FANUC Robotics SYSTEM R-30*i*A EtherNet/IP Setup and Operations Manual

MAROC77EN01101E Rev A

Version 7.70

About This Manual

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

Copyrights and Trademarks

FANUC Robotics America, Inc 3900 W. Hamlin Road Rochester Hills, Michigan 48309-3253

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

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

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

Warning

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

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

FANUC Robotics America, Inc Training Department 3900 W. Hamlin Road
Rochester Hills, Michigan48309-3253

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

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

Patents

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

FRA Patent List

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

FANUC LTD Patent List

4,571,694 4,626,756 4,700,118 4,706,001 4,728,872 4,732,526 4,742,207 4,835,362 4,894,596
4,899,095 4,920,248 4,931,617 4,934,504 4,956,594 4,967,125 4,969,109 4,970,370 4,970,448
4,979,127 5,004,968 5,006,035 5,008,834 5,063,281 5,066,847 5,066,902 5,093,552 5,107,716
5,111,019 5,130,515 5,136,223 5,151,608 5,170,109 5,189,351 5,267,483 5,274,360 5,292,066
5,300,868 5,304,906 5,313,563 5,319,443 5,325,467 5,327,057 5,329,469 5,333,242 5,337,148
5,371,452 5,375,480 5,418,441 5,432,316 5,440,213 5,442,155 5,444,612 5,449,875 5,451,850
5,461,478 5,463,297 5,467,003 5,471,312 5,479,078 5,485,389 5,485,552 5,486,679 5,489,758
5,493,192 5,504,766 5,511,007 5,520,062 5,528,013 5,532,924 5,548,194 5,552,687 5,558,196
5,561,742 5,570,187 5,570,190 5,572,103 5,581,167 5,582,750 5,587,635 5,600,759 5,608,299
5,608,618 5,624,588 5,630,955 5,637,969 5,639,204 5,641,415 5,650,078 5,658,121 5,668,628
5,687,295 5,691,615 5,698,121 5,708,342 5,715,375 5,719,479 5,727,132 5,742,138 5,742,144
5,748,854 5,749,058 5,760,560 5,773,950 5,783,922 5,799,135 5,812,408 5,841,257 5,845,053
5,872,894 5,887,122 5,911,892 5,912,540 5,920,678 5,937,143 5,980,082 5,983,744 5,987,591
5,988,850 6,023,044 6,032,086 6,040,554 6,059,169 6,088,628 6,097,169 6,114,824 6,124,693
6,140,788 6,141,863 6,157,155 6,160,324 6,163,124 6,177,650 6,180,898 6,181,096 6,188,194
6,208,105 6,212,444 6,219,583 6,226,181 6,236,011 6,236,896 6,250,174 6,278,902 6,279,413
6,285,921 6,298,283 6,321,139 6,324,443 6,328,523 6,330,493 6,340,875 6,356,671 6,377,869
6,382,012 6,384,371 6,396,030 6,414,711 6,424,883 6,431,018 6,434,448 6,445,979 6,459,958
6,463,358 6,484,067 6,486,629 6,507,165 6,654,666 6,665,588 6,680,461 6,696,810 6,728,417
6,763,284 6,772,493 6,845,296 6,853,881 6,888,089 6,898,486 6,917,837 6,928,337 6,965,091
6,970,802 7,038,165 7,069,808 7,084,900 7,092,791 7,133,747 7,143,100 7,149,602 7,131,848
7,161,321 7,171,041 7,174,234 7,173,213 7,177,722 7,177,439 7,181,294 7,181,313 7,280,687
7,283,661 7,291,806 7,299,713 7,315,650 7,324,873 7,328,083 7,330,777 7,333,879 7,355,725

7,359,817 7,373,220 7,376,488 7,386,367 7,464,623 7,447,615 7,445,260 7,474,939 7,486,816 7,495,192 7,501,778 7,502,504 7,508,155 7,512,459 7,525,273 7,526,121

VersaBell, ServoBell and SpeedDock Patents Pending.

Conventions

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

Warning

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

Caution

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

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

Contents

List of Figures

List of Tables

Safety

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

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

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

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

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

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

CONSIDERING SAFETY FOR YOUR ROBOT INSTALLATION

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

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

Keeping People and Equipment Safe

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

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

Using Safety Enhancing Devices

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

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

Setting Up a Safe Workcell

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

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

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

Staying Safe While Teaching or Manually Operating the Robot

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

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

Warning

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

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

Warning

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

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

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

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

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

Staying Safe During Automatic Operation

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

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

Staying Safe During Inspection

When inspecting the robot, be sure to

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

Staying Safe During Maintenance

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

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

Warning

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

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

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

Warning

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

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

KEEPING MACHINE TOOLS AND EXTERNAL DEVICES SAFE

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

Programming Safety Precautions

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

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

Mechanical Safety Precautions

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

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

KEEPING THE ROBOT SAFE

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

Operating Safety Precautions

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

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

Programming Safety Precautions

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

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

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

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 personal 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 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.

Warning

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.

Warning

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.

INTRODUCTION

Contents

The EtherNet/IP interface supports an I/O exchange with other EtherNet/IP enabled devices over an Ethernet network. The EtherNet/IP specification is managed by the Open DeviceNet Vendors Association (www.odva.org).

From the EtherNet/IP Specification (Release 1.0) Overview :

> “ EtherNet/IP (Ethernet/Industrial Protocol) is a communication system suitable for use in industrial environments. EtherNet/IP allows industrial devices to exchange time-critical application information. These devices include simple I/O devices such as sensors/actuators, as well as complex control devices such as robots, programmable logic controllers, welders, and process controllers.
>
> EtherNet/IP uses CIP (Control and Information Protocol), the common network, transport and application layers also shared by ControlNet and DeviceNet. EtherNet/IP then makes use of standard Ethernet and TCP/IP technology to transport CIP communications packets. The result is a common, open application layer on top of open and highly popular Ethernet and TCP/IP protocols.
>
> EtherNet/IP provides a producer/consumer model for the exchange of time-critical control data. The producer/consumer model allows the exchange of application information between a sending device (e.g., the producer) and many receiving devices (e.g., the consumers) without the need to send the data multiple times to multiple destinations. For EtherNet/IP, this is accomplished by making use of the CIP network and transport layers along with IP Multicast technology. Many EtherNet/IP devices can receive the same produced piece of application information from a single producing device.
>
> EtherNet/IP makes use of standard IEEE 802.3 technology; there are no non-standard additions that attempt to improve determinism. Rather, EtherNet/IP recommends the use of commercial switch technology, with 100 Mbps bandwidth and full-duplex operation, to provide for more deterministic performance. ”

The terms adapter and scanner are used throughout this manual. Although EtherNet/IP is a producer/consumer network, these terms are still appropriate to describe a device which creates the I/O connection (the scanner), and a device which responds to connection requests (the adapter). The scanner can also be called the connection originator. The adapter can also be called the connection target.

The following steps are necessary to configure EtherNet/IP with the robot as the adapter:

1. **Design and install the network.** It is critical to follow good network design and installation practices for a reliable network. Refer to Section 8.1 .
2. **Set the IP addresses.** All devices on the network require a valid IP address. Refer to Section 2.3 for additional information for the robot.
3. **Configure the adapter devices.** Adapter devices might require configuration such as setting I/O sizes. Refer to Section 3.2.1 to configure the robot as an adapter.
4. **Configure the scanner devices.** Scanners must be configured with a list of devices (adapters) to connect to along with parameters for each connection. Refer to Section 3.2.2 to configure an Allen Bradley ControlLogix PLC to connect to the robot.

5. **Map EtherNet/IP I/O to digital, group, or UOP I/O points within the robot.** Refer to Section 6.2 for more information. Scanner connections can also be mapped to analog. Refer to Section 4.2.5 .

6. **Backup the configuration.** Refer to Section 6.3 for details on doing this for the robot.

Note If you need to perform diagnostics or troubleshooting, refer to Chapter 9 *DIAGNOSTICS AND TROUBLESHOOTING* .

Chapter 2

SYSTEM OVERVIEW

Contents

2.1 OVERVIEW

The robot supports 32 connections. Each connection can be configured as either a Scanner connection, or as an Adapter connection. Adapter connections are normally to a cell controller or PLC to exchange cell interface I/O data. The EtherNet/IP Adapter option must be loaded to support this functionality.

Each Scanner connection can be configured to exchange I/O with a remote device capable of acting as an adapter on an EtherNet/IP network. The EtherNet/IP Scanner option must be loaded to support this functionality (the EtherNet/IP Scanner option includes the adapter functionality as well).

The EtherNet/IP interface corresponds to Rack 89 in the robot for I/O mapping. The slot number reflects the connection number from the EtherNet/IP interface user interface screen. Any amount of I/O can be mapped within EtherNet/IP, up to the maximum supported on the robot. Analog I/O is supported on scanner connections.

Good network design is critical to having reliable communications. Excessive traffic and collisions must be avoided or managed. Refer to Section 8.1 for details.

2.2 SPECIFICATION OVERVIEW

Table 2–1 provides an overview of specifications for EtherNet/IP.

Table 2–1. Specification Overview

Item	Specification
Number Adapter Connections	0–32
Number Scanner Connections	32 minus the number of adapter connections
Minimum RPI	8 msec
Number of Inputs	Up to the full amount of I/O supported on the robot.
Number of Outputs	Up to the full amount of I/O supported on the robot.
Supported Signal Types	Digital, Group, UOP, Analog (for scanner connections only)

Note In order for the scanner to work, the EtherNet/IP Scanner option must be loaded.

2.3 ETHERNET CONNECTION AND IP ADDRESS ASSIGNMENT

The robot must have a valid IP (Internet protocol) address and subnet mask to operate as an EtherNet/IP node. Details on the Ethernet interface and TCP/IP configuration can be found in the *Internet Options Setup and Operations Manual* .

The Ethernet interface supports 10Mbps and 100Mbps baud rates, along with half and full duplex communication. By default both interfaces will auto-negotiate and should be connected to a switch which supports 100Mbps full duplex connections. The LEDs located near the RJ45 connectors on the main CPU board are useful in confirming link establishment (for details on the LEDs, refer to appendix “Diagnostic Information” in the *Internet Options Setup and Operations Manual*).

The IP address(es) can be configured in the following ways :

- Manually configured on the robot teach pendant – Refer to the "Setting Up TCP/IP" chapter in the *Internet Options Setup and Operations Manual.*
- DHCP (Dynamic Host Configuration Protocol) – Refer to the “Dynamic Host Configuration Protocol” chapter in the *Internet Options Setup and Operations Manual* .

  Note DHCP is an optional software component. It is important to utilize static or infinite lease IP addresses when using EtherNet/IP.

Either one or both Ethernet ports can be configured for use with EtherNet/IP. Note that in order to use both ports at the same time they must be properly configured on separate subnets. Refer to the "Setting Up TCP/IP" chapter in the *Internet Options Setup and Operations Manual.* Also note that port 2 (CD38B) is optimized for Ethernet I/O protocols such as EtherNet/IP. The preferred setup is to connect port 1 (CD38A) to your building network to access the robot through HTTP, FTP, and so forth , and to connect port 2 (CD38B) to an isolated network for use by EtherNet/IP.

Note Be sure that all EtherNet/ IP node IP addresses are configured properly before you perform the functions in this manual. The PING utility can be used to verify basic communications. Refer to Chapter 9 *DIAGNOSTICS AND TROUBLESHOOTING* for more information.

2.4 ADAPTER MODE CONFIGURATION OUTLINE

Perform the following steps to configure the adapter connection on the robot:

- Configure the I/O size on the robot. Refer to Section 3.2.1 .
- Map the physical EtherNet/IP I/O to logical I/O points (digital, group, or UOP) on the robot. Refer to Section 6.2 .
- Configure the remote scanner (for example, ControlLogix PLC). Refer to Section 3.2.2 .

Table 2–2 provides a summary of the adapter configuration. This information is used in the scanner device (for example, PLC) configured to communicate with the robot EtherNet/IP Adapter interface.

Table 2–2. Adapter Configuration Summary

ITEM	DESCRIPTION
Vendor ID	356
Product Code	2
Device Type	12
Communication Format	Data – INT
Input Assembly Instance	101–132
Input Size	User Configurable, Set in 16-bit Words
Output Assembly Instance	151–182
Output Size	User Configurable, Set in 16-bit Words
Configuration Instance	100
Configuration Size	0

The default I/O size for the adapter connection is four words for both inputs and outputs. This corresponds to 64 I/O points based on a 16-bit word. This size must be configured on the robot teach pendant, as well as on the remote scanner (for example, PLC).

Refer to Chapter 3 *ADAPTER CONFIGURATION* for details.

2.5 SCANNER MODE CONFIGURATION OUTLINE

The robot must be configured to initiate EtherNet/IP connections. Up to 32 scanner connections are supported. Perform the following steps to configure the scanner connection on the robot :

- Configure the robot scan list on the teach pendant. Refer to Section 4.2.3 .
- Map the physical EtherNet/IP I/O to logical I/O points (for example, digital, group, analog, or UOP) on the robot. Refer to Section 6.2 .

For each connection the following data must be provided on the robot teach pendant. (Refer to the manual that applies to the adapter device being configured for more information.)

- Name/IP address
- Vendor ID
- DeviceType

- Product Code
- Input Size (16-bit words or 8-bit byes)
- Output Size (16-bit words or 8-bit byes)
- RPI (ms)
- Input assembly instance
- Output assembly instance
- Configuration instance

Note The robot currently cannot be configured for devices with a non-zero configuration size from the teach pendant (monochrome or *i* Pendant). Refer to Appendix A for information on third party configuration tools.

Chapter 3

ADAPTER CONFIGURATION

Contents

3.1 OVERVIEW

The robot supports up to 32 adapter connections. These connections are normally to a cell controller or PLC to exchange cell interface I/O data. The EtherNet/IP Adapter Option must be loaded to support this functionality.

The following steps are required to configure the adapter connection on the robot :

- Configure I/O size on the robot. Refer to Section 3.2.1 .
- Map the physical EtherNet/IP I/O to logical I/O points (digital, group, or UOP) on the robot. Refer to Section 6.2 .
- Configure the remote scanner (for example, ControlLogix PLC). Refer to Section 3.2.2 .

3.2 SETTING UP YOUR ROBOT

3.2.1 Configuring the Robot I/O Size

The Input size and Output size are set in 16-bit word sizes. This means if 32 bits of input and 32 bits of output are needed then the Input size and Output Size would be set to 2 words each. The default size of the adapter connection is 4 words (64 bits) of input and 4 words (64 bits) of output. Changes in I/O size require you to turn off and turn on the robot to take effect.

Refer to Procedure 3-1 to configure I/O size on the robot.

Table 3–1 describes the items displayed on the EtherNet/IP Status screen.

Table 3–2 describes the items on the EtherNet/IP Configuration screen.

Table 3–1. EtherNet/IP Status Screen Descriptions

ITEM	DESCRIPTION
Description Default: ConnectionX where X is the slot number of the Adapter.	This item is the description of the adapter or scanner. This can be set as desired to coordinate with your equipment.
TYP Default: ADP	This item indicates whether the connection is configured as an Adapter, or as a Scanner.
Enable Default: TRUE (for Adapter 1, FALSE for Adapters 2–32)	This item indicates whether the adapter or scanner is enabled (TRUE) or disabled (FALSE).

Table 3–1. EtherNet/IP Status Screen Descriptions (Cont'd)

ITEM	DESCRIPTION
Status	The **Status** field can have the following values : • OFFLINE– the connection is disabled. • ONLINE – the connection is enabled but is not active (for example, waiting for a connection). • RUNNING - the connection is enabled and active (I/O is being exchanged). • <RUNNING> - the connection is enabled and active (I/O is being exchanged), and auto-reconnect is enabled. See Table 4–4 for more information on the auto-reconnect setting. • PENDING – changes have taken place in configuration. You must turn off the robot, then turn it on again.
Slot	This item is the value used when mapping EtherNet/IP I/O to digital, group, or UOP I/O signals.

Table 3–2. EtherNet/IP Configuration Screen Descriptions

ITEM	DESCRIPTION
Description	This item is the comment that shows up on the Status screen. It is set on the Status screen as well.
Input size (words) Default:	This item is the number of 16 bit words configured for input.
Output size (words) Default:	This item is the number of 16 bit words configured for output.
Alarm Severity Default: WARN	This item indicates the severity of alarm that will be posted by the adapter connection. The valid choices are STOP, WARN, and PAUSE.
Scanner IP	The IP address of the connected scanner.
API 0 => T	Actual Packet Interval at which the scanner/originator is producing.
API T => 0	Actual Packet Interval at which the adapter/target is producing.

Procedure 3-1 Configuring I/O Size on the Robot

Steps

1. Press MENUS.
2. Select I/O.

3. Press F1, [TYPE], and select EtherNet/IP. You will see a screen similar to the following.

```
I/O EtherNet/IP               JOINT  10 %
EtherNet/IP List(Rack 89)            1/8
 Description  TYP  Enable  Status  Slot
 Connection1  ADP  TRUE     ONLINE    1
 Connection2  ADP  FALSE   OFFLINE    2
 Connection3  ADP  FALSE   OFFLINE    3
 Connection4  ADP  FALSE   OFFLINE    4
 Connection5  ADP  FALSE   OFFLINE    5
 Connection6  ADP  FALSE   OFFLINE    6
 Connection7  ADP  FALSE   OFFLINE    7
 Connection8  ADP  FALSE   OFFLINE    8
```

Refer to Table 3–1 for descriptions of these screen items.

4. Move the cursor to select a connection. If the connection is configured as a scanner, move the cursor to the TYP column and press F5. This configures the connection as an adapter.

5. Move the cursor to select the desired adapter. If you plan to make changes to the adapter configuration, you must first disable the connections. Otherwise, the configuration screen is read-only.

   Note If the adapter connection is Enabled, the first line of the adapter configuration screen will display “Adapter config (Read-only)” and the items on the screen cannot be modified. To make changes to the adapter configuration screen, you must disable the adapter connection on the EtherNet/IP Status screen.

6. **To change adapter status:**

   a. Move the cursor to highlight the field in the Enable column for the adapter.

   b. To disable the adapter and change the status to OFFLINE, press F5, FALSE.

      To enable the adapter and change the status to ONLINE, press F4, TRUE.

7. Move the cursor to the Description column. Press F4, CONFIG. You will see a screen similar to the following.

```
Adapter configuration :

     Description :      Connection1
     Input size (words) :  4
     Output size(words) :  4
     Alarm Severity : WARN

     Scanner IP : ****************
     API O=>T :       0
     API T=>O :       0
```

Refer to Table 3–2 for descriptions of these screen items.

8. To change the I/O size:

a. Move the cursor to select "Input size (words)."

b. Type the value you want and press Enter.

c. Move the cursor to select "Output size (words)."

d. Type the value you want and press Enter.

e. Move the cursor to select the alarm severity.

f. Pres F4, [CHOICE], and select the desired severity.

g. To return to the previous screen, press F3, [PREV].

9. After modifying the adapter configuration, you must enable the connection on the EtherNet/IP status screen. If any changes were made, the status will show as “PENDING”. This indicates that you must cycle power in order for the changes to take effect.

Note To map EtherNet/IP I/O to digital, group, or UP I/O, refer to Section 6.2 .

3.2.2 Configuring the Remote Scanner

The EtherNet/IP Interface status screen should show that the adapter connection is ONLINE. This means it is available and waiting for a request from a scanner (for example, PLC) to exchange I/O. If the adapter status is not ONLINE, refer to Procedure 3-1 , Step 6 .

Table 3–3 provides a summary of the adapter configuration. This information is used to configure the remote scanner (for example, PLC).

