction.....	970
WAIT INSTRUCTIONS.....	188
Wait Release.....	313
Warnings.....	482
Warnings (Be sure to read this section for safety.).....	620
Wide screen mode.....	907
WORLD FRAME ORIGIN.....	1005
Write Protection.....	132
<X>	
XML file for Password function.....	517

REVISION RECORD

Edition	Date	Contents
01	July., 2011	

B-83134EN-1/01



* B - 8 3 1 3 4 E N - 1 / 0 1 V 2 *