Table 3–3. Adapter Configuration Summary

ITEM	DESCRIPTION
Vendor ID	356
Product Code	2
Device Type	12
Communication Format	Data – INT
Input Assembly Instance	101–132
Input Size	User Configurable, Set in 16-bit Words
Output Assembly Instance	151–182
Output Size	User Configurable, Set in 16-bit Words
Configuration Instance	100
Configuration Size	0

Table 3–4. Connection Points

Slot Number	Input Assembly Instance	Output Assembly Instance
1	101	151
2	102	152
3	103	153
4	104	154
5	105	155
6	106	156
7	107	157
8	108	158
9	109	159
10	110	160
11	111	161
12	112	162
13	113	163
14	114	164
15	115	165
16	116	166
17	117	167

Table 3–4. Connection Points (Cont'd)

Slot Number	Input Assembly Instance	Output Assembly Instance
18	118	168
19	119	169
20	120	170
21	121	171
22	122	172
23	123	173
24	124	174
25	125	175
26	126	176
27	127	177
28	128	178
29	129	179
30	130	180
31	131	181
32	132	182

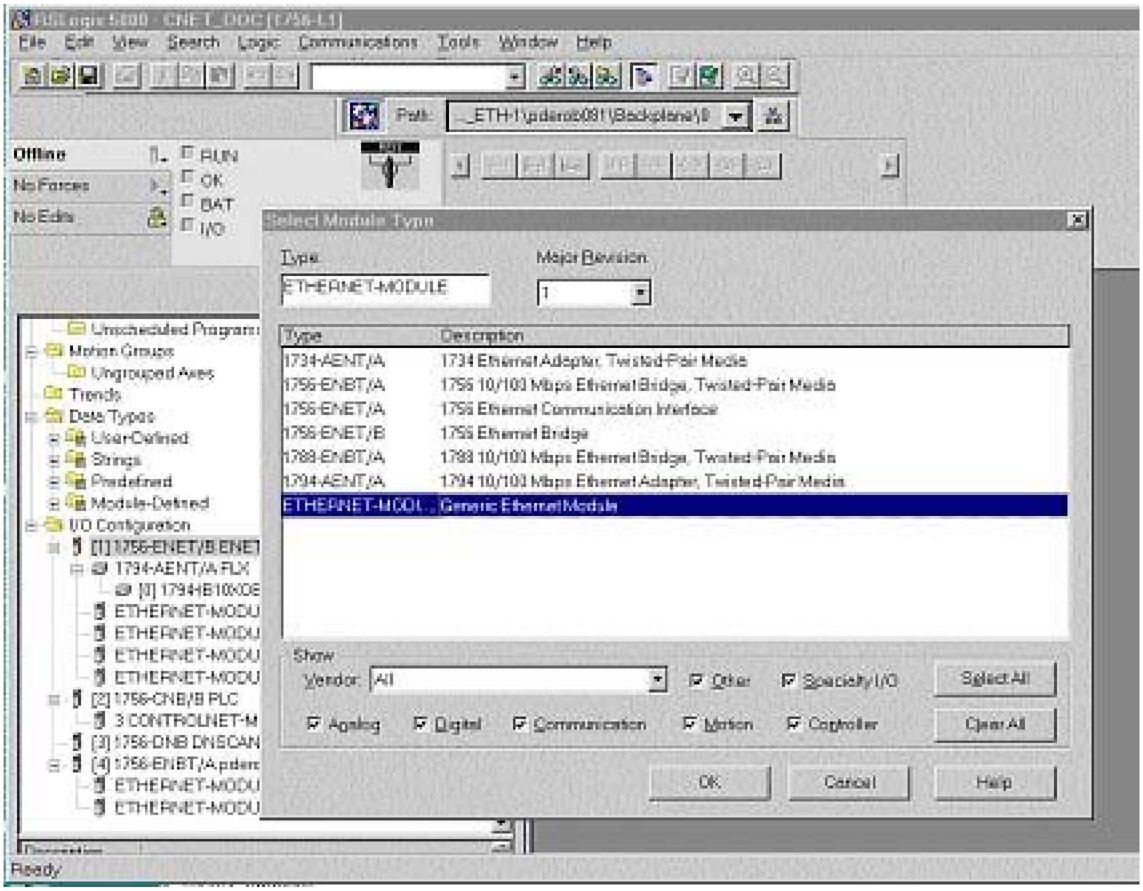
Use Procedure 3-2 to configure the Allen Bradley ControlLogix PLC. for other scanners, refer to their configuration software in conjunction with Table 3–3 .

Procedure 3-2 Configuring the Scanner Using RS-Logix5000 Software

Steps

Note The following screens show how to configure the scanner using RS-Logix5000 software, which is used with the Allen Bradley ControlLogix PLC. This example assumes that an EtherNet/IP Bridge module has been added to the configuration in the ControlLogix PLC.

1. To add the robot adapter connection to the configuration, right-click the EtherNet/IP Bridge module in the PLC, and select“ New Module”.
2. Select “Generic Ethernet Module,” and click OK. You will see a screen similar to the following.



3. In the next screen, RSLogix will ask for information regarding the communication to the robot. Type a name for the robot adapter connection.

In the example below we call the module “Example_Robot”. This name will create a tag in RSLogix, which can be used to access the memory location in the PLCs memory where the data for the Example_Robot will be stored. A description can also be added if desired (optional).

4. Select the “Comm_Format,” which tells RSLogix the format of the data.

In this example we select Data-INT, which will represent the data in the robot as a field of 16-bit words.

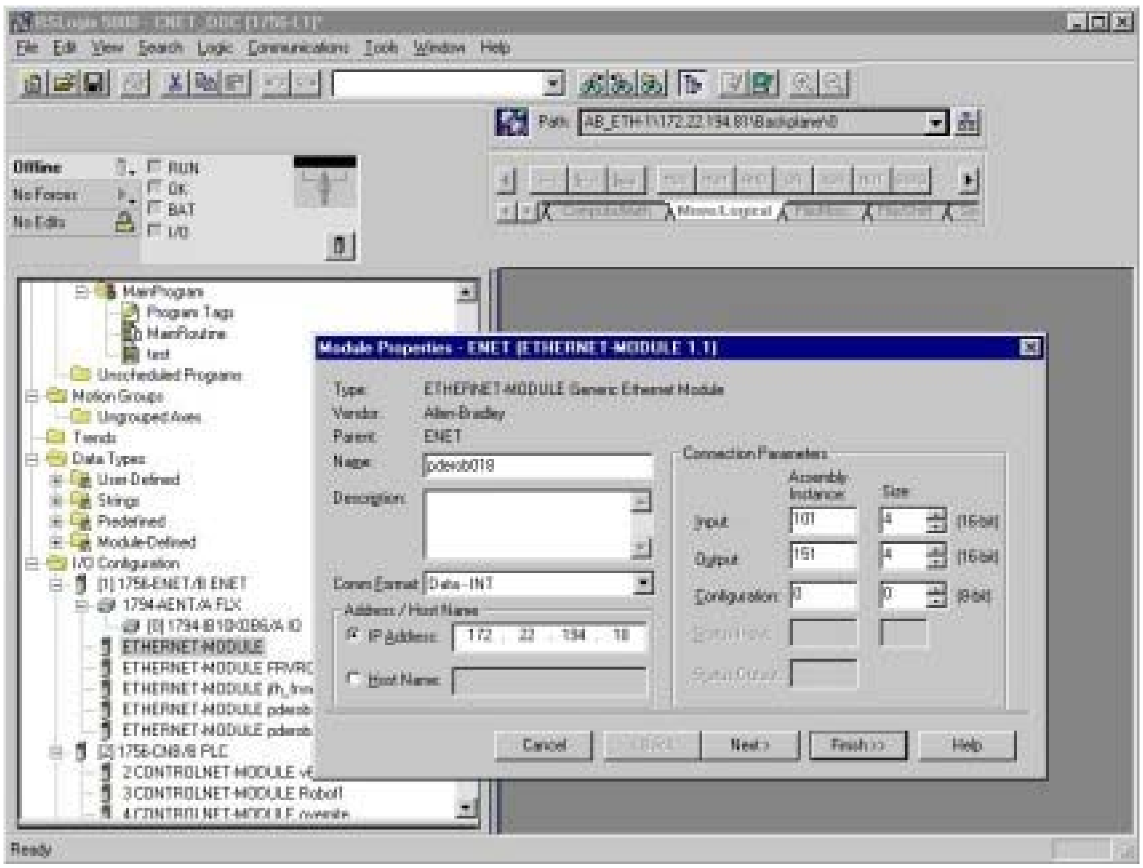
5. Set connection parameters as follows:

Each of the 32 Connections have different Connection parameters, which are based on the slot number. Refer to Table 3–4 to determine the correct parameters. The size of the input connection and the output connection must correspond to the size that we configured for the robot. In this example we configured the robot for 4 (16 bit) words of input data and output data. The configuration instance should be set to 100 and size of the configuration instance is set to 0.

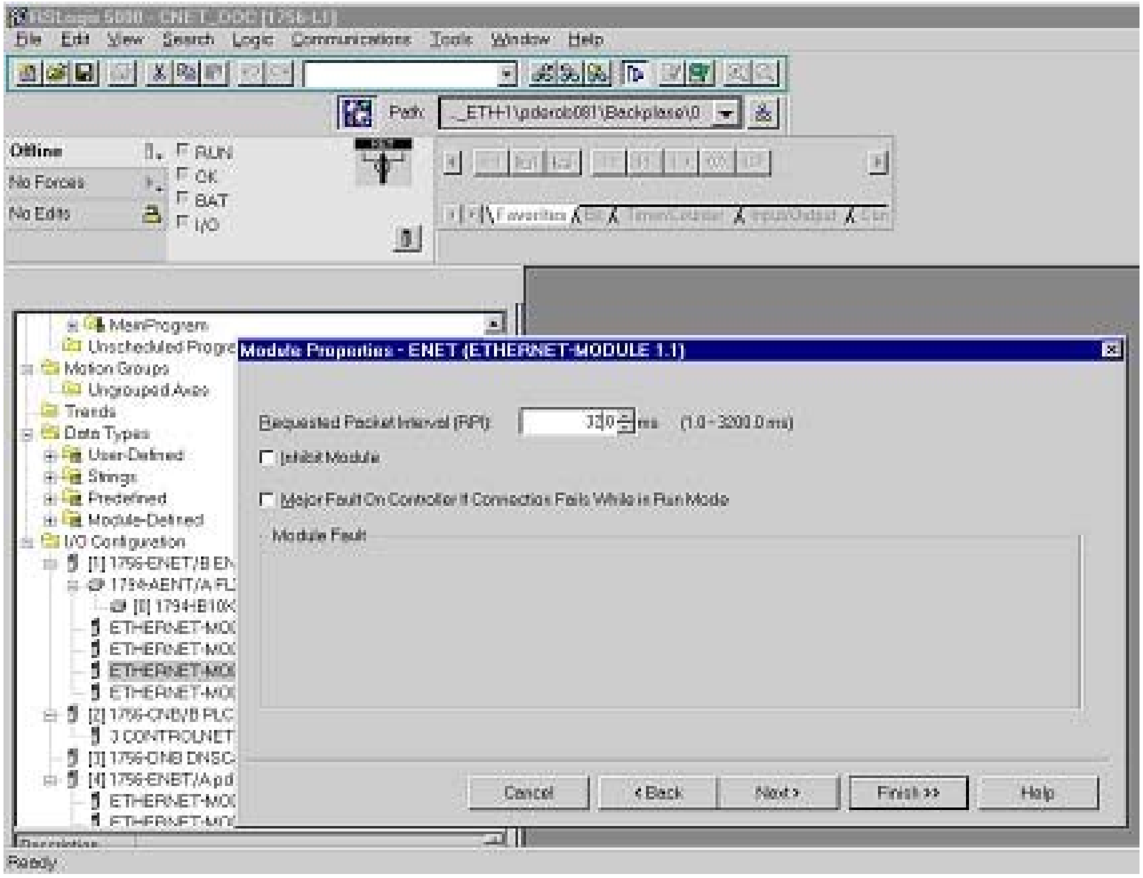
In the following example, we will be setting up connection parameters corresponding to the adapter in slot 1.

6. Type the IP address that we have configured for the module.

This could be the Host Name if the DNS (Domain Name Service) is configured and available for the PLC to resolve names to IP addresses (if names are used be sure the DNS server is very reliable and always available to the PLC during operation). You will see a screen similar to the following.

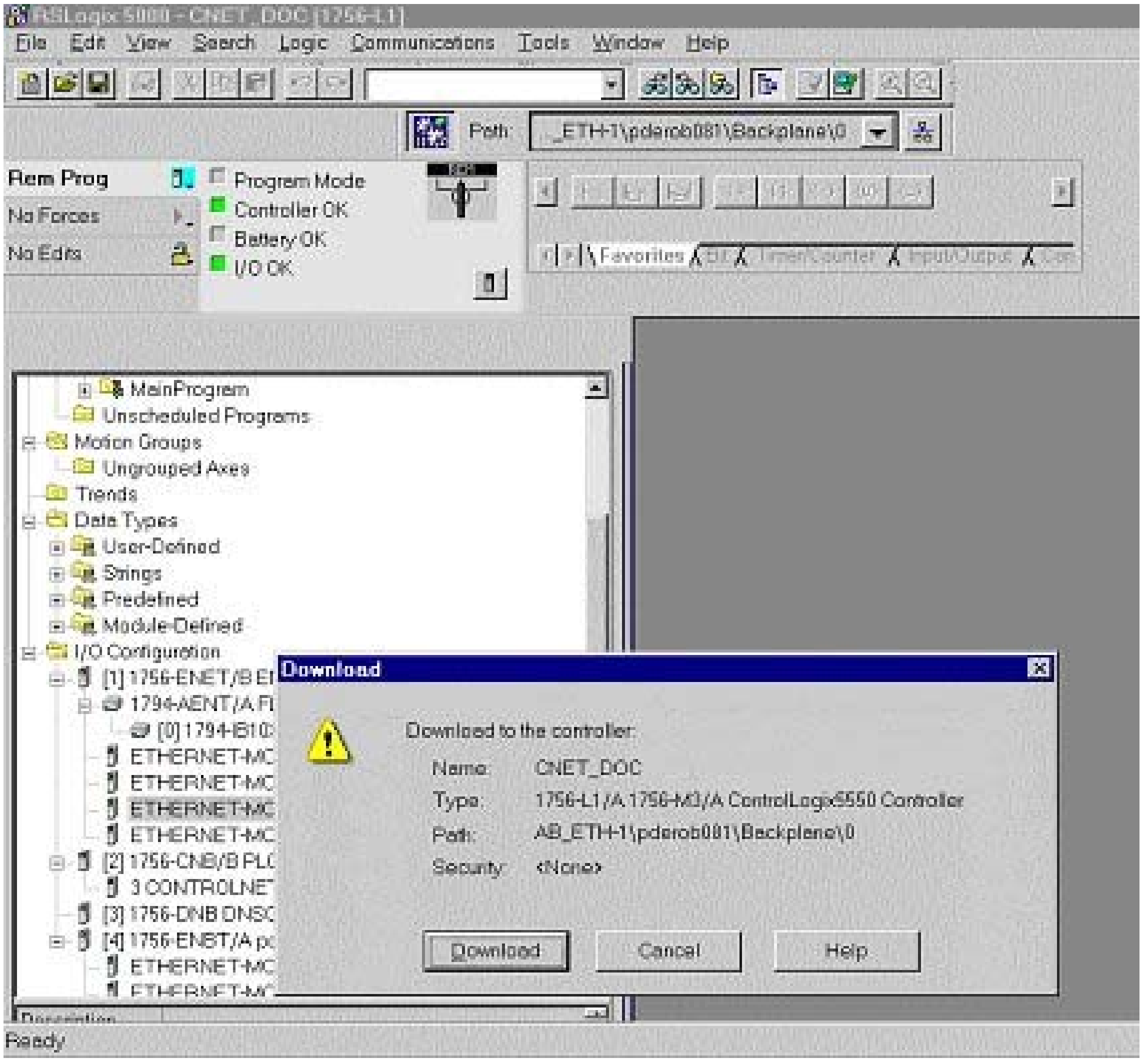


7. The RPI (requested packet interval) is set on the next screen. This sets the rate for I/O updates. In the following example the RPI is set to 32ms. This means the PLC will send its outputs to the robot every 32ms, and the robot will send its inputs to the PLC every 32ms. You will see a screen similar to the following.



8. Press Finish to complete this step.

9. To download the new configuration to the ControlLogix PLC, select Download from the Communications menu. You will see a screen similar to the following.



10. If there are any errors, a warning triangle will be present on the Example_Robot in the I/O configuration listing. Double click the module to view any error that is reported.

3.2.3 Common Errors

The robot will post an alarm indicating that the adapter connection is idle if it is enabled but no scanner has connected to it. This is an informational alarm (warning level) by default. This message is reposted whenever the robot RESET button is pressed and the adapter connection is idle. If desired, the error severity level of EtherNet/IP adapter alarms can be increased. See Step Step 8 of Procedure 3-1 for more details.

If the adapter connection is lost the values of any mapped inputs will be zeroed out (by default). The last state behavior can be changed by setting the following system variable : $EIP_CFG.$KEEP_IO_ADP. The values are :

- FALSE : The last state values of the adapter inputs will be zero (default)
- TRUE : The last state values of the adapter inputs will be their last value

Note If some of the EtherNet/IP adapter I/O signals are configured as UOP HOLD/IMSTP signals and communication is interrupted, then the default behavior will cause the robot to stop. This is due to the UOP inputs going to zero (last state behavior) and causing UOP HOLD and IMSTP alarms to be posted. It is typical for the adapter connection to be configured so that alarms are of “WARNING” severity and the “Last State” is set to FALSE (default behavior), which allows UOP in/out signals to stop the robot operation.

Chapter 4

SCANNER CONFIGURATION

Contents

4.1 OVERVIEW

The robot supports up to 32 scanner connections. Each connection can be configured to exchange I/O with a remote device capable of acting as an adapter on an EtherNet/IP network. The EtherNet/IP Scanner option must be loaded to support this functionality. The EtherNet/IP Scanner option includes the adapter functionality as well. An example of an EtherNet/IP adapter device that the robot would connect to might be an I/O block.

The robot must be configured to initiate EtherNet/IP connections. Up to 32 scanner connections are supported. Perform the following steps to configure the scanner connection on the robot :

- Configure the adapter device if required. Refer to Section 4.2.2 .
- Configure the robot scan list on the teach pendant. Refer to Section 4.2.3 or configure the robot scan list from RSNetWorx for EtherNet/IP (Refer to Appendix A).
- Map the physical EtherNet/IP I/O to logical I/O points (digital, group, analog, or UOP) on the robot. Refer to Section 6.2 .

Note All scanlist configurations must either be done entirely from the *i*Pendant, or be done entirely from a third-party configuration tool such as RSNetWorx for EtherNet/IP.

4.2 SETTING UP YOUR ROBOT

4.2.1 Overview

For each connection, the following data must be provided on the robot teach pendant (see documentation for the adapter device being configured for more information) :

- Name/IP address
- Vendor ID
- Device Type
- Product Code
- Input Size (16-bit words or 8-bit byes)
- Output Size (16-bit words or 8-bit byes)
- RPI (ms)
- Input assembly instance
- Output assembly instance
- Configuration instance

Note The robot currently cannot be configured for devices with a non-zero configuration size from the teach pendant (monochrome or *i* Pendant). Refer to Appendix A for information on third party configuration tools.

4.2.2 Configure the Adapter Device

See the documentation for the adapter device being configured. Configuring the adapter is typically a matter of connecting the device to the network and setting the IP address. The device should successfully respond to a PING request before proceeding. Refer to Chapter 9 *DIAGNOSTICS AND TROUBLESHOOTING* for details on using PING.

Note The number of scanner connections equals 32 minus the number of adapter connections.

4.2.3 Configure the robot scan list

Use Procedure 4-1 to configure the robot scan list from the teach pendant.

Table 4–1 describes the items displayed on the EtherNet/IP Status screen. Table 4–2 describes the items displayed on the EtherNet/IP scanner configuration screen. Table 4–3 lists RPI minimum values.

Table 4–1. EtherNet/IP Status Screen Item Descriptions

ITEM	DESCRIPTION
Description Default: ConnectionX where X is the slot number of the Adapter.	This item is the description of the adapter or scanner. This can be set as desired to coordinate with your equipment.
TYP Default: ADP	This item indicates whether the connection is configured as an Adapter, or as a Scanner.
Enable Default: TRUE (for Adapter 1, FALSE for Adapters 2–32)	This item indicates whether the adapter or scanner is enabled (TRUE) or disabled (FALSE).

Table 4–1. EtherNet/IP Status Screen Item Descriptions (Cont'd)

ITEM	DESCRIPTION
Status	The **Status** field can have the following values : • OFFLINE– the connection is disabled. • ONLINE – the connection is enabled but is not active (for example, waiting for a connection). • RUNNING - the connection is enabled and active (I/O is being exchanged). • <RUNNING> - the connection is enabled and active (I/O is being exchanged), and auto-reconnect is enabled. See Table 4–4 for more information on the auto-reconnect setting. • PENDING – changes have taken place in configuration. You must turn off the robot, then turn it on again.
Slot	This item is the value used when mapping EtherNet/IP I/O to digital, group, or UOP I/O signals.

Table 4–2. Scanner Configuration Screen Item Descriptions

ITEM	DESCRIPTION
Description	This item is the comment that shows up on the Status screen.
Name/IP address	This item is the hostname or IP address of the device to which you are connecting. If a hostname is used, it must be in the local host table or available through DNS.
Vendor ID	This item is the vendor ID of the device to which you are connecting. Refer to the adapter (target) device's documentation of EDS files for assigned value. The vendor ID, Device Type, and Product Code can be entered if electronic keying is needed (this information must match the device in order to make a successful connection). If the fields are left at 0 then the keying is ignored.
Device Type	This item is the Device Type of the device to which you are connecting. Refer to the adapter (target) device's documentation or EDS file for assigned value. The vendor ID, Device Type, and Product Code can be entered if electronic keying is needed (this information must match the device in order to make a successful connection). If the fields are left at 0 then the keying is ignored.
Product code	This item is the product code of the device to which you are connecting. Refer to the adapter (target) device's documentation or EDS file for assigned value. The vendor ID, Device Type, and Product Code can be entered if electronic keying is needed (this information must match the device in order to make a successful connection). If the fields are left at 0 then the keying is ignored.

Table 4–2. Scanner Configuration Screen Item Descriptions (Cont'd)

ITEM	DESCRIPTION
Input size Range: 0 – 64 Default: 0	This item is the number of words or bytes configured for input. The default data type is 16-bit words, but can be configured as 8-bit bytes. To change the data type, modify the I/O Data Type field in the EtherNet/IP Advanced Scanner Configuration Screen (See Table 4–4). The Input size and Output size need to match the adapter device to which the robot will connect.
Output size Range: 0 – 64 Default: 0	This item is the number of words or bytes configured for output.The default data type is 16-bit words, but can be configured as 8-bit bytes. To change the data type, modify the I/O Data Type field in the EtherNet/IP Advanced Scanner Configuration Screen (See Table 4–4). The Input size and Output size need to match the adapter device to which the robot will connect.
RPI (ms) Min: 8 ms Max:5000 Default: 32	This item is the requested packet interval. This defines how often I/O updates are done. The minimum value allowed is 8 ms, however this value should be set based on application requirements. Be aware that fast I/O updates cause excessive network traffic. Refer to Table 4–3 for a guide to minimum RPI values within the robot. As a rule of thumb, the robot controller can support a maximum of 1250 packet per second. Both Originator-to-Target and Target-to-Originator packets must be factored into this calculation.
Assembly instance (input)	The Input, Output, and Configuration instance values need to be set based on the adapter device to which the robot will connect.
Assembly instance (output)	The Input, Output, and Configuration instance values need to be set based on the adapter device to which the robot will connect.
Configuration instance	The Input, Output, and Configuration instance values need to be set based on the adapter device to which the robot will connect.

Table 4–3. Requested Packet Interval (RPI) Minimum Values

Number of Connections	Minimum RPI for any connection (ms)
1	8
2	8
3	8
4	8
5	8
6	12
7	12
8	16
9	16
10	16

Table 4–3. Requested Packet Interval (RPI) Minimum Values (Cont'd)

Number of Connections	Minimum RPI for any connection (ms)
11	20
12	20
13	24
14	24
15	24
16	28
17	28
18	32
19	32
20	32
21	36
22	36
23	36
24	40
25	40
26	44
27	44
28	44
29	48
30	48
31	48
32	52

Procedure 4-1 Configuring the Robot Scan List

Steps

1. Press MENUS.
2. Select I/O.
3. Press F1, [TYPE], and select EtherNet/IP. You will see a screen similar to the following.

```
I/O EtherNet/IP              JOINT  10 %
EtherNet/IP List(Rack 89)           1/8
  Description  TYP  Enable  Status  Slot
  Connection1  ADP  TRUE     ONLINE    1
  Connection2  ADP  FALSE   OFFLINE    2
  Connection3  ADP  FALSE   OFFLINE    3
  Connection4  SCN  FALSE   OFFLINE    4
  Connection5  SCN  FALSE   OFFLINE    5
  Connection6  SCN  FALSE   OFFLINE    6
  Connection7  SCN  FALSE   OFFLINE    7
  Connection8  SCN  FALSE   OFFLINE    8
```

4. Move the cursor to the connection you want to set. If the connection is configured as an adaptor, move the cursor to the TYP column, and press F4. This configures the connection as a scanner.

   Note If the scanner connection is enabled, the first line of the scanner configuration screen will display “Scanner config (Read-only)” and the items on the screen cannot be modified. To make changes to the read-only scanner configuration screen, you must disable the scanner connection on the EtherNet/IP status screen.

5. **To change scanner status:**

   a. Move the cursor to highlight the field in the Enable column for the scanner you want to modify.

   b. To disable the scanner and change the status to OFFLINE, press F5, [FALSE].

      To enable the scanner and change the status to RUNNING, press F4, [TRUE].

      Note The status will not change until the connection has been established and I/O is being exchanged.

6. Press F4, CONFIG. You will see a screen similar to the following.

```
I/O EtherNet/IP              JOINT  10 %
Scanner configuration :            1/10
    Description :     Connection1
    Name/IP address : 192.168.0.12
    Vendor Id :    0
    Device Type :    0
    Product code :    0
    Input size (words):  1
    Output size (words):  1
    RPI (ms) :   32
    Assembly instance(input) :  1
    Assembly instance(output) :  2
    Configuration instance : 4
```

7. Move the cursor to select each item and set the appropriate value.

Note If you make changes to I/O size, you must turn off then turn on the controller in order for the changes to take effect. Other changes in the configuration do not require you to turn off then on the controller to take effect.

8. Press the PREV key to return to the EtherNet/IP Status screen. You can enable the connection. If the status is PENDING then you must turn off then turn on the controller in order for the changes to take effect.

Note Any enabled scanner connections which are not RUNNING or PENDING will be retried each time the robot is RESET.

Note To map EtherNet/IP I/P to digital, group, analog, or UOP I/O, refer to Section 6.2 .

4.2.4 Advanced EtherNet/IP Scanner Configuration

An advanced EtherNet/IP configuration screen is provided to allow you to access advanced scanner configuration options. Table 4–4 describes the items displayed on the Advanced Scanner Configuration Screen. The advance screen can be accessed and configured by using Procedure 4-2 .

Table 4–4. EtherNet/IP Advanced Scanner Configuration Screen Item Descriptions

ITEM	DESCRIPTION
I/O Data Type Default: 16-bit words	This item indicates allows changing the data type to 16-bit words or 8-bit bytes.
Timeout Multiplier Default: DEFAULT	This item indicates allows changing the timeout multiplier. When set to DEFAULT, the controller will intelligently choose an appropriate multiplier based on the RPI value.
Reconnect Default: FALSE	If this item is set to TRUE, the scanner will attempt to re-establish the connection when the connection is enabled and in an OFFLINE state. **Note** The reconnect parameter was designed for tool changing applications. Enabling reconnect has the following side effect. While enabled, all EtherNet/IP alarms relating to connection establishment and connection time-outs for the corresponding connection will be masked (will not be posted). See Appendix B in the manual for more details. As such, it is recommended that non-tool changing applications do not enable this parameter in a production environment.

Table 4–4. EtherNet/IP Advanced Scanner Configuration Screen Item Descriptions (Cont'd)

ITEM	DESCRIPTION
Major Revision Default: 0	This item indicates the major revision number of the device being scanned. Is sometimes required by third-party configuration devices.
Minor Revision Default: 0	The minor revision number of the device being scanned. Is sometimes required by third-party configuration devices.
Alarm Severity	This item indicates the severity of alarm that will be posted by the scanner connection. The valid choices are STOP, WARN, and PAUSE.
Originator To Target RPI(ms) Default: 32	This item indicates the Requested Packet Interval for the scanner to produce at in milliseconds. This field allows for the scanner to have different RPIs for producing and consuming data.
Transport Type Default: UNICAST	This item allows the scanner to request that the adapter send data using a point-to-point/unicast connection, or to multicast data. If multicasting is not required, we strongly recommend setting this value to UNICAST. However, a small number of adapter devices only support the MULTICAST setting.
Target To Originator RPI(ms) Default: 32	This item indicates the Requested Packet Interval for the scanner to consume at in milliseconds. This field allows for the scanner to have different RPIs for producing and consuming data.
Connection Type Default: (blank)	This item allows the user to set up a scanner connection of type Exclusive-Owner, Input-Only, or Listen-Only. When a connection type is selected, the O=>T Format and T=>O Format fields will automatically be modified to correspond with the selected Connection Type. This field will be blank after each power-cycle, as this field is only an aid in selecting the proper O=>T and T=>O formats. Exclusive-Owner is the must common connection type.
O=>T Format Default: Run/Idle Header	The format of the producer's data packet. By default this is set to Run/Idle Header, consistent with an Exclusive-Owner Connection Type.

Table 4–4. EtherNet/IP Advanced Scanner Configuration Screen Item Descriptions (Cont'd)

ITEM	DESCRIPTION
T=>O Format Default: Modeless	The format of the consumer's data packet. By default this is set to Modeless, consistent with an Exclusive-Owner Connection Type.
Configuration String Status Size(bytes)	Some EtherNet/IP adapters accept or require a non-zero length configuration string. This configuration data can only be configured on the robot using a third party configuration tool such as RSNetWorx for EtherNet/IP (Refer to Appendix A in the manual). This status item displays how much configuration data is currently configured for the connection. If no third party configuration tool is used, this item will always be 0.

Procedure 4-2 Configuring Advanced Scanner Options

1. Press MENUS.
2. Select I/O.
3. Press F1, [TYPE], and select EtherNet/IP.
4. Move the cursor to a Scanner connection.
5. Press F4, [CONFIG].
6. Press F2, [ADV]. You will see a screen similar to the following:

```
I/O EtherNet/IP              JOINT 100 %
Advanced configuration :           1/12
General
   I/O Data Type :      16-BIT WORDS
   Timeout Multiplier :4
   Reconnect :          FALSE
   Major Revision :     0
   Minor Revision :     0
   Alarm Severity :     STOP
Originator To Target
   RPI :                  32
Target To Originator
   Transport Type :     UNICAST
   RPI :                  32
Connection Type
   Type :               Exclusive-Owner
     O=>T Format :      Run/Idle Header
     T=>O Format :      Modeless

Configuration String Status
   Size(bytes) :          0
```

7. Move the cursor to select each item and set the appropriate value.
8. Press the PREV key to return to the EtherNet/IP Scanner configuration screen.
9. Press the PREV key to return to the EtherNet/IP Status screen. You can enable the connection. If the status is PENDING then you must turn off then turn on the controller in order for the changes to take effect.

4.2.5 Analog I/O

4.2.5.1 Overview

I/O for EtherNet/IP Scanner connections can be mapped to analog. Analog and digital I/O can be intermixed—for example a device may produce sixteen points of Digital Inputs and two words of Analog Inputs on the same connection to the robot controller.

Table 4–5 describes the items displayed on the Scanner Analog Configuration Screen. The analog screen can be accessed and configured by using Procedure 4-3 .

Table 4–5. Scanner Analog Configuration Screen Setup Items

ITEM	DESCRIPTION
RANGE Default: 1 – maximum allocated I/O	The Range of I/O points to be mapped to an analog channel, or a digital start point.
TYPE Default: Digital	The type of I/O being mapped: Analog or Digital.
START PT/CHNL Default: 1	The analog channel or the digital start point.

Procedure 4-3 Configuring Scanner Analog I/O

1. Press MENUS.
2. Select I/O.
3. Press F1, [TYPE], and select EtherNet/IP.
4. Move the cursor to a Scanner connection.
5. Press F4, [CONFIG].
6. Type the correct input and output sizes.
7. Press F4, [ANALOG]. You will see a screen similar to the following.

```
I/O EtherNet/IP              JOINT 100 %
Map Inputs:                         1/1
  #     RANGE       TYPE    START PT/CHNL
  1 [    1- 128]    Digital       1
```

8. Move the cursor to the RANGE column and select the range of the first collection of Inputs. If you do not want to intermix Analog and Digital Inputs, do not modify this column.
9. Select the type of Inputs, Analog or Digital, in the TYPE column.
10. Select the channel for Analog Input, or the point for Digital Input in the START PT/CHNL column.
11. Repeat as necessary as additional rows are automatically created.
12. Press F2 [IN/OUT]. You will see a screen similar to the following.

```
I/O EtherNet/IP              JOINT 100 %
Map Outputs:                        1/1
  #     RANGE       TYPE    START PT/CHNL
  1 [    1- 128]    Digital       1
```

13. Repeat Step 8 through Step 11 as necessary for Outputs.
14. Press the PREV key to return to the EtherNet/IP Scanner configuration screen.
15. Press the PREV key to return to the EtherNet/IP Status screen. You can enable the connection. If the status is PENDING then you must turn off then turn on the controller in order for the changes to take effect.

Note The 16-bits of each analog I/O channel can be byte-swapped (toggled from big-endian to little-endian) on a per connection basis by toggling the system variable $EIP_SC[].$ANALOGFMT. When the system variable is set to 0, the data will be produced and consumed in big-endian format. When set to 1, little-endian format will be used.

4.2.5.2 Examples

Suppose a device produces sixteen consecutive points of digital input followed by two words of analog input. A properly configured EtherNet/IP Analog In screen would look like the following:

```
I/O EtherNet/IP              JOINT 100 %
Map Inputs:                       1/2
  #     RANGE     TYPE     START PT/CHNL
  1 [    1-   16]  Digital        1
  2 [   17-   48]  Analog         1
```

Suppose a device produces sixteen consecutive points of digital input followed by two words of analog input followed by eight more consecutive points of digital input. A properly configured EtherNet/IP Analog In screen would look like the following. Note that the 49th connection input point will be mapped as the 17th digital input point.

```
I/O EtherNet/IP              JOINT 100 %
Map Inputs:                       1/3
  #     RANGE      TYPE    START PT/CHNL
  1 [    1-   16]   Digital       1
  2 [   17-   48]   Analog        1
  3 [   49-   56]   Digital       17
```

4.2.6 Common Errors

If the connection is lost, the values of any mapped inputs will be zeroed out. The last state behavior can be changed by setting the following system variable : $EIP_CFG.$KEEP_IO_SCN. The values are :

- FALSE : The last state values of the adapter inputs will be zero (default)

- TRUE : The last state values of the adapter inputs will be their last value

This setting applies to all the scanner connections.

Any enabled scanner connections which are not RUNNING or PENDING will be retried each time the robot is RESET.

Chapter 5

ETHERNET/IP TO DEVICENET ROUTING

Contents

5.1 OVERVIEW

EtherNet/IP and DeviceNet are both based on the Common and Industrial Protocol (CIP) which was initially defined by Rockwell with specifications managed by ODVA (www.odva.org). The robot can have connections to both Ethernet and DeviceNet networks and this feature provides the capability to route messages between two networks for configuration and diagnostics purposes.

This feature allows you to configure and manage the local robot DeviceNet network from personal computers (PCs) connected to their plant's Ethernet network. This eliminates someone having to connect a laptop PC to the physical robot in the DeviceNet network for certain third party device configuration and diagnostics functions. The PC software used is typically a tool such as RS-Networx for DeviceNet, which supports the following features:

- CIP routing
- Functions such as remotely configuring a device attached to the local DeviceNet network
- Network "who" of the local robot in the DeviceNet network from the PC connected to Ethernet.
- Explicit Messaging Connections to the devices in DeviceNet via "Class Instance Editor"

For more detailed technical information on EtherNet/IP to DeviceNet routing, refer to Chapter 10, Bridging and Routing, in *The CIP Common specification (EtherNet/IP specification volume 1).*

5.2 GUIDELINES

Review the following guidelines before you use routing:

- Any errors returned from devices in DeviceNet are posted in the third party application software. (e.g. RSNetworx for DeviceNet)
- The G3_ONLY feature is supported on SST DN3 cards only
- Routing is limited to explicit messages directed to the connection manager object using the unconnected send service. Routing of I/O is not supported.
- Do not change the status of the devices while Routing is performed. This disrupts the connection between the master (robot in the DeviceNet network) and slave (device in DeviceNet network).

5.3 SETTING UP ETHERNET/IP TO DEVICENET ROUTING

EtherNet/IP to DeviceNet Routing is installed with the EthernetIP I/O RTR option. The default method for using Routing is to have it configured to start when the controller is turned on. Even though the EtherNet/IP to DeviceNet Routing interface screen is available (Refer to 13.3), some features can only be configured by setting system variables.

Note $EIP_RTR.$G3_ONLY is added to protect I/O performance. Group 2 devices need to set up Predefined Master/Slave connections to exchange Explicit Messaging packets and I/O packets. The priorities of Group 2 messages are predefined by ODVA. For example, Explicit Message request and response have higher priority than the Master's I/O poll request. Thus, there is a chance that an Explicit Message request and response will win over a Master's I/O poll request. This could affect I/O performance. When $EIP_RTR.$G3_ONLY is enabled, routing to Group 2 devices are not allowed.

$EIP_RTR.$DIN_NUM is added to enable/disable routing dynamically based on a digital input. For example, this feature can be useful for I/O sensitive processes such as dispensing around a windshield. When the corresponding DIN is ON, the routing is disabled. Refer to Table 5–1 for information on the system variables used in routing.

Table 5–1. EtherNet/IP to DeviceNet Routing System Variables

System Variable	Default Value	Description
$EIP_RTR.$ENABLE	TRUE	Enable EIP Router*
$EIP_RTR.$BOARD_1	FALSE	When this is enabled, DeviceNet Board 1 routes packets from EtherNet to DeviceNet.
$EIP_RTR.$BOARD_2	TRUE	When this is enabled, DeviceNet Board 2 routes packets from EtherNet to DeviceNet.
$EIP_RTR.$BOARD_3	FALSE	When this is enabled, DeviceNet Board 3 routes packets from EtherNet to DeviceNet.
$EIP_RTR.$BOARD_4	FALSE	When this is enabled, DeviceNet Board 4 routes packets from EtherNet to DeviceNet.
$EIP_RTR.$G3_ONLY	FALSE	When this is enabled, CIP packets are only routed to Group 3 only (UCMM capable) devices.
$EIP_RTR.$DIN_NUM	0	When DIN input port is specified and input port is ON, no packets are routed to DeviceNet network.*

* $ENABLE and $DIN_NUM can be set via user interface screen.

5.4 USING ETHERNET/IP TO DEVICENET ROUTING

After system variables are configured, enabling and disabling Router and Setting DIN[] can be done using the EtherNet/IP Configuration screen. Refer to Procedure 5-1 to set up EtherNet/IP to DeviceNet Routing. Refer to Procedure 5-2 for information on Routing using RSNetworx for DeviceNet.

Procedure 5-1 Setting Up EtherNet/IP to DeviceNet Routing

Conditions

- The controller is turned on.

Steps

1. Press MENUS.
2. Select I/O.
3. Press F1, [TYPE].
4. Select EtherNet/IP. You will see a screen similar to the following.

```
I/O EtherNet/IP              JOINT  10 %
EtherNet/IP List(Rack 89)         1/8
  Description  TYP  Enable  Status  Slot
 1      Conn1  ADP  TRUE    ONLINE   1
 2      Conn2  ADP  FALSE  OFFLINE   2
 3      Conn3  ADP  FALSE  OFFLINE   3
 4      Conn4  ADP  FALSE  OFFLINE   4
 5      Conn5  ADP  FALSE  OFFLINE   5
 6      Conn6  ADP  FALSE  OFFLINE   6
 7      Conn7  ADP  FALSE  OFFLINE   7
 8      Conn8  ADP  FALSE  OFFLINE   8
```

5. Press NEXT and then press F3 [RTR]. You will see a screen similar to the following.

```
I/O EtherNet/IP              JOINT  10 %
Router configuration :            1/2


     Enable : TRUE
     Disable when DIN[   0] is ON
```

6. Make sure Enable is set TRUE. If it is not, move the cursor to Enable, and press F4, TRUE.
7. If you want to disable routing when a particular DIN is ON, move the cursor to DIN[ ], and type the port number. Note that DIN[] port 0 does not exist in the robot I/O. Thus, DIN[0] would not disable Routing unless the port number is changed.

   Note This feature is optional and can be useful for I/O sensitive processes such as dispensing around a windshield.

Procedure 5-2 Routing using RSNetworx for DeviceNet

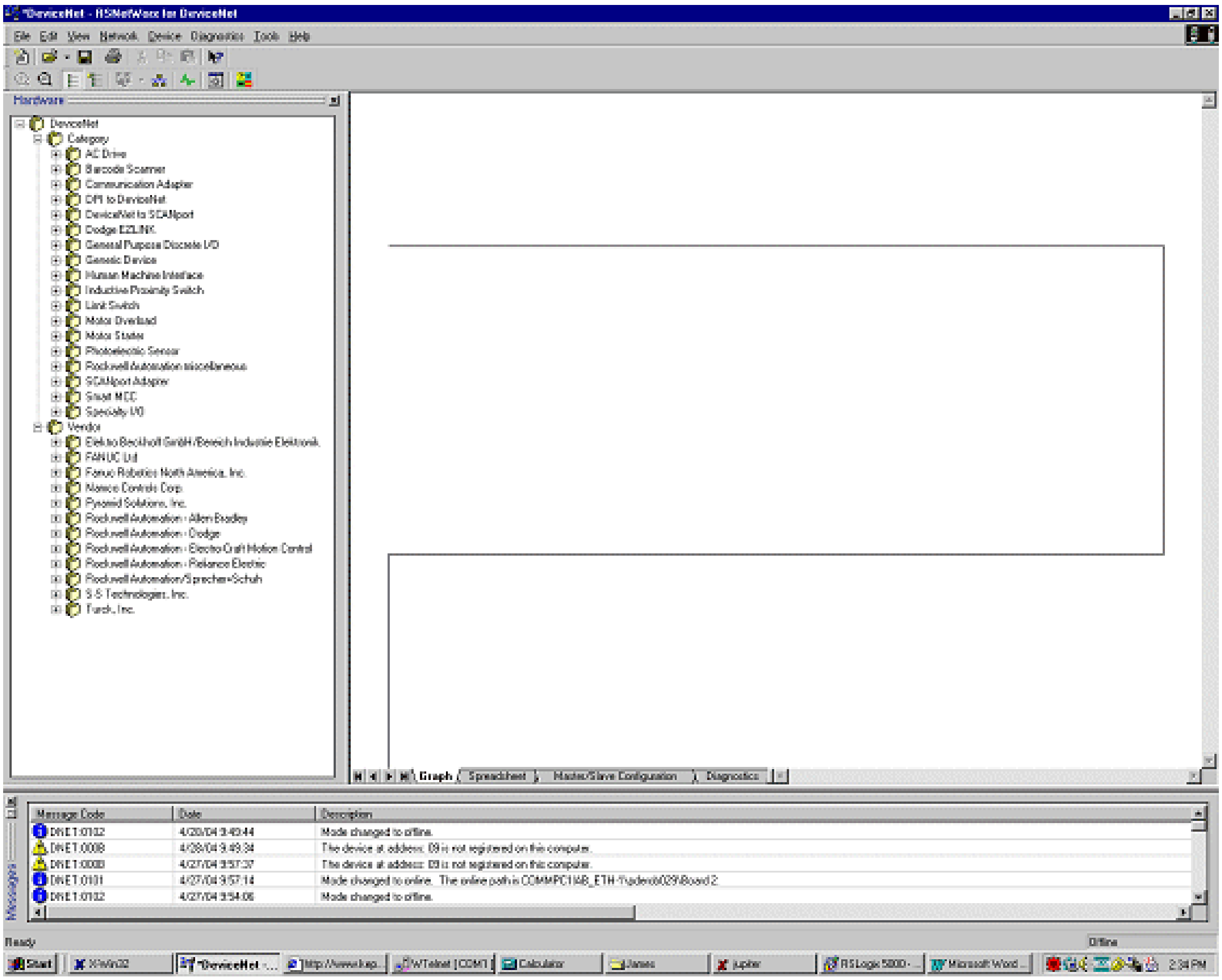
Conditions

- $EIP_RTR.$BOARD_2 is set TRUE.
- Board 2 in DeviceNet is ONLINE.
- You are using a personal computer (PC) with Microsoft NT, and have installed RSNetworx for DeviceNet.

Steps

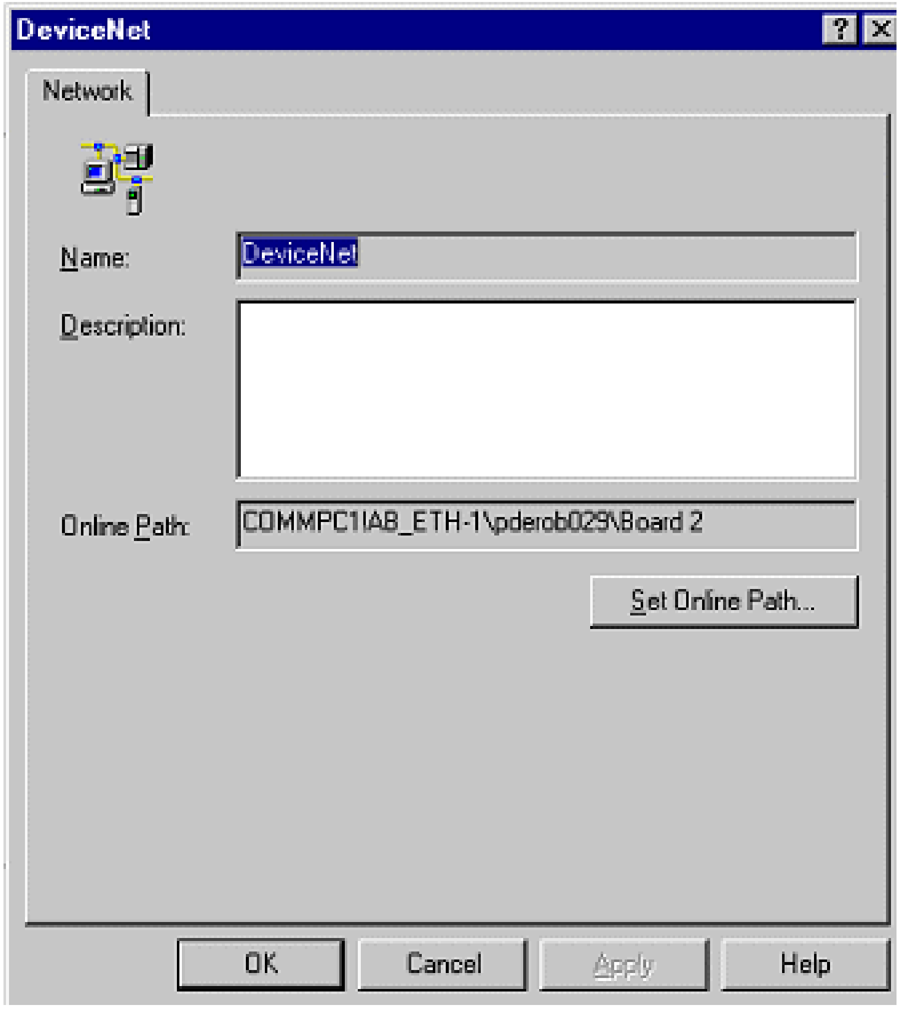
1. Launch RSNetworx for DeviceNet. For example on your PC, select Start, Programs, Rockwell software, RSNetworx, and then RSNetworx for DeviceNet. You will see a screen similar to the following.

Figure 5–1. RSNetworx for DeviceNet First Screen



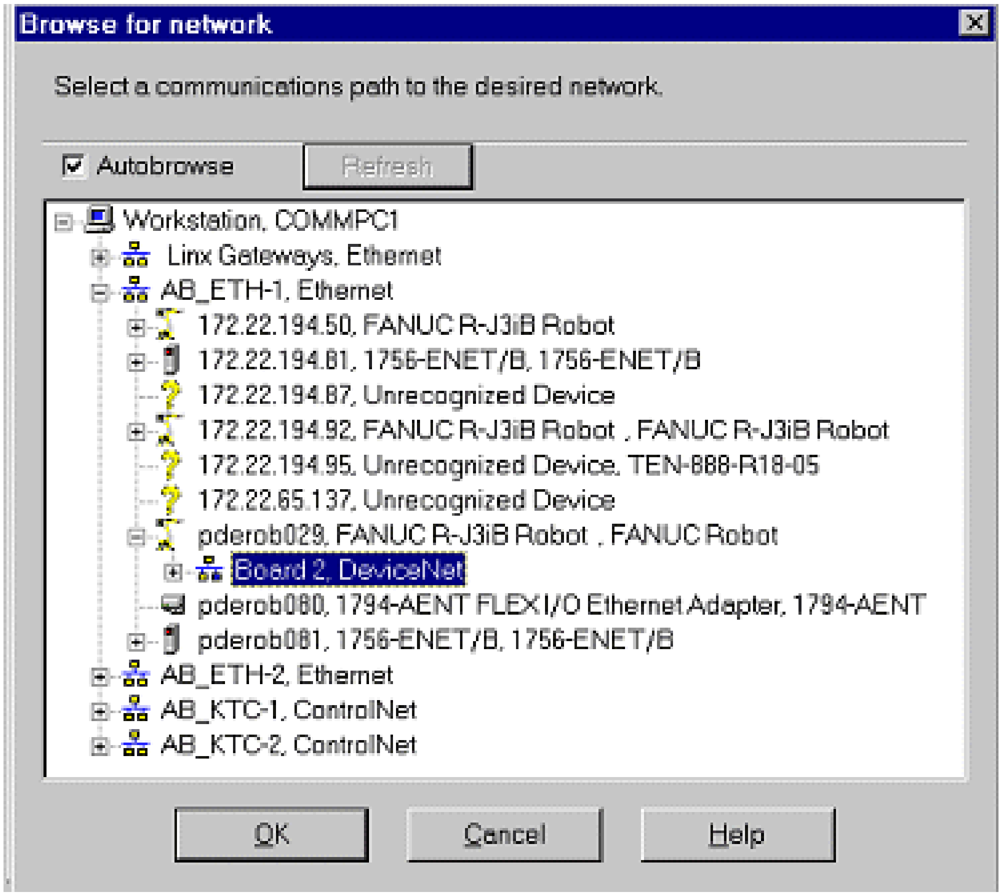
2. Select Properties under Network tab. You will see a screen similar to the following.

Figure 5–2. DeviceNet Network Screen



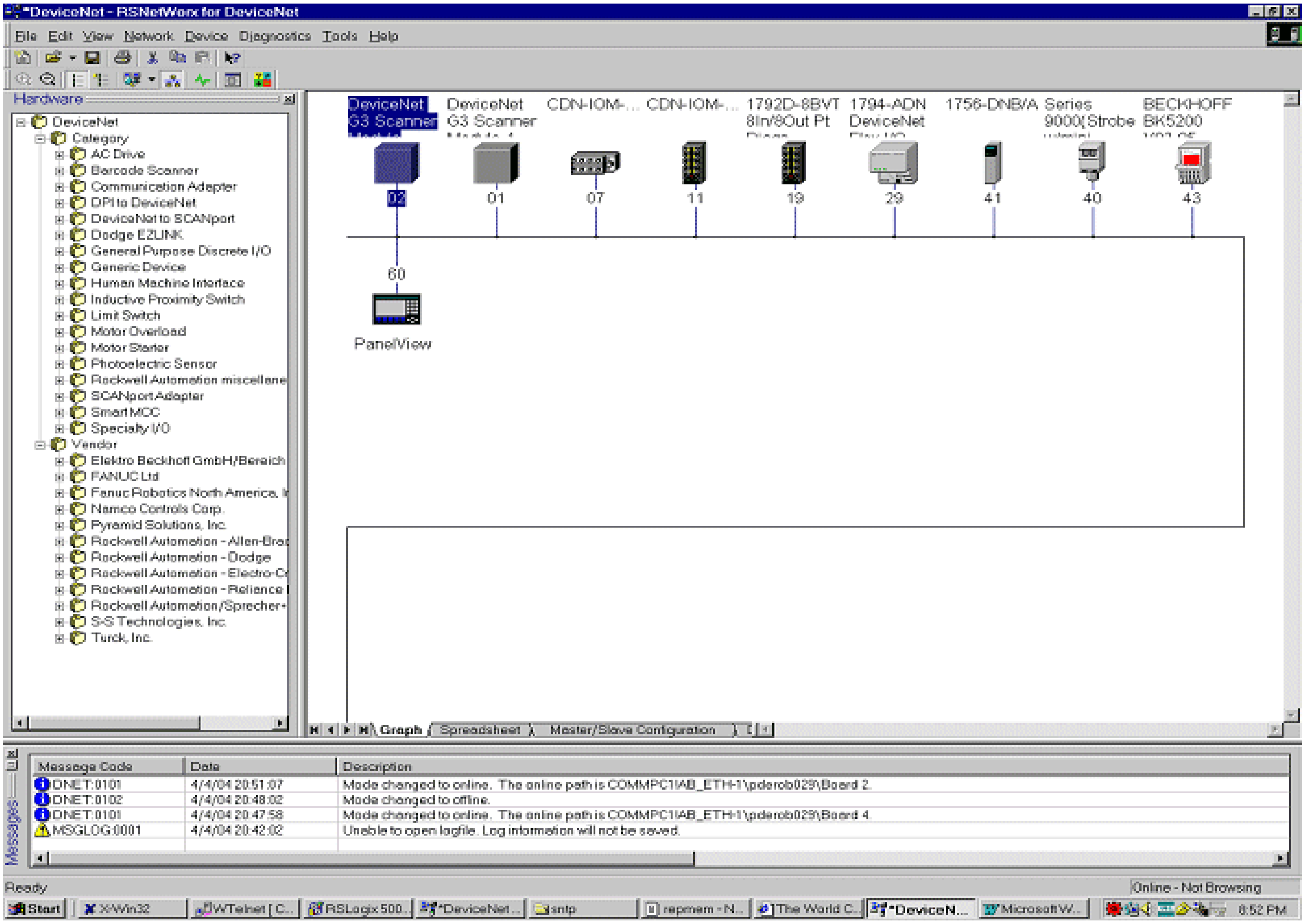
3. Click Set Online Path.. You will see a screen similar to the following.

Figure 5–3. Set Online Path.. Screen



4. Find the robot under Ethernet channel. In this example, the robot named pderob029 is under AB-ETH1, EtherNet.
5. Select Board 2, DeviceNet under the robot (pderob029), and click OK. You should see new path in Online Path text box. This is shown as COMMPC1\AB_ETH-1\pderob029\Board 2 in this example.
6. Click Apply, and then click OK. After the path is set, you are ready to browse the devices in DeviceNet.
7. Select Online under the Network tab.
8. Click OK to begin browsing the network. After this is done, you will see the screen similar to the following.

Figure 5–4. Local DeviceNet



9. At this point you can use the Class Instance Editor to get or set device parameters. For more information on how to use RSNetworx for DeviceNet, refer to the *RSNetworx for DeviceNet Manual* .

Chapter 6

I/O CONFIGURATION

Contents

6.1 OVERVIEW

This chapter describes how to make EtherNet/IP I/O available within the robot by mapping it to digital, group, and UOP I/O points. Scanner connection I/O can also map to analog. Refer to Section 4.2.5 .

This chapter also describes procedures for backing up and restoring the EtherNet/IP and I/O configurations.

6.2 MAPPING I/O ON THE ROBOT

The EtherNet/IP I/O can be mapped to digital, group, or UOP I/O points within the robot scanner connection. I/O can also map to analog. This is similar to mapping other I/O points on the robot where the rack, slot, and starting point number are used to map physical I/O to logical I/O within the I/O map.

All EtherNet/IP I/O uses rack 89. The slot number for each connection is shown in the EtherNet/IP Status screen.

Use Procedure 6-1 to map I/O on the robot.

Procedure 6-1 Mapping I/O on the Robot

Steps

1. Press MENUS.
2. Select I/O.
3. Press F1, [TYPE], and select Digital, Group, UOP, or analog (analog is supported on scanner connections).
4. Press F2, CONFIG.
5. Set the Range to the appropriate value. For analog, set the channel to the appropriate value.
6. Set the Rack to 89 and set the appropriate slot number and starting point as required.

   Note Refer to the Input/Output (I/O) Setup chapter in the application-specific *Setup and Operations Manual* for additional information on I/O configuration.

Note See Procedure 4-3 for additional information on configuring Analog I/O on the controller.

In some application software the I/O is automatically configured when you turn on the controller. The system variable $IO_AUTO_CFG (for digital I/O) and $IO_AUTO_UOP (for UOP I/O) controls this behavior. If the system has already automatically configured the I/O, and sizes are changed, the I/O assignments can be cleared to force the system to remap all the I/O. This is done by clearing assignments (CLR_ASG). Use Procedure 6-2 to clear I/O assignments.

Procedure 6-2 Clearing I/O Assignments

Steps

1. Press MENUS.
2. Select I/O.
3. Press F1, [TYPE].
4. Select Link Device.
5. Press F5, CLR_ASG.
6. To remap all I/O, turn the controller off and back on.

   Note This clears all I/O assignments. The I/O will be remapped when you turn off then turn on the controller based on the settings of $IO_AUTO_CFG (for digital I/O) and $IO_AUTO_UOP (for UOP I/O).

6.3 BACKING UP AND RESTORING ETHERNET/IP AND I/O CONFIGURATION

There are two files which contain information on the configuration of EtherNet/IP and I/O mappings :

- DIOCFGSV.IO contains general I/O configuration and all I/O mappings (for example, mappings between EtherNet/IP and digital, group, and UOP I/O).
- SYSEIP.SV contains EtherNet/IP specific configuration including all adapter and scanner settings.

Use Procedure 6-3 to back up files manually.

Use Procedure 6-4 to do a full application backup, which includes DIOCFGSV.IO and SYSEIP.SV.

Procedure 6-3 Backing Up Files Manually

Steps

1. Select the default file device where files will be saved:
   a. Press MENUS.
   b. Select File.
   c. Press F5, [UTIL], and choose SET DEVICE.
   d. Select the device to which you want to save the files.
2. Save DIOCFGSV.IO:
   a. Press MENUS.

b. Select I/O.

c. Press F1, [TYPE], and choose DIGITAL.

d. Press FCTN.

e. Select Save to save DIOCFGSV.IO to the default device.

3. Save SYSEIP.SV:

a. Press MENUS.

b. Select I/O.

c. Press F1, [TYPE], and choose EtherNet/IP.

d. Press FCTN.

e. Select Save to save SYSEIP.SV to the default device.

Procedure 6-4 Performing a Full Application Backup

Steps

1. Select the default file device (where files will be saved):

a. Press MENUS.

b. Select File.

c. Press F5, [UTIL], and choose SET DEVICE.

d. Select the device to which you want to save the files.

2. Press F4, [BACKUP], and choose "All of above".

Chapter 7

EXPLICIT MESSAGING

Contents

7.1 Overview

The robot controller is an explicit message server and supports connected and unconnected explicit messaging. Up to six explicit message connections are supported. The EtherNet/IP Adapter option must be loaded to support this functionality. The following objects are supported by the controller:

- Identity Object (0x01)
- Message Router Object (0x02)
- Assembly Object (0x04)
- Connection Manager Object (0x06)
- Vendor Specific Register Object--Integers (0x6B)
- Vendor Specific Register Object--Reals (0x6C)
- Vendor Specific Active Alarm Object (0xA0)
- Vendor Specific Alarm History Object (0xA1)
- Vendor Specific Motion Alarm Object (0xA2)
- Vendor Specific System Alarm Object (0xA3)
- Vendor Specific Application Alarm Object (0xA4)
- Vendor Specific Recovery Alarm Object (0xA5)
- Vendor Specific Communications Alarm Object (0xA6)
- Connection Configuration Object (0xF3)
- Port Object (0xF4)
- TCPIP Object (0xF5)
- Ethernet Link Object (0xF6)

This chapter does not go into the details of the standard CIP objects defined by ODVA. This chapter will instead document FANUC's Vendor Specific Alarm and Register objects, and the Assembly Object instances numbers for accessing I/O on the robot controller.

In general, no configuration is required on the robot controller to use explicit messaging. I/O must be configured on the robot if it is to be accessed through the Assembly Object.

7.2 Robot Explicit Messaging Client

7.2.1 Overview

The Robot Explicit Messaging Client allows you to send simple EtherNet/IP explicit messages to other EtherNet/IP enabled devices on the network. You can access this option from the main EtherNet/IP screen and execute queries from the teach pendant.

The explicit messaging feature implements the Get Attribute Single, and Set Attribute Single services.

Explicit messaging clients require up to five values for configuration. These values are usually described in hexadecimal notation, with the exception of the IP address. These values are shown in Table 7–1 .

Table 7–1. Explicit Messaging Configuration Values

ITEM	DESCRIPTION
IP Address	The address of the remote device to be queried.
Class	The class of Object to which the explicit message is being sent.
Instance	The instance number defines which instance of the class will receive the message.
Attribute	Defines which attribute of the instance is being accessed.
Service	The action to be performed. The robot explicit messaging client only supports the Get Attribute Single, and Set Attribute Single services.

An explicit message configuration file can also be created that performs a batch of commands. (Refer to Procedure 7-3). Normally this is used with the Set Attributes Single service to configure a remote device.

Procedure 7-1 describes how to use the Get Attribute Single service. Procedure 7-2 describes how to use the Set Attribute Single service.

Procedure 7-1 Get Attribute Single Service

1. Press MENUS.
2. Select I/O.
3. Press F1, [TYPE].
4. Select EtherNet I/P.

5. Press NEXT and then press F2 [EXP-MSG]. (If you scrolled to an active connection, that connection's IP address will be used as the default IP address). You will see a screen similar to the following.

```
I/O EtherNet/IP       JOINT 10 %

Explicit Message Query           1/8
     Input Mode:        Manual
     IP Addr:
     Class:             1
     Instance:          1
     Attribute:         1
     Service:           Get Att
     Value Size:        Byte(1)
     Value:             0
```

6. Select Input Mode, and choose Manual (the default).
7. Set the IP Addr field to the address of the remote device.
8. Set the Class, Instance, and Attribute fields. This information should either describe standard CIP objects, or be provided by the vendor of the remote device.
9. Select Service field and choose Get Att (the default).
10. Press F4 [EXEC] (you will have to cursor to a field other then Services and Value Size to see the F4 [EXEC] function). The robot will attempt to send the Explicit Message query to the device. A valid response or any errors will be displayed at the bottom of the screen.

Procedure 7-2 Set Attribute Single Service

1. Press MENUS.
2. Select I/O.
3. Press F1, [TYPE].
4. Select EtherNet I/P.
5. Press NEXT and then press F2 [EXP-MSG]. (If you scrolled to an active connection, that connection's IP address will be used as the default IP address). You will see a screen similar to the following.

```
I/O EtherNet/IP       JOINT 10 %

Explicit Message Query           1/8
     Input Mode:        Manual
     IP Addr:
     Class:             1
     Instance:          1
     Attribute:         1
     Service:           Get Att
     Value Size:        Byte(1)
     Value:             0
```

6. Select Input Mode, and choose Manual (the default).
7. Set the IP Addr field to the address of the remote device.
8. Set the Class, Instance, and Attribute fields. This information should either describe standard CIP objects, or be provided by the vendor of the remote device.
9. Select Service field and choose Set Att.
10. Set the Value Size, and Value fields. The supported Value Size fields are Byte, Word, and Long.
11. Press F4, [EXEC] (you will have to cursor to a field other then Services and Value Size to see the F4 [EXEC] function). The robot will attempt to send the Explicit Message query to the device. The result of the query or any errors will be displayed at the bottom of the screen.

Procedure 7-3 Explicit Messaging Batch File Method

1. Press MENUS.
2. Select I/O.
3. Press F1, [TYPE].
4. Select EtherNet I/P.
5. Press NEXT and then press F2 [EXP-MSG]. You will see a screen similar to the following.

```
I/O EtherNet/IP       JOINT 10 %
Explicit Message Query           1/8

     Input Mode:        Manual
     IP Addr:
     Class:             1
     Instance:          1
     Attribute:         1
     Service:           Get Att
     Value Size:        Byte(1)
     Value:             0
```

6. Move the cursor to Input Mode, and choose File. You will see a screen similar to the following.

```
I/O EtherNet/IP       JOINT 10 %

Explicit Message Query           1/2
  Device:   MC:\
     Input Mode:         File
     Config File Name:   None
```

7. Move the cursor to the Config File Name field. Press F2, [DEVICE] to select the device where the explicit message config file is located.
8. Press F4, [CHOICE] to select the config file. (Only files with a .EM extension can be selected.)
9. Press F3, [EXEC] to execute the config file. The robot will attempt to send the Explicit Message query to the device. The result of the query or any errors will be displayed at the bottom of the screen.

7.2.2 Creating a Configuration File for the Batch File Method

The configuration file format is documented in this section. All files must be name with the .EM file extension (for example, MYCONFIG.EM). Lines beginning with ‘*’ are comments. Comments and blank lines are ignored. See Batch File Example .

The following seven (7) lines MUST exist for each query followed by a space and corresponding value:

```
QUERY:
IPADDR:
CLASS:
INSTANCE:
ATTRIBUTE:
SIZE:
VALUE:
```

There can be multiple queries within a file. The Set Attribute Single service is assumed in all cases. Each Query begins with a query number, which is unique and generally sequential.

The Size field refers to the Data Size of the parameter (Value), the following three are supported.

- 1 (BYTE or 1 byte)
- 2(WORD or 2 bytes)
- 4 (LONG or 4 bytes)

Batch File Example

```
*  File Name: EMCFG.EM
*  Author: Joe User
*  Date: 03/15/2004
* File must be saved with an .EM extension
* Lines beginning with '*' are comments.
* Comments and blank lines are ignored.
* Following 7 lines MUST exist for each query.
* There can be multiple queries within a file.
* The "SET ATT" service is assumed in all cases.
* Each Query begins with a query number, which is
* unique and generally sequential.
* Size field refers to the Data Size of the
* parameter, the following three are supported
* 1 (BYTE or 1 byte), 2(WORD or 2 bytes), 4 (LONG or 4 bytes)
QUERY: 1
IPADDR: 172.22.200.147
CLASS: 15
INSTANCE: 2
ATTRIBUTE: 1
SIZE: 1
VALUE: 4
QUERY: 2
IPADDR: 172.22.200.147
CLASS: 15
INSTANCE: 3
ATTRIBUTE: 1
SIZE: 2
VALUE: 300
```

7.3 Remote Explicit Messaging Client Configuration

Explicit messaging clients require up to four values for configuration. These values are usually described in hexadecimal notation. These values are shown in Table 7–2 .

Table 7–2. Configuration Values

ITEM	DESCRIPTION
Class	The class of Object to which the explicit message is being sent.
Instance	The instance number defines which instance of the class will receive the message.

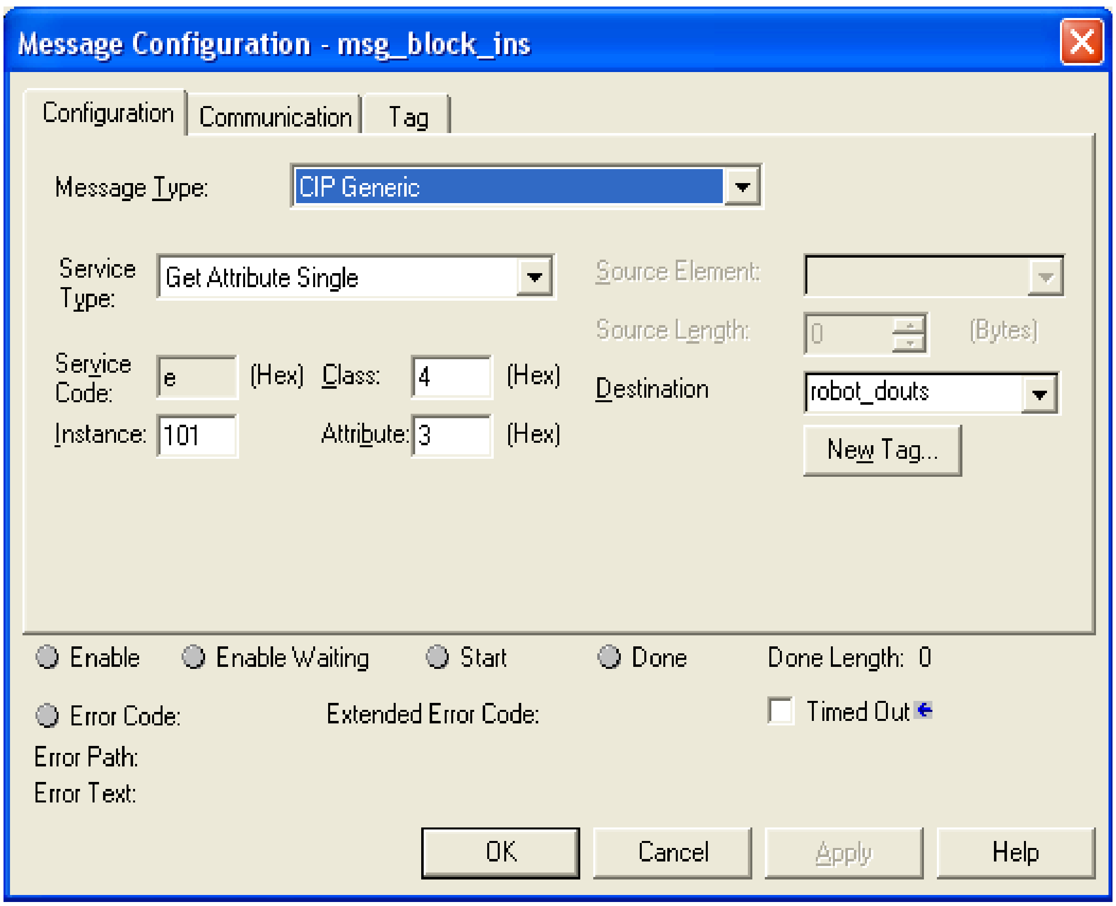
Table 7–2. Configuration Values (Cont'd)

Attribute	Defines which attribute of the instance is being accessed.
Service	The action to be performed.

These values are documented in this manual for all FANUC's Vendor Specific Objects.

Here is an example of configuring an Explicit Message in RSLogix5000. Note the Service Code, Class, Instance and Attribute fields.

Figure 7–1. Message Configuration



7.4 Vendor Specific Register Object (0x6B) and (0x6C)

Numeric registers can be read and written through FANUC's Registers Object.

Numeric registers on the robot controller can be one of two types: Integer type, or Real type. Normally, the type is transparent to the user. For example, if R[1] is set to a value of 49 (R[1] = 49), the numeric register will automatically be configured as an Integer type. However, if R[2] is set to a value of 1.61803 (R[2] = 1.61803), the numeric register will be automatically configured as a Real type.

Because of the format of the data transfer through explicit messaging, this automatic configuration cannot be done. Therefore, two Register Objects have been created: Register Object--Integers (0x6B), and Register Object--Reals (0x6C). To read or write a numeric register as an Integer value, the Register Object--Integers (0x6B) should be accessed. To read or write a numeric register as a Real value, the Register Object--Reals (0x6C) should be accessed.

Note that when writing to the Register Object--Integers, the numeric register will be changed to the Integer type. Likewise, when writing to the Register Object--Reals, the numeric register will be changed to the Real type.

However, when reading a numeric register using the Register Object--Integers, if the numeric register is configured as a Real type, an “Undefined Attribute” (0x14) error will be returned. Likewise, when reading a numeric register using the Register Object--Reals, if the numeric register is configured as an Integer type, an “Undefined Attribute” (0x14) error will be returned.

Note FANUC’s Register objects only allow reading and writing of the first 255 registers using the Get_Attribute_Single and Set_Attribute_Single services. The Get_Attribute_All service allows reading of up to the first 124 registers.

7.4.1 Instance Attributes

FANUC’s Register Objects support a single instance: instance 1. Each attribute in the instance corresponds to a register. For example, instance 1 corresponds to R[1] and instance 5 corresponds to R[5]. Refer to Table 7–3 .

Table 7–3. Instance Attributes

Attribute ID	Name	Data Type	Description of Attribute
1	R[1]	32-bit integer	Register 1
2	R[2]	32-bit integer	Register 2
...			
n-1	R[n-1]	32-bit integer	Register n-1
n*	R[n]	32-bit integer	Register n

*Where n is the total number of registers on the controller or 255, whichever is smaller.

7.4.2 Common Services

FANUC's Register Objects provide the Common Services at the Instance level shown in Table 7–4 . No Class level services are provided.

Table 7–4. Common Services

Service Code	Service Name	Description of Service
0E hex	Get_Attribute_Single	Returns the content of the specified attribute.
01 hex	Get_Attribute_All	Returns a listing of the object's attributes (See the Get_Attribute_All definition below).
10 hex	Set_Attribute_Single	Sets the specified attribute to the specified value.

7.4.2.1 Get_Attribute_All Response

At the Instance level, the attributes are returned in the order shown in Table 7–5 using little-endian byte-swapping.

Table 7–5. Get_Attribute_All Response

32–bit integer	Byte 0	Byte 1	Byte 2	Byte 3
1	Register 1 (R[1])			
2	Register 2 (R[2])			
...				
n-1	Register n-1 (R[n-1])			
n*	Register n (R[n])			

*Where n is the total number of registers on the controller or 124, whichever is smaller.

7.4.3 Errors

FANUC's Vendor Specific Register Objects will return the errors shown in Table 7–6 .

Table 7–6. FANUC's Vendor Specific Register Object Errors

Error Status	Error Description
Undefined Attribute (0x14)	Returned when the Register requested does not exist.
Unsupported Service (0x08)	Returned when the requested service is unsupported.
Undefined Class Instance (0x05)	Returned when the requested instance number is unsupported.

7.4.4 Examples

The examples below assume the numeric register(s) have type integer, and use object 0x6B. The same examples can be used for numeric registers of type Real by using Class 0x6C, instead of Class 0x6B.

7.4.4.1 Read Register 5

To read R[5] from the controller, assuming R[5]'s type is an integer, the explicit message client would be configured with the values shown in Table 7–7 .

Table 7–7. Client Configuration

Class	0x6B
Instance	0x01
Attribute	0x05
Service	0x0E

The explicit message server on the robot controller would return R[5] to the client as a 32-bit integer.

7.4.4.2 Read All Registers

To read up to the first 124 Registers from the controller, and assuming all these registers are configured as type integer, the explicit message client would be configured with the values shown in Table 7–8 .

Table 7–8. Read All Registers

Class	0x6B
Instance	0x01
Attribute	Not Required
Service	0x01

The explicit message server on the robot controller would return up to the first 124 Registers as an array of 32-bit integers as described in Section 7.4.2.1 .

7.4.4.3 Write Register 5

To write the integer value 49 to R[5], the explicit message client would be configured with the values shown in Table 7–9 .

Table 7–9. Write Value to R[5]

Class	0x6B
Instance	0x01
Attribute	0x05
Service	0x10
Value	49

The explicit message server on the robot controller would write the value 49 to R[5] as a 32-bit integer.

7.5 Vendor Specific Active Alarm Object (0xA0)

Information about Active Alarms can be read through FANUC's Active Alarm Object. Each instance of the object corresponds to an active alarm. For example, instance 1 corresponds to the most recent Active Alarm, and instance 5 corresponds to the 5th most recent Active Alarm.

7.5.1 Instance Attributes

Table 7–10. Instance Attributes

Attribute ID	Name	Data Type	Description of Attribute
1	Alarm ID	16-bit integer	The Alarm ID, or Alarm Code.
2	Alarm Number	16-bit integer	The Alarm Number
3	Alarm ID Cause Code	16-bit integer	The Cause Code of the Alarm ID.
4	Alarm Num Cause Code	16-bit integer	The Cause Code of the Alarm Number.
5	Alarm Severity	16-bit integer	The Alarm Severity.
6	Time Stamp	32-bit integer	The Alarm Time Stamp in 32-bit MS-DOS format.
7	Date/Time String	24 character string	The Alarm Time Stamp in a human readable string.
8	Alarm Message	80 character string	The Alarm Message in a human readable string.
9	Cause Code Message	80 character string	The Alarm Cause Code Message in a human readable string.
10	Alarm Severity String	24 character string	The Alarm Severity in a human readable string.

7.5.2 Common Services

FANUC's Active Alarm Object provides the following Common Services at the Instance level. No Class level services are provided. Refer to Table 7–11 .

Table 7–11. Common Services

Service Code	Service Name	Description of Service
0E hex	Get_Attribute_Single	Returns the content of the specified attribute.
01 hex	Get_Attribute_All	Returns a listing of the object's attributes (See the Get_Attribute_All definition below).

7.5.2.1 Get_Attribute_All Response

At the Instance level, the attributes are returned in the order shown in Table 7–12 using little-endian byte-swapping for 16-bit and 32-bit integers.

Table 7–12. Get_Attribute_All Responses

32-bit integer	Byte 0	Byte 1	Byte 2	Byte 3
1	Alarm ID		Alarm Number	
2	Alarm ID Cause Code		Alarm Num Cause Code	
3	Alarm Severity		PAD (All Zeros)	
4	Time Stamp			
5	Date/Time String (24 bytes)			
...	...			
11	Alarm Message (80 bytes)			
...	...			
31	Cause Code Message (80 bytes)			
...	...			
51	Alarm Severity String (24 bytes)			
...	...			

7.5.3 Errors

FANUC's Vendor Specific Active Alarm Object will return the errors shown in Table 7–13 .

Table 7–13. Errors

Error Status	Error Description
Undefined Attribute (0x14)	Returned when the Alarm Attribute requested does not exist.
Unsupported Service (0x08)	Returned when the requested service is unsupported.
Undefined Class Instance (0x05)	Returned when the requested instance number does not exists. For example, if there is only 1 single active alarm, requesting the Active Alarm Object instance 2 will cause this error to be returned.

7.5.4 Examples

7.5.4.1 Read Most Recent Active Alarm Cause Code

To read the most recent active alarm's cause code from the controller, the explicit message client would be configured with the values shown in Table 7–14 .

Table 7–14. Read Most Recent Active Alarm Cause Code

Class	0xA0
Instance	0x01
Attribute	0x03
Service	0x0E

The explicit message server on the robot controller would return the cause code to the client as a 16-bit integer.

7.5.4.2 Read All Alarm Information from the Second Most Recent Active Alarm

To read all alarm information about the second most recent active alarm from the controller, the explicit message client would be configured with the values shown in Table 7–15 .

Table 7–15. Read All Alarm Information from the Second Most Recent Active Alarm

Class	0xA0
Instance	0x02
Attribute	Not Required
Service	0x01

The explicit message server on the robot controller would return all Registers as described in Section 7.5.2.1 .

7.6 Vendor Specific Alarm History Object (0xA1)

Information about the Alarm History can be read through FANUC's Alarm History Object. Each instance of the object corresponds to an alarm in the history. For example, instance 1 corresponds to the most recent Alarm in the alarm history, and instance 5 corresponds to the 5th most recent Alarm.

7.6.1 Instance Attributes

The instance attributes are identical to those of FANUC's Active Alarm Object (0xA0) and are documented in Section 7.5.1 .

7.6.2 Common Services

The common services are identical to those of FANUC's Active Alarm Object (0xA0) and are documented in Section 7.5.2 .

7.6.3 Errors

The errors are identical to those of FANUC's Active Alarm Object (0xA0) and are documented in Section 7.5.3 .

7.6.4 Examples

7.6.4.1 Read Most Recent Alarm Cause Code

To read the most recent alarm's cause code from the controller, the explicit message client would be configured with the values shown in Table 7–16 .

Table 7–16. Read Most Recent Alarm Cause Code

Class	0xA1
Instance	0x01
Attribute	0x03
Service	0x0E

The explicit message server on the robot controller would return the cause code to the client as a 16-bit integer.

7.6.4.2 Real All Alarm Information from the Second Most Recent Alarm

To read all alarm information about the second most recent alarm from the controller, the explicit message client would be configured with the values shown in Table 7–17 .

Table 7–17. Real All Alarm Information from the Second Most Recent Alarm

Class	0xA1
Instance	0x02
Attribute	Not Required
Service	0x01

The explicit message server on the robot controller would return all Registers as described in Section 7.5.2.1 .

7.7 Vendor Specific Motion Alarm Object (0xA2)

Information about Motion Alarms can be read through FANUC's Motion Alarm Object. Each instance of the object corresponds to a motion alarm. For example, instance 1 corresponds to the most recent motion alarm, and instance 5 corresponds to the 5th most recent motion alarm.

7.7.1 Instance Attributes

The instance attributes are identical to those of FANUC's Active Alarm Object (0xA0) and are documented in Section 7.5.1 .

7.7.2 Common Services

The common services are identical to those of FANUC's Active Alarm Object (0xA0) and are documented in section Section 7.5.2 .

7.7.3 Errors

The errors are identical to those of FANUC's Active Alarm Object (0xA0) and are documented in Section 7.5.3 .

7.7.4 Examples

7.7.4.1 Read Most Recent Motion Alarm Cause Code

To read the most recent motion alarm's cause code from the controller, the explicit message client would be configured with the values shown in Table 7–18 .

Table 7–18. Read Most Recent Motion Alarm Cause Code

Class	0xA2
Instance	0x01
Attribute	0x03
Service	0x0E

The explicit message server on the robot controller would return the cause code to the client as a 16-bit integer.

7.7.4.2 Read All Alarm Information from the Second Most Recent Motion Alarm

To read all alarm information about the second most recent motion alarm from the controller, the explicit message client would be configured with the values shown in Table 7–19 .

Table 7–19. Read All Alarm Information from the Second Most Recent Motion Alarm

Class	0xA2
Instance	0x02
Attribute	Not Required
Service	0x01

The explicit message server on the robot controller would return all Registers as described in Section 7.5.2.1 .

7.8 Vendor Specific System Alarm Object (0xA3)

Information about System Alarms can be read through FANUC's System Alarm Object. Each instance of the object corresponds to a system alarm. For example, instance 1 corresponds to the most recent system alarm, and instance 5 corresponds to the 5th most recent system alarm.

7.8.1 Instance Attributes

The instance attributes are identical to those of FANUC's Active Alarm Object (0xA0) and are documented in Section 7.5.1 .

7.8.2 Common Services

The common services are identical to those of FANUC's Active Alarm Object (0xA0) and are documented in Section 7.5.2 .

7.8.3 Errors

The errors are identical to those of FANUC's Active Alarm Object (0xA0) and are documented in Section 7.5.3 .

7.8.4 Examples

7.8.4.1 Read Most Recent System Alarm Cause Code

To read the most recent system alarm's cause code from the controller, the explicit message client would be configured with the following values:

Table 7–20. Read Most Recent System Alarm Cause Code

Class	0xA3
Instance	0x01
Attribute	0x03
Service	0x0E

The explicit message server on the robot controller would return the cause code to the client as a 16-bit integer.

7.8.4.2 Read All Alarm Information from the Second Most Recent System Alarm

To read all alarm information about the second most recent system alarm from the controller, the explicit message client would be configured with the following values:

Table 7–21. Read All Alarm Information from the Second Most Recent System Alarm

Class	0xA3
Instance	0x02
Attribute	Not Required
Service	0x01

The explicit message server on the robot controller would return all Registers as described in Section 7.5.2.1 .

7.9 Vendor Specific Application Alarm Object (0xA4)

Information about Application Alarms can be read through FANUC's Application Alarm Object. Each instance of the object corresponds to an application alarm. For example, instance 1 corresponds to the most recent application alarm, and instance 5 corresponds to the 5th most recent application alarm.

7.9.1 Instance Attributes

The instance attributes are identical to those of FANUC's Active Alarm Object (0xA0) and are documented in Section 7.5.1 .

7.9.2 Common Services

The common services are identical to those of FANUC's Active Alarm Object (0xA0) and are documented in Section 7.5.2 .

7.9.3 Errors

The errors are identical to those of FANUC's Active Alarm Object (0xA0) and are documented in Section 7.5.3 .

7.9.4 Examples

7.9.4.1 Read Most Recent Application Alarm Cause Code

To read the most recent alarm's cause code from the controller, the explicit message client would be configured with the values shown in Table 7–22 .

Table 7–22. Read Most Recent Application Alarm Cause Code

Class	0xA4
Instance	0x01
Attribute	0x03
Service	0x0E

The explicit message server on the robot controller would return the cause code to the client as a 16-bit integer.

7.9.4.2 Read All Alarm Information from the Second Most Recent Application Alarm

To read all alarm information about the second most recent application alarm from the controller, the explicit message client would be configured with the values shown in Table 7–23 .

Table 7–23. Read All Alarm Information from the Second Most Recent Application Alarm

Class	0xA4
Instance	0x02
Attribute	Not Required
Service	0x01

The explicit message server on the robot controller would return all Registers as described in Section 7.5.2.1 .

7.10 Vendor Specific Recovery Alarm Object (0xA5)

Information about Recovery Alarms can be read through FANUC's Recovery Alarm Object. Each instance of the object corresponds to a recovery alarm. For example, instance 1 corresponds to the most recent recovery alarm, and instance 5 corresponds to the 5th most recent recovery alarm.

7.10.1 Instance Attributes

The instance attributes are identical to those of FANUC's Active Alarm Object (0xA0) and are documented in Section 7.5.1 .

7.10.2 Common Services

The common services are identical to those of FANUC's Active Alarm Object (0xA0) and are documented in Section 7.5.2 .

7.10.3 Errors

The errors are identical to those of FANUC's Active Alarm Object (0xA0) and are documented in Section 7.5.3 .

7.10.4 Examples

7.10.4.1 Read Most Recent Recovery Alarm Cause Code

To read the most recent recovery alarm's cause code from the controller, the explicit message client would be configured with the values shown in Table 7–24 .

Table 7–24. Read Most Recent Recovery Alarm Cause Code

Class	0xA5
Instance	0x01
Attribute	0x03
Service	0x0E

The explicit message server on the robot controller would return the cause code to the client as a 16-bit integer.

7.10.4.2 Read All Alarm Information from the Second Most Recent Recovery Alarm

To read all alarm information about the second most recent recovery alarm from the controller, the explicit message client would be configured with the values shown in Table 7–25 .

Table 7–25. Read All Alarm Information from the Second Most Recent Recovery Alarm

Class	0xA5
Instance	0x02

Table 7–25. Read All Alarm Information from the Second Most Recent Recovery Alarm (Cont'd)

Attribute	Not Required
Service	0x01

The explicit message server on the robot controller would return all Registers as described in Section 7.5.2.1 .

7.11 Vendor Specific Communications Alarm Object (0xA6)

Information about Communications Alarms can be read through FANUC's Communications Alarm Object. Each instance of the object corresponds to a communications alarm. For example, instance 1 corresponds to the most recent communications alarm and instance 5 corresponds to the 5th most recent communications alarm.

7.11.1 Instance Attributes

The instance attributes are identical to those of FANUC's Active Alarm Object (0xA0) and are documented in Section 7.5.1 .

7.11.2 Common Services

The common services are identical to those of FANUC's Active Alarm Object (0xA0) and are documented in Section 7.5.2 .

7.11.3 Errors

The errors are identical to those of FANUC's Active Alarm Object (0xA0) and are documented in Section 7.5.3 .

7.11.4 Examples

7.11.4.1 Read Most Recent Communication Alarm Cause Code

To read the most recent alarm's cause code from the controller, the explicit message client would be configured with the values shown in Table 7–26 .

Table 7–26. Read Most Recent Communication Alarm Cause Code

Class	0xA6
Instance	0x01
Attribute	0x03
Service	0x0E

The explicit message server on the robot controller would return the cause code to the client as a 16-bit integer.

7.11.4.2 Read All Alarm Information from the Second Most Recent Communications Alarm

To read the most recent alarm's cause code from the controller, the explicit message client would be configured with the values shown in Table 7–27 .

Table 7–27. Read All Alarm Information from the Second Most Recent Communications Alarm

Class	0xA6
Instance	0x02
Attribute	Not Required
Service	0x0E

The explicit message server on the robot controller would return all Registers as described in Section 7.5.2.1 .

7.12 Accessing I/O using Explicit Messaging

7.12.1 Accessing I/O Specific to an Implicit EtherNet/IP Connection

I/O can be accessed through ODVA's standard Assembly Object. Table 7–28 describes the Assembly Instance numbers that can be used to read or write I/O specific to an Implicit EtherNet/IP connection configured on a FANUC robot controller.

Note I/O can not be written to the writable assembly instances using explicit messaging while an active implicit connection to the instance is running.

Detailed instructions for mapping I/O on the robot controller can be found in Section 6.2 of this manual.

Table 7–28. Accessing I/O

Instance Number	Read/Write	Input/Output	Slot
101	r	output	1
102	r	output	2
103	r	output	3
104	r	output	4
105	r	output	5
106	r	output	6
107	r	output	7
108	r	output	8
109	r	output	9
110	r	output	10
111	r	output	11
112	r	output	12
113	r	output	13
114	r	output	14
115	r	output	15
116	r	output	16

Table 7–28. Accessing I/O (Cont'd)

Instance Number	Read/Write	Input/Output	Slot
151	r/w*	input	1
152	r/w*	input	2
153	r/w*	input	3
154	r/w*	input	4
155	r/w*	input	5
156	r/w*	input	6
157	r/w*	input	7
158	r/w*	input	8
159	r/w*	input	9
160	r/w*	input	10
161	r/w*	input	11
162	r/w*	input	12
163	r/w*	input	13
164	r/w*	input	14
165	r/w*	input	15
166	r/w*	input	16

* Only instances corresponding to Adapter connections are writable using Explicit Messaging; however, these instances will not be writable while an active implicit connection to the instance is running. If the corresponding connection is a Scanner connection, then the instance will be read-only.

For example, suppose the controller is configured with one Adapter connection on slot 1 with a 16-bit word of input and a 16-bit word of output, which are mapped to DI[1-16] and to DO[1-16] respectively. Once an implicit connection is establish to the adapter, the output values in DO[1-16] can be accessed through explicit messaging with the values shown in Table 7–29 .

Table 7–29. Output Values

Class	0x04
Instance	0x65
Attribute	0x03
Service	0x0E

And the input values in DI[1-16] can be accessed through explicit messaging with the values Table 7–30 .

Table 7–30. Input Values

Class	0x04
Instance	0x97
Attribute	0x03
Service	0x0E

7.12.2 Accessing General I/O

In addition to I/O mapped from EtherNet/IP connections, other types of I/O can be read with explicit messaging using the following Assembly Object Instance numbers.

Table 7–31. Accessing General I/O

I/O Type	Instance Number (hexadecimal)
Digital input	0x320
Digital output	0x321
Analog input	0x322
Analog output	0x323
Tool output	0x324
PLC input	0x325
PLC output	0x326
Robot digital input	0x327
Robot digital output	0x328
Brake output	0x329
Operator panel input	0x32a
Operator panel output	0x32b
Teach pendant digital input	0x32d
Teach pendant digital output	0x32e
Weld input	0x32f
Weld output	0x330
Group input	0x331
Group output	0x332
User operator panel input	0x333

Table 7–31. Accessing General I/O (Cont'd)

I/O Type	Instance Number (hexadecimal)
User operator panel output	0x334
Laser DIN	0x335
Laser DOUT	0x336
Laser AIN	0x337
Laser AOUT	0x338
Weld stick input	0x339
Weld stick output	0x33a
Memory image boolean	0x33b
Memory image DIN	0x33c
Dummy boolean port type	0x33d
Dummy numeric port type	0x33e
Process axes (ISDT)	0x33f
Internal operator panel input	0x340
Internal operator panel output	0x341
Flag (F[ ])	0x342
Marker (M[ ])	0x343

For example, the values shown in Table 7–32 would access all Digital Outputs (DOs) with explicit messaging.

Table 7–32. Accessing Digital Outputs

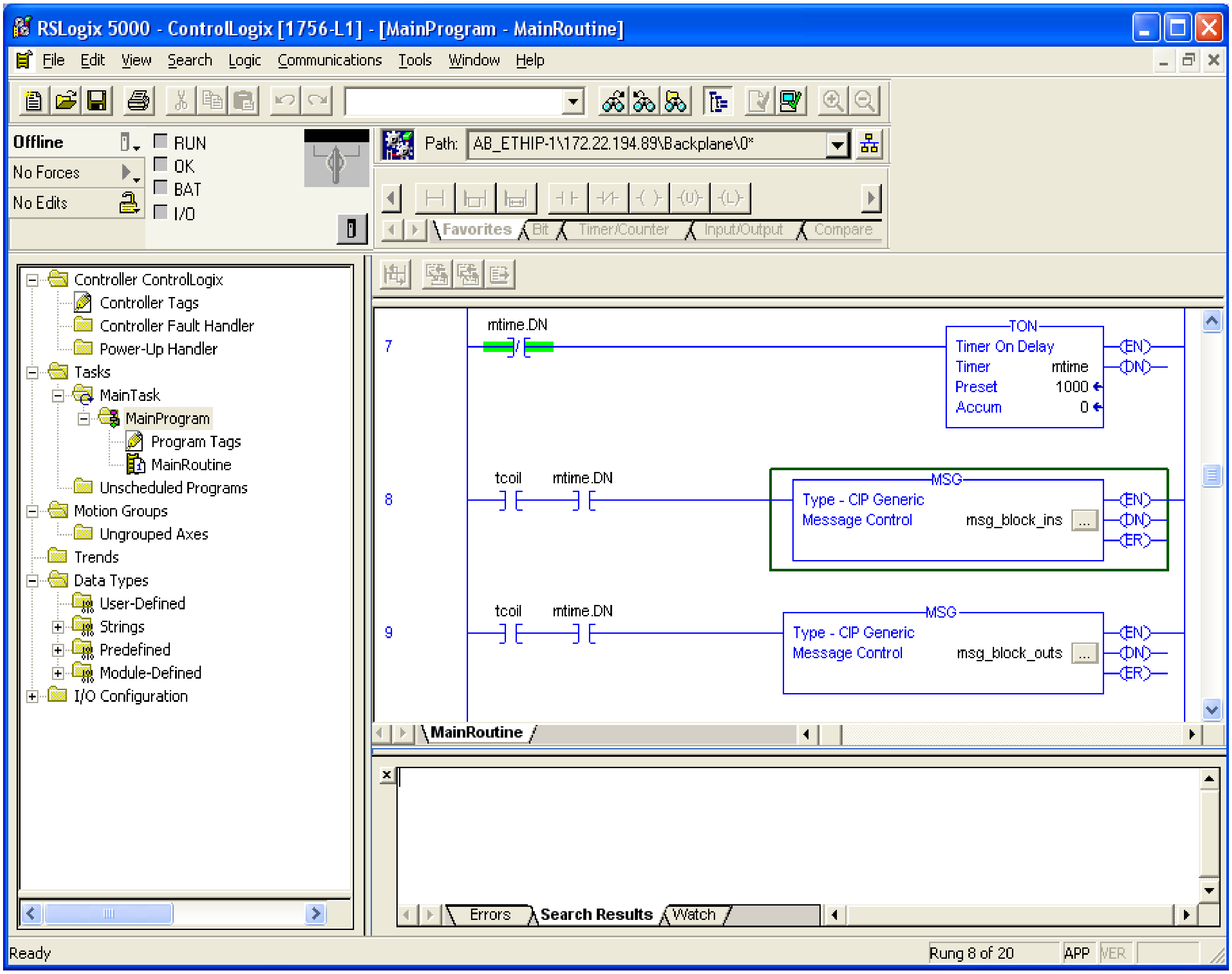
Class	0x04
Instance	0x321
Attribute	0x03
Service	0x0E

7.13 Using Explicit Messaging in RSLogix 5000

This section steps through an example of how to configure an I/O read and write operation on a robot controller using RSLogix5000. In this example, an I/O read and write is done on Rack 89 Slot 1 of the robot controller every 1000ms.

Three rungs are created in our main program. The first rung is a timer rung. This timer, mtime, will trigger read and write messages to be sent to the robot controller every 1000ms. The next two rungs have MSG blocks, where the individual Explicit Messages will be defined. Figure 7–2 shows the three rungs.

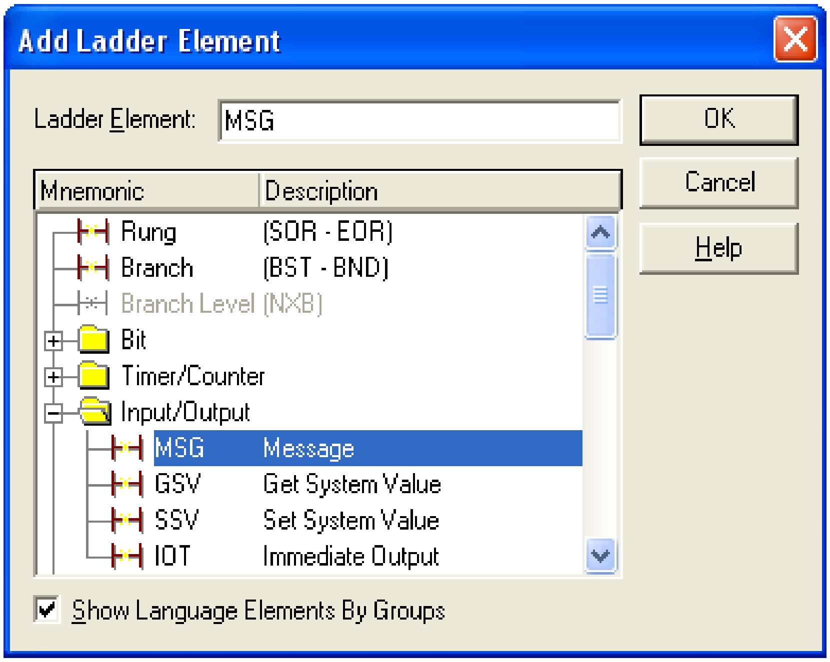
Figure 7–2. RsLogix 5000 Example Rungs



Note We have created the element tcoil to turn the sending of the two Explicit Messages on and off.

To add a message block, you will need to add a MSG ladder element. Figure 7–3 shows an example of adding a MSG ladder element.

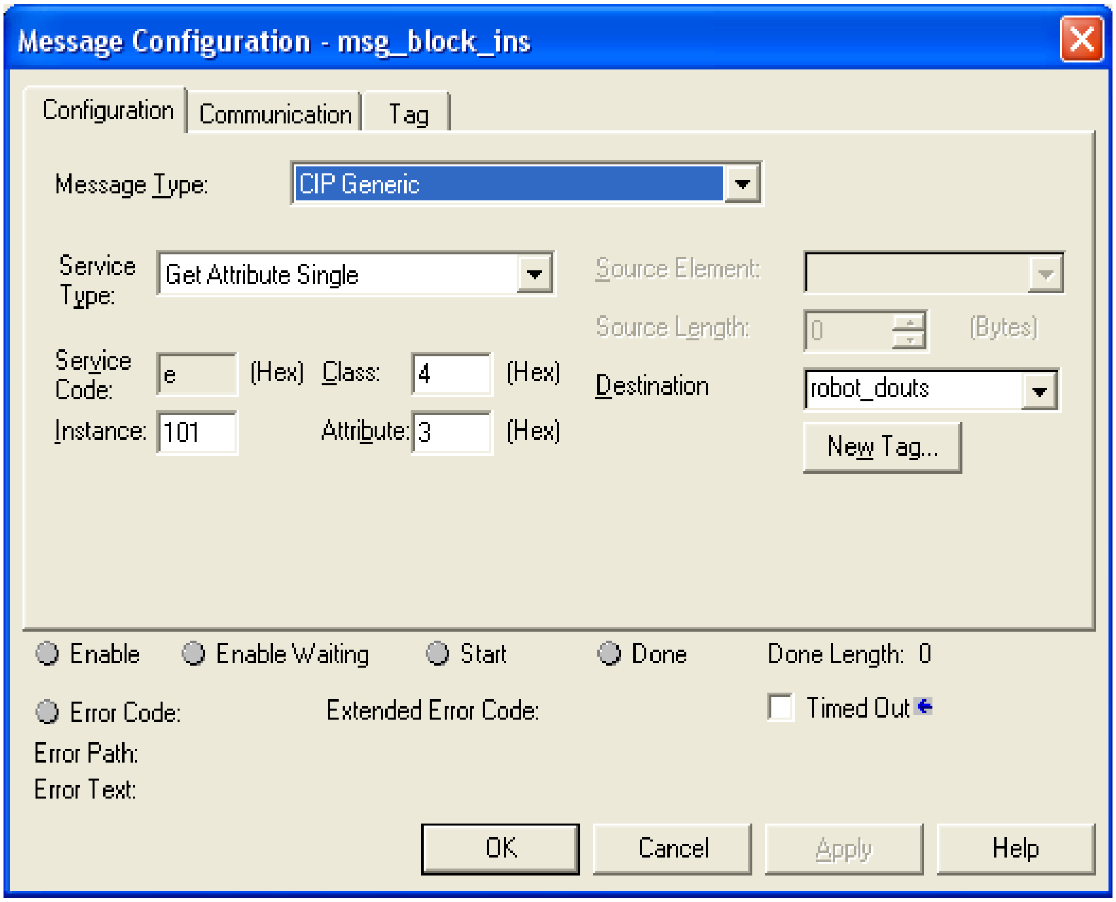
Figure 7–3. RSLogix 5000 Add MSG Block



Configuration of the message block requires the Class, Instance, Attribute, and Service values as discussed in Section 7.3 of this manual. For example, to read the robot controller outputs at Rack 89 Slot 1, we would access the Assembly Class (0x04), Attribute 3 (0x03), Instance 101 (0x65), and Service Get_Attribute_Single (0x0e). See Section 7.12 for more details.

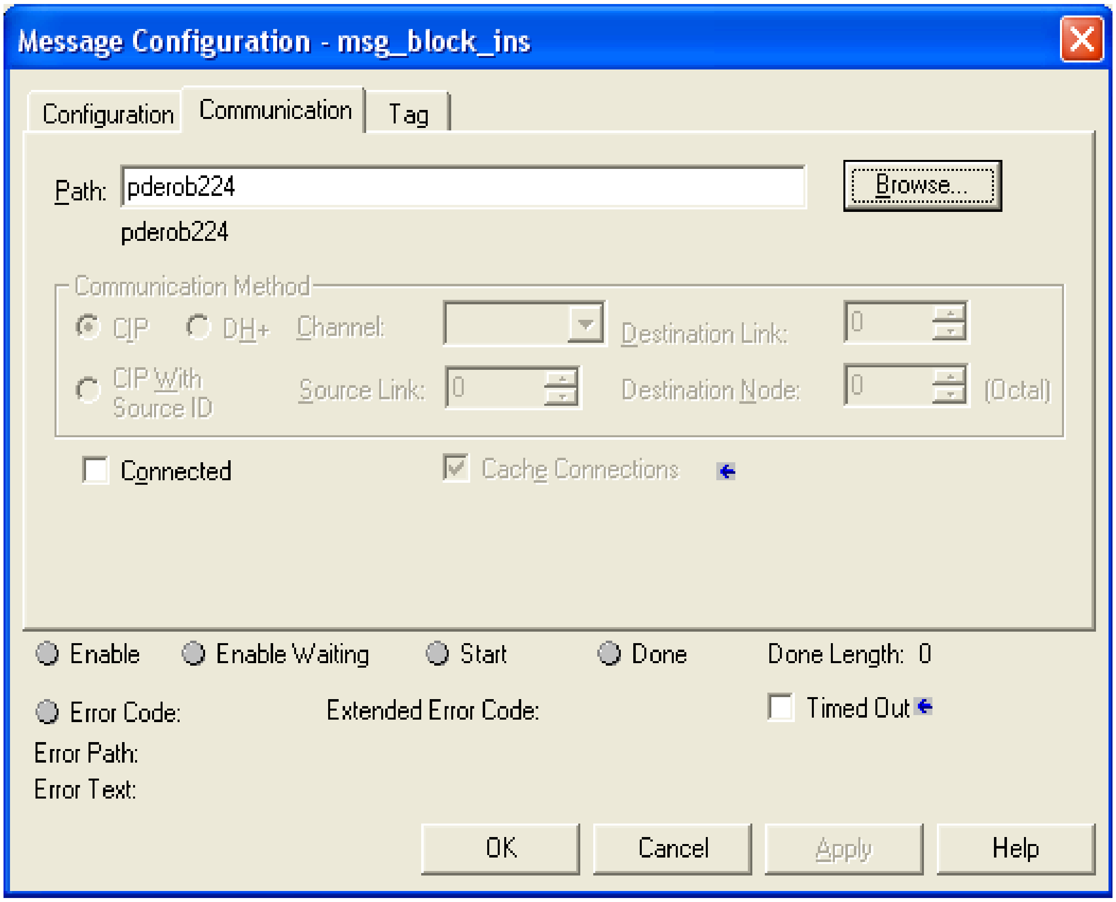
Thus, once configured, the MSG block should look similar to Figure 7–4 . You will need to create a destination for the robot controller outputs to be read into. In this example, we created an array named robot_douts.

Figure 7–4. MSG Block: Read Robot DOUTs



Next, click on the Communications tab. From this tab you should be able to browse to the device to which you want to connect. In Figure 7–5 , we browsed to and are connecting to a device named pderob224. Note that the robot controller must already be configured in RSLogix5000's I/O Configuration for the Browse button to successfully find the robot.

Figure 7–5. MSG Block Communication Tab

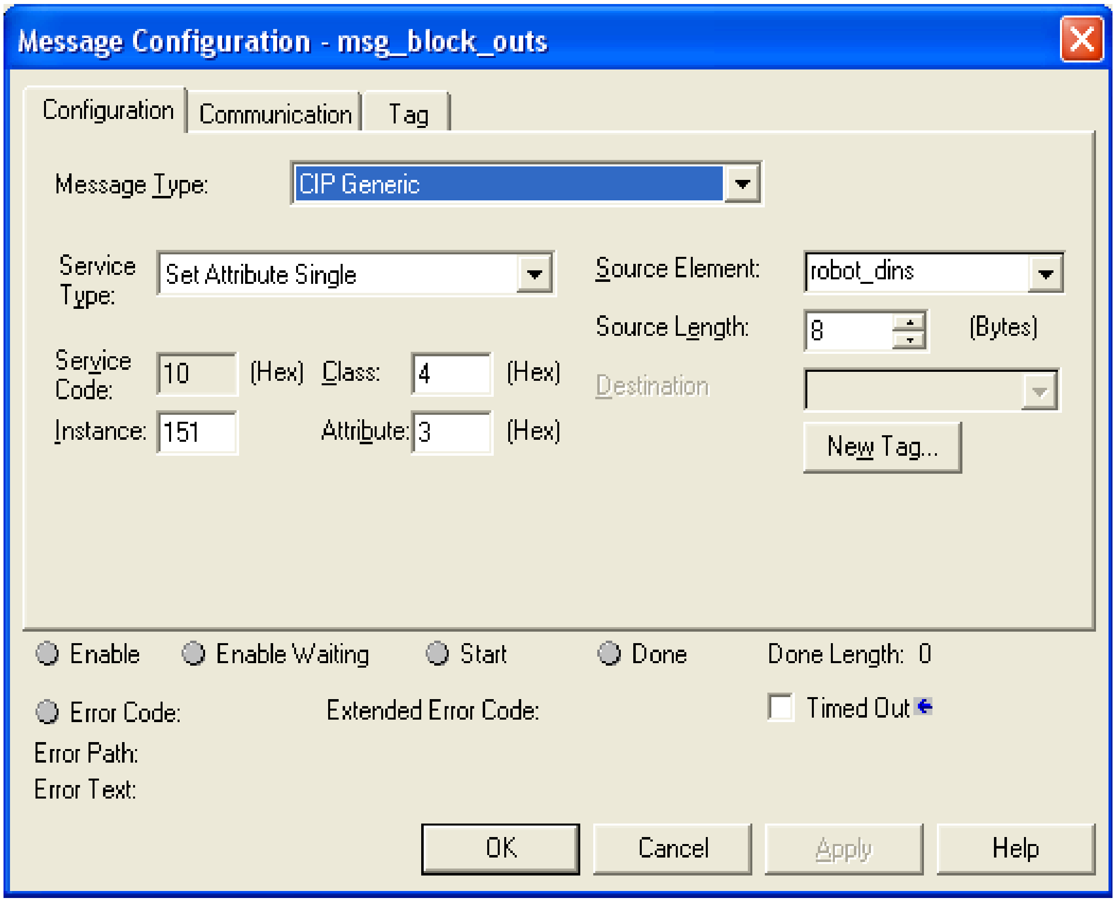


Documentation for RSLogix5000 also provides two additional ways of expressing the Path. First, the path can be an expression of comma-separated values that indicate the route for the MSG starting at the Ethernet module on the PLC and ending at the target device. For example “ENET,2,192.168.1.224” would be a path from the ENET module, port 2 on the ENET module (this value should always be 2), to the IP address of the robot controller.

Secondly, the same path could be expressed as “1,2,2,192.168.1.224”. Where the 1 represents the slot number of the processor in the rack, and the 2 in the second position represents the slot number of the ENET module. The 2 in the third position would represent port two on the ENET module, and the fourth position contains the IP address of the robot.

Figure 7–6 shows the MSG block used to write to the robot controller’s inputs. In this case we use Class 0x04, Instance 151 (0x97), Attribute 0x03, and Service Set_Attribute_Single (0x10). We also created the array robot_dins to write to the robot controller.

Figure 7–6. MSG Block: Write Robot DINs



Chapter 8

NETWORK DESIGN AND PERFORMANCE

Contents

8.1 GUIDELINES FOR USING ETHERNET/IP

Good network design is critical for reliable operation. It is important to pay special attention to wiring guidelines and environmental conditions affecting the cable system and equipment. It is also necessary to control network traffic to avoid wasted network bandwidth and device resources.

Keep in mind the following wiring guidelines and environmental considerations:

- Use category 5 twisted pair (or better) rated for 100-BaseTX Ethernet applications and the application environment. Consider shielded versus unshielded twisted pair cabling.
- Pay careful attention to wiring guidelines such as maximum length from the switch to the device (100 meters).
- Do not exceed recommended bending radius of specific cabling being used.
- Use connectors appropriate to the environment. There are various industrial Ethernet connectors in addition to the standard open RJ45 that should be used where applicable. For example, connectors are available with IP65 or IP67 ratings.
- Route the wire runs away from electrical or magnetic interference or cross at ninety degrees to minimize induced noise on the Ethernet network.

Keep the following in mind as you manage network traffic:

- Control or eliminate collisions by limiting the collision domain.
- Control broadcast traffic by limiting the broadcast domain.
- Control multicast traffic with intelligent routing.
- Use QOS (Quality of Service) techniques in very demanding applications.

Collisions are a traditional concern on an Ethernet network but can be completely avoided by using switches—rather than hubs—and full duplex connections. It is critical to use switches and full duplex connections for any Ethernet I/O network, because it reduces the collision domain to only one device so that no collisions will occur. The robot interface will autonegotiate by default and use the fastest connection possible. Normally this is 100Mbps and full duplex. The robot can be set for a specific connection speed and duplex (refer to the chapter titled "Setting Up TCP/IP" in the *Internet Options Setup and Operations Manual*). However be very careful that both ends of the connection use the same speed and duplex mode. The LEDs near the RJ45 connector on the robot will confirm connection status (refer to the Appendix titled "Diagnostic Information" in the *Internet Options Setup and Operations Manual* for details on the LEDs).

Broadcast traffic is traffic that all nodes on the subnet must listen for and in some cases respond to. Excessive broadcast traffic wastes network bandwidth and wastes resources in all effected nodes. The broadcast domain is the range of devices (typically the entire subnet) that must listen to all broadcasts. FANUC Robotics recommends limiting the broadcast domain to only the control devices (for example, EtherNet/IP nodes) by using a separate subnet for the control equipment or by using

VLANs (virtual LANs) supported by some higher end switches. If the EtherNet/IP network is completely isolated as a separate control network this is not a concern. However, when connecting into larger networks this becomes important.

Some network environments have a significant amount of multicast traffic. A basic layer 2 switch will treat multicast traffic like broadcast traffic and forward to all ports in the switch wasting network bandwidth and node resources on traffic which is ultimately dropped for the nodes that are not interested in the multicast traffic. Switches that support "IGMP snooping" will selectively send multicast traffic only to the nodes which have joined a particular group. EtherN et/IP UDP packet has a TTL (time to link) value of one. You will not be able to route I/O traffic across more than one switch.

Quality of Service (QOS) techniques provide mechanisms to prioritize network traffic. Generally on an Ethernet network all packets are equal. Packets can be dropped or delayed within network infrastructure equipment (for example, switches) in the presence of excessive traffic. Which packets are dropped or delayed is random.

QOS is a term covering several different approaches to prioritizing packets including:

- MAC layer (layer 2) prioritization (IEEE 802.1p).
- IP layer (layer 3) prioritization using source/destination IP addresses.
- Transport layer (layer 4) prioritization using source/destination ports.

These QOS mechanisms are generally implemented within the network infrastructure equipment and are beyond the scope of this manual. Some form of QOS should be considered on complex networks requiring the highest possible level of determinism in I/O exchanges within the control network.

It is important to select the proper switch in order for EtherNet/IP to function correctly. The switch should support :

- 100 Mbps baud rate
- Full duplex connections
- Port auto-negotiation
- Environmental specifications appropriate for the application (for example, temperature)
- Power supply requirements and redundancy (for example, support for 24vdc or 120vac and support for a second redundant power supply if warranted)

Note If there is a significant amount of multicast traffic, the switch should support IGMP snooping (multicast aware).

Note If the control network will be part of a larger network, the control network should be on a separate VLAN or subnet. This can be done within the control switch or possibly based on how the larger network connects to the control switch.

Some examples of switch products are:

- Telesyn AT-FS716, 16 port unmanaged switch – www.alliedtelesyn.com
- Cisco 2950-12, 12 port managed switch – www.cisco.com
- Cisco 2955 (industrialized version of 2950) – www.cisco.com
- Hischmann MICE (modular industrial switch) – www.hirschmann.de
- Phoenix Contact (managed/unmanaged industrial switch) – www.ethernetrail.com
- N-Tron 508TX-A, 8 port industrial switch with advanced firmware – www.n-tron.com

8.2 I/O RESPONSE TIME

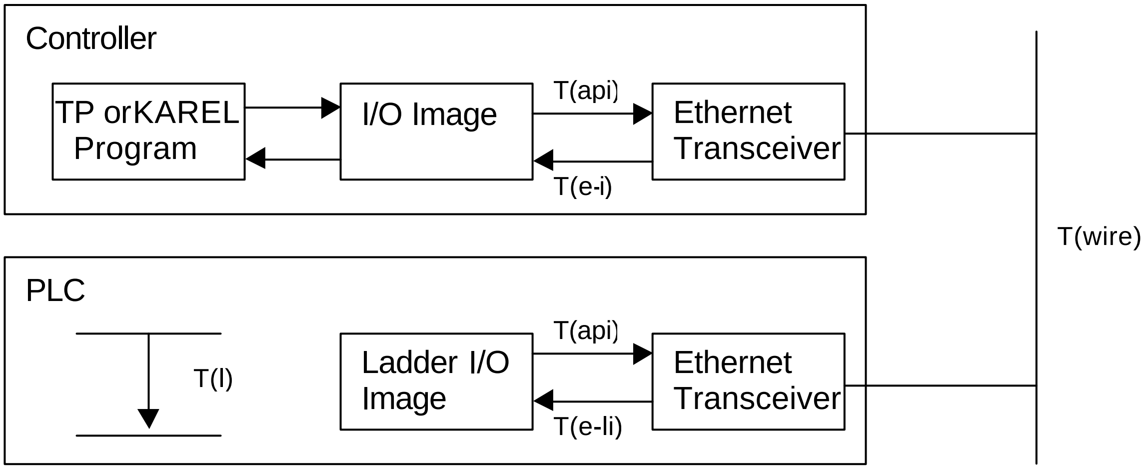
The system response time is the amount of time it takes an I/O signal to propagate through the system to its destination and back again. For a Controller -to- PLC system this time would be from the time an output is sent to the time a modified input is read. The system response time depends on many factors including:

- The Actual Packet Interval (API is based on RPI)
- PLC Ladder Scan Time
- No lost or delayed packets due to excessive traffic or noise

To calculate the response time, keep in mind that the response time is asynchronous but has a deterministic upper limit. After a signal is set in the I/O Image, it will take a maximum of one API before it gets transmitted to any node on the network.

Figure 8–1 shows a case where a DO is transferred to a PLC and back as a DI. In this case, after the DO is set in the I/O Image, it will take a maximum of one API, t(api), to get the DO to the Ethernet transceiver. After the DO is in the Ethernet transceiver, it is sent to the destination (PLC) at wire speed, t(wire) (assumes full duplex link so no collisions).

Figure 8–1. EtherNet/IP Response Time Diagram



In the PLC, the inputs are scanned into the Ladder I/O Image, which fixes them for the entire Ladder Scan, t(e-li). The inputs are processed during the ladder scan t(l) and then set back to the PLC's Ethernet transceiver at it's API rate.

The PLC outputs are transferred at Ethernet wire speed back to the controller. They are then transferred to the I/O Image t(e-i) where they can be read by the KAREL or teach pendant program.

```
T(Controller -to- PLC -to- Controller) = t(api) + t(wire) + t(e-li) + t(l) +
     t(api) + t(wire) + t(e-i)
```

- t(api) = KAREL or teach pendant outputs get immediately set in the I/O Image. The time necessary to get to them to Ethernet transceiver can be a maximum of one API. In this example it is assumed robot->plc API and plc->robot API values are the same.
- t(wire) = The time it takes for the packet to traverse the network including any switch delays due to queueing.
- t(e-l i) = After the signal is in the PLC Ethernet transceiver, it must get processed through the PLC network stack and placed in an appropriate ladder image data file to be accessed by the PLC Ladder.
- t(l) = The input value needs to be fixed for the entire scan in the ladder. The PLC Scan can usually be obtained by examining an appropriate status register in the PLC. After the signal has been processed, the reverse process must take place.
- T(e-i) = After the signal is in the robot Ethernet transceiver, it must get processed through the robot network stack and placed in I/O image area to be accessed by the TP or Karel program.

For example, using V7.10 or higher and a ControlLogix PLC over a simple network with a 20ms API, the following times were calculated. This example assumes wire time is negligible (100Mbps network, no switch delays), input packets are processed through the network stack and into image area within 1ms, and ladder scan time is 5ms.

```
T(Controller - to - PLC - to - Controller
) = t(api) + t(wire) + t(e-li) + t(l) + t(api) + t(wire) + t(e-i)


T(Controller - to - PLC - to - Controller) = 20ms + 0ms + 1ms + 5ms +
20ms + 0ms + 1ms


T(Controller - to - PLC - to - Controller) = 47ms ***
```

*** This value assumes no delayed/lost packets due to excessive traffic or noise.

Your actual PLC ladder scan times might vary from the example. Most PLCs offer the capability to get the actual scan time from a programmer or monitor.

This example assumes the packet is not delayed or dropped in a network switch or at the source/destination node. Packets can be dropped due to the following reasons:

- Excessive traffic can cause queue delays or dropped packets in the switch or source/destination nodes depending on extent of traffic and queue sizes.
- Packet corruption due to noise can cause a bad CRC check on the packet (a packet with a bad CRC is dropped).

The maximum upper limit is based on EtherNet/IP timeout values. Timeouts will occur when a consumer does not receive data from a producer within a multiple of the API. Typically this timeout value is 3-4 times the API value. If a timeout occurs, an error is posted. The error severity and last state I/O behavior can be configured. Refer to Section 3.2.3 for adapter.

Chapter 9

DIAGNOSTICS AND TROUBLESHOOTING

Contents

9.1 VERIFYING NETWORK CONNECTIONS

There are two basic tools for verifying network connections:

- Ethernet status LEDs
- PING

The LEDs and PING utility are basic tools but they give a good indication of whether or not devices are able to communicate on the network. If the LINK LED is off, or if PING times out, then no other network functionality will work for that device.

Refer to Section 9.1.1 for more information about Ethernet status LEDs.

Refer to Section 9.1.2 for more information about the PING utility.

9.1.1 Ethernet Status LEDs

The Ethernet status LEDs at the Ethernet RJ45 connector on the robot will indicate if the robot is connected. Most Ethernet switches and other equipment will have similar LEDs indicating a physical connection. If the LINK LED is off then there is no Ethernet connectivity at all. This generally implies a disconnected or bad cable or bad connections. The speed and duplex must match between the robot and the switch. For more information about the Ethernet status LEDs, refer to the appendix titled “Diagnostic Information” in the *Internet Options Setup and Operations Manual* . Details on auto-negotiating and manually setting speed and duplex level can be found in the chapter titled “ Setting Up TCP/IP” in the of the *Internet Options Setup and Operations Manual* . The robot will auto-negotiate by default and should not be changed in most cases.

9.1.2 PING Utility

PING is a network utility that sends a request to a specific IP address and expects a response. The request is essentially "Can you hear me?" The destination node will send a response that it received the request. The requesting node will either receive the response or timeout. PING is a basic network utility that is included with most operating systems, such as Windows and Unix, and is also supported on the robot. Even devices that do not support generating PING requests (for example, an EtherNet/IP block with no user interface) will respond to the PING request.

The robot supports PING directly from the EtherNet/IP status screen. Use Procedure 9-1 .

The PING utility is also available on the robot to PING any name or IP address. Use Procedure 9-2 .

The PING utility is also available from any windows PC. Use Procedure 9-3 .

Procedure 9-1 Using PING from the EtherNet/IP Status Screen

Steps

1. Press MENUS.
2. Select I/O.
3. Press F1, [TYPE].
4. Select EtherNet/IP.
5. Move the cursor to the connection with the device you want to PING.
6. Press F2, PING.

The prompt line on the teach pendant will indicate if the PING was successful or if the PING request timed out.

Note This function only works on the adapter connection (connection #1) if there is a scanner connected.

Procedure 9-2 Using PING on the Robot

Steps

1. Press MENUS.
2. Select Setup
3. Press F1, [TYPE].
4. Select Host Comm.
5. Move the cursor to select PING in the Protocol List and press ENTER.
6. Enter the name or IP address of the node to PING.
7. Press F2, PING.

The prompt line on the teach pendant will indicate if the PING was successful or if the PING request timed out.

Procedure 9-3 Using PING on a Windows PC

Steps

1. Open a DOS command prompt.
2. Type the following command, replacing the IP address with the IP address you want to PING, and press ENTER.

```
PING 192.168.0.10
```

The following image shows a successful PING.

```
C:\>ping 172.22.200.65

Pinging 172.22.200.65 with 32 bytes of data:

Reply from 172.22.200.65: bytes=32 time<1ms TTL=128
Reply from 172.22.200.65: bytes=32 time<1ms TTL=128
Reply from 172.22.200.65: bytes=32 time<1ms TTL=128
Reply from 172.22.200.65: bytes=32 time<1ms TTL=128

Ping statistics for 172.22.200.65:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```

The following image shows an unsuccessful PING.

```
C:\>ping 172.22.200.240

Pinging 172.22.200.240 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 172.22.200.240:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>
```

If the LINK LED is on but a PING request fails it usually indicates a problem with IP address configuration. Either no IP address is configured, or the combination of IP address and subnet mask is inconsistent for the network. Refer to the chapter titled “Setting Up TCP/IP” in the *Internet Options Setup and Operations Manual* for details on configuring the IP address and subnet mask for the robot.

9.2 ERROR CODES

The following error codes are defined by the EtherNet/IP January 2005 Specification. When a controller scanner connection fails when establishing a connection to a target device, the controller posts an error in the following format (PRIO-350 is the error code, and PRIO-358 is the cause code).

```
PRIO-350  EtherNet/IP Scanner Error (#)
PRIO-358  EtherNet/IP Fwd Open Fail (0x#)
```

The PRIO-350 code (#) specifies on which connection the error has occurred. The PRIO-358 code (0x#) specifies the **extended status** of the error returned by the target device (in hexadecimal format).

Table 9–1 lists the descriptions of the extended status error codes:

Note EtherNet/IP alarms are documented in the *Error Code Manual* .

Table 9–1. Forward Open Failure Error codes

GENERAL STATUS	EXTENDED STATUS	DESCRIPTION
0x01	0x0100	Connection in Use or Duplicate Forward Open
0x01	0x0103	Transport Class and Trigger combination not supported
0x01	0x0106	Ownership Conflict
0x01	0x0107	Connection not found at target application.
0x01	0x0108	Invalid Connection Type. Indicates a problem with either the Connection Type or Priority of the Connection
0x01	0x0109	Invalid Connection Size
0x01	0x0110	Device not configured
0x01	0x0111	RPI not supported. Might also indicate problem with connection timeout multiplier or production inhibit time.
0x01	0x0113	Connection Manager cannot support any more connections
0x01	0x0114	Either the Vendor Id or the Product Code in the key segment did not match the device
0x01	0x0115	Product Type in the key segment did not match the device
0x01	0x0116	Major or Minor Revision information in the key segment did not match the device
0x01	0x0117	Invalid Connection Point
0x01	0x0118	Invalid Configuration Format
0x01	0x0119	Connection request fails since there is no controlling connection currently open.
0x01	0x011A	Target Application cannot support any more connections
0x01	0x011B	RPI is smaller than the Production Inhibit Time.
0x01	0x0203	Connection cannot be closed since the connection has timed out
0x01	0x0204	Unconnected Send timed out waiting for a response.
0x01	0x0205	Parameter Error in Unconnected Send Service
0x01	0x0206	Message too large for Unconnected message service
0x01	0x0207	Unconnected acknowledge without reply
0x01	0x0301	No buffer memory available

Table 9–1. Forward Open Failure Error codes (Cont'd)

0x01	0x0302	Network Bandwidth not available for data
0x01	0x0303	No screeners available
0x01	0x0304	Not Configured to send real-time data
0x01	0x0311	Port specified in Port Segment Not Available
0x01	0x0312	Link Address specified in Port Segment Not Available
0x01	0x0315	Invalid Segment Type or Segment Value in Path
0x01	0x0316	Error in close path
0x01	0x0317	Scheduling not specified
0x01	0x0318	Link Address to Self Invalid
0x01	0x0319	Resources on Secondary Unavailable
0x01	0x031A	Connection already established
0x01	0x031B	Direct connection already established
0x01	0x031C	Miscellaneous
0x01	0x031D	Redundant connection mismatch
0x01	0x031E	No more consumer resources available in the producing module
0x01	0x031F	No connection resources exist for target path
0x01	0x320 — 0x7FF	Vendor specific

Appendix A

THIRD-PARTY CONFIGURATION TOOLS

Contents

A.1 Tools Overview

Robot Scanner connections can be configured from the EtherNet/IP Interface screens, or from third party tools such as RSNetWorx for EtherNet/IP using the Connection Configuration Object (CCO). Certain devices require detailed configuration data, and can only be added to the robot scanlist by using offline tools such as an Allen Bradley Flex I/O block with attached modules. Other devices can be configured through both interfaces, such as another FANUC Robot, or an RJ-Lynx I/O block. To use the offline tools, an EDS file for each device is required.

It is recommended that either all scanlist configurations be done entirely from the *i*Pendant, or be done entirely from RSNetWorx for EtherNet/IP. In certain situations RSNetWorx for EtherNet/IP version 4.11 might delete scanlist entries configured through the *i*Pendant.

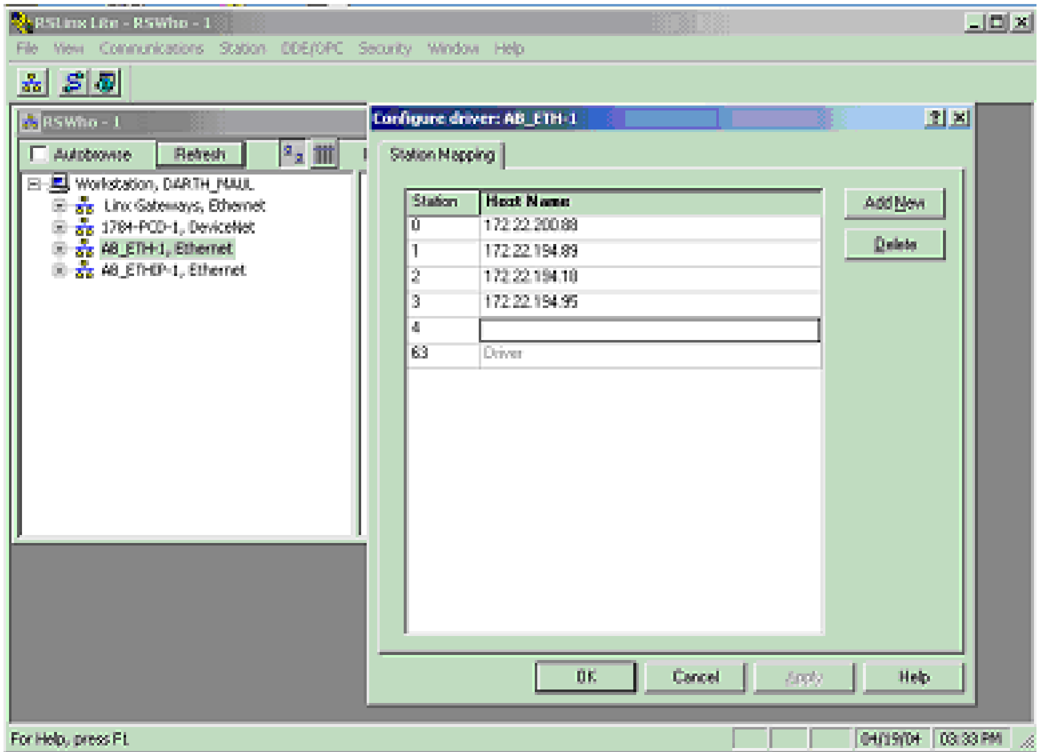
Note TIP: Some third party tools cannot import scanlist configurations from the robot controller unless both the configured revision numbers (major and minor) exactly match the revision numbers in the EDS file that the third party tool had loaded for the corresponding device. To set the revision numbers, see Section 4.2.4 .

Procedure A-1 Configure the Robot Scanners Using RSNetworX for EtherNet/IP

Steps

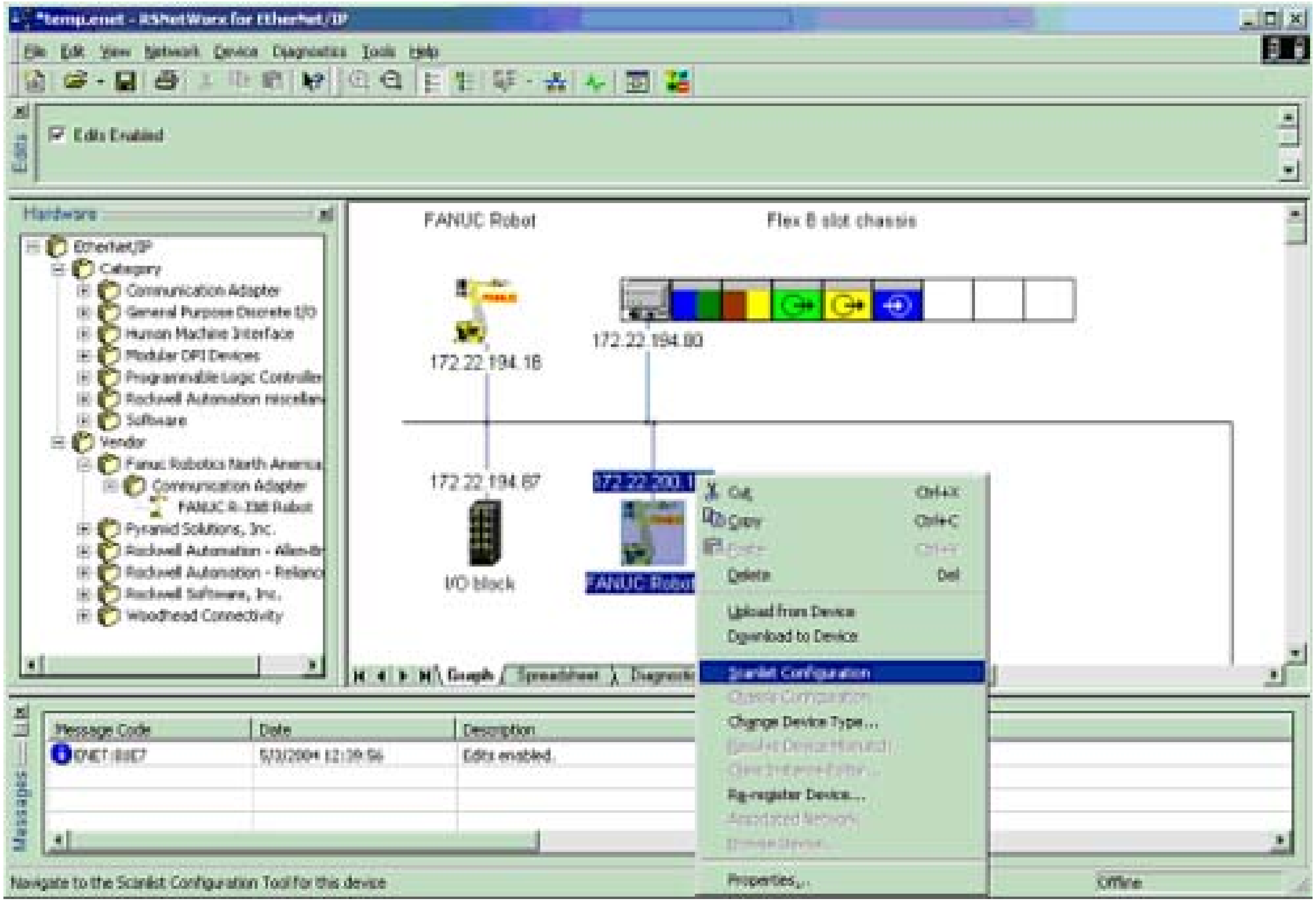
1. From the EtherNet/IP Interface Status screen, create and configure the desired number of scanner connections. See Section 4.2.3 .
2. Perform a Controlled start. Refer to the application-specific *FANUC Robotics Setup and Operations Manual* .
3. Configure an AB_ETH driver in RsLinx. The configuration screen should be similar to the screen shown in Figure A–1 .

Figure A–1. Configuring the Driver



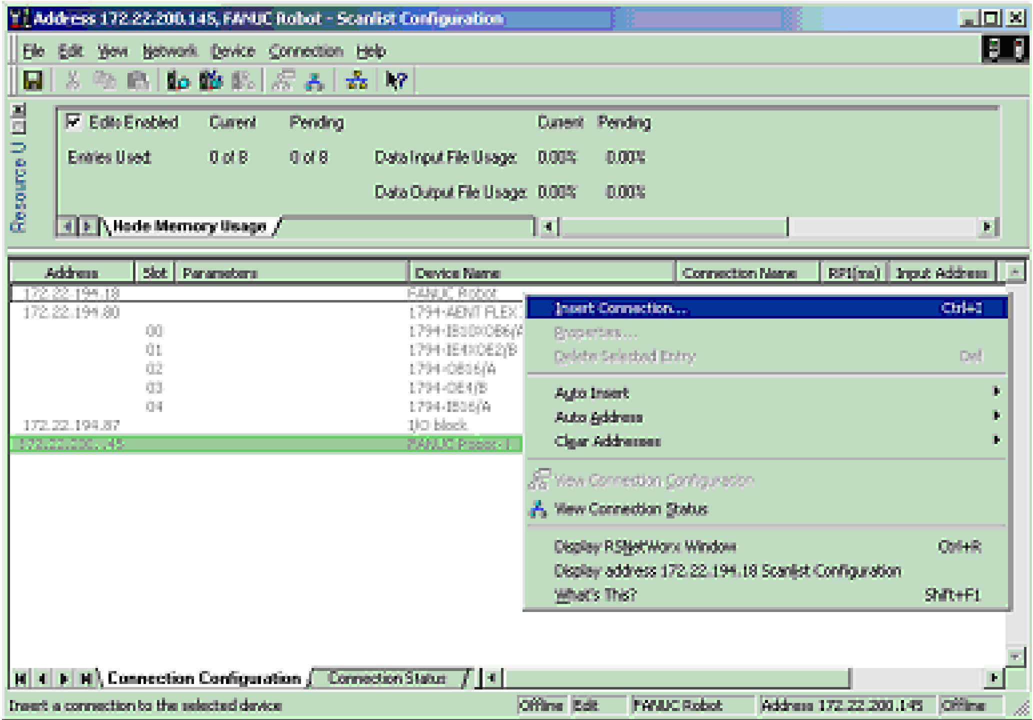
4. In RSNetWorx, under Tools, select the EDS Wizard and register the FANUC Robot EDS file.
5. Create an EtherNet/IP project in RSNetWorx that includes the devices configured in RSLinx.
6. Right click on the robot icon, and select Scanlist Configuration. You will see a screen similar to the one shown in Figure A–2

Figure A–2. Scanlist Configuration Screen



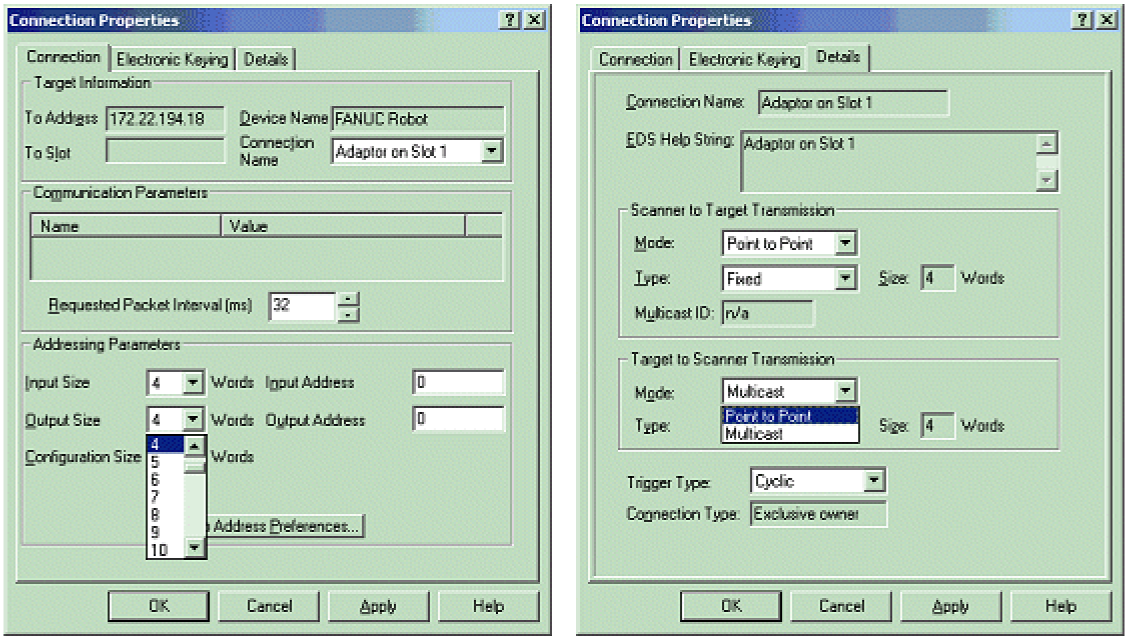
7. A new window appears entitle “FANUC Robot – Scanlist Configuration”. In this configuration window, right click over the device you want to add to the robot’s scanlist, and select Insert Connection. See Figure A–3

Figure A–3. Insert Connection Screen



8. Configure the connection properties and click on OK. Pay specific attention to the selecting the appropriate Connection Name, Input Size, and Output Size. Other configurable values might include RPI, and Target to Scanner Transmission Mode. Note that the robot does not use values inserted for Input Address or Output Address. The screens below show a FANUC Robot being added to the scanlist of another FANUC Robot.

Figure A–4. Adding a Robot



9. In the Scanlist Configuration window, select Device/Download to Device.

10. When finished, Cold start the controller.

Appendix B

KAREL PROGRAMS FOR ETHERNET/IP

Contents

B.1 OVERVIEW

The EtherNet/IP Scanner option installs the following KAREL programs:

- EN_OFFLN Allows a teach pendant program to turn an EtherNet/IP scanner connection off
- EN_ONLN Allows a teach pendant program to turn an EtherNet/IP scanner connection on
- EN_AROFF - Allows a teach pendant program to turn off auto-reconnect for an EtherNet/IP scanner connection.
- EN_ARON - Allows a teach pendant program to turn on auto-reconnect for an EtherNet/IP scanner connection.
- EN_STCHK - Allows a teach pendant program to check the status of an EtherNet/IP scanner connection.

B.2 KAREL PROGRAM DESCRIPTIONS AND PARAMETERS

The following are the KAREL program descriptions and parameters.

EN_OFFLN (INTEGER slot_number)

This program allows a teach pendant program to turn an EtherNet/IP scanner connection offline. This program takes the slot number as an argument. The valid values for a slot number are 1 through 32. For example, 1 corresponds to the EtherNet/IP Connection on slot 1, rack 89, or the first connection displayed on the EtherNet/IP Status Screen. There is no difference between this call and disabling the connection from the teach pendant.

EN_ONLN (INTEGER slot_number, INTEGER <wait_time>)

This program allows a teach pendant program to turn an EtherNet/IP scanner connection online. This program takes the slot number as an argument. The valid values for a slot number are 1 through 32. For example, 1 corresponds to the EtherNet/IP Connection on slot 1, rack 89, or the first connection displayed on the EtherNet/IP Status Screen. The optional argument, wait_time, is used as follows:

- **If wait_time is not used.** If wait_time is not explicitly specified (it is an optional argument), its value will be defaulted to 15 and EN_ONLN follows the: if wait_time is not 0 rule.
- **If wait_time is not 0.** The EtherNet/IP scanner connection will be enabled. Auto-reconnect will also be enabled, causing the scanner to attempt to make a connection to the adapter device every 2 seconds until successful. Note that EN_ONLN will block and will not return until a successful connection is made, or until the user aborts the teach pendant program. An alarm will be posted if wait_time seconds pass before a connection is established. After the alarm is posted and the robot faults, a reset/resume from either the PLC or teach pendant will restart/resume the program inside of the EN_ONLN call and the wait_time timer will be reset. Before EN_ONLN returns, auto-reconnect will set to its original state (its state before EN_ONLN was called).
- **If wait_time is used and set to 0.** Auto-reconnect will not be enabled--the user must explicitly enable Auto-reconnect if needed. The EtherNet/IP connection will be enabled and the call will

return immediately (will not block). There is no difference between this call and enabling the connection from the teach pendant. The application or user programs can then use EN_STCHK to check the status if it needs to confirm the status of the connection.

EN_AROFF (INTEGER slot_number)

This program allows a teach pendant program to turn off auto-reconnect for an EtherNet/IP scanner connection. This program takes the slot number as an argument. The valid values for a slot number are 1 through 32. For example, 1 corresponds to the EtherNet/IP Connection on slot 1, rack 89, or the first connection displayed on the EtherNet/IP Status Screen.

There is no difference between this call and disabling auto-reconnect from the teach pendant.

EN_ARON (INTEGER slot_ number)

This program allows a teach pendant program to turn on auto-reconnect for an EtherNet/IP scanner connection. This program takes the slot number as an argument. The valid values for a slot number are 1 through 32. For example, 1 corresponds to the EtherNet/IP Connection on slot 1, rack 89, or the first connection displayed on the EtherNet/IP Status Screen.

Enabling auto-reconnect has the following side effects. While enabled, all EtherNet/IP alarms relating to connection establishment and connection time- outs for this slot number will be masked (will not be posted). The EtherNet/IP scanner corresponding to the slot number will attempt to make a connection to the adapter device every 2 seconds until successful. Before each retry, the ARP cache in the TCP/IP stack will be flushed of the target IP address. Also, the status on the teach pendant will become encapsulated in < and > as in <STATUS>, for example.

There is no difference between this call and enabling auto-reconnect from the teach pendant.

EN_STCHK (INTEGER slot_number, INTEGER register_number)

This program allows a teach pendant program to check the status of an EtherNet/IP connection. This program takes the slot number, and register_number as arguments. The valid values for a slot number are 1 through 32. For example, 1 corresponds to the EtherNet/IP Connection on slot 1, rack 89, or the first connection displayed on the EtherNet/IP Status Screen. The possible status values returned in the register_number are:

- 0 Offline
- 1 Error
- 2 Pending
- 3 Enabled but not connected or trying to connection
- 4 Enabled but not connected. Is trying to connect.
- 5 Online and connected but I/O is not being received from adapter
- 6 Online and I/O is being exchanged

Note When a connection is taken offline, if a background application were to access I/O belonging to that connection, an unassigned port alarm would be posted. This should be taken into consideration by the teach pendant programmer when using the EN_OFFLN program.

B.3 USING KAREL PROGRAMS IN TEACH PENDANT PROGRAMS

Procedure B-1 shows how to use the EN_STCHK KAREL program. The other programs listed in this section can be used in the same way. Procedure E.1 Placing the Call to the KAREL Program in the Teach Pendant Program

Procedure B-1 Placing the Call to the KAREL Program in the Teach Pendant Program

1. Press SELECT.
2. Display the appropriate list of programs. If F1, [TYPE], is not displayed on the screen, press >, NEXT, until it is displayed.
   a. Press F1, [TYPE].
   b. Select the list you want:
3. Move the cursor to the name of the program you want to modify and press ENTER.
4. Turn the teach pendant ON/OFF switch to ON.
5. Select F4, [INST].
6. Select Call from the list of options that appear at the top of the screen.
7. Select Call Program and press ENTER.
8. Press F3, [KAREL] to display the available KAREL programs at the top of the screen.
9. Select EN_STCHK and press ENTER.
10. Place the cursor to the right of the word EN_STCHK.
11. Press F4, [CHOICE].
12. Select Constant from the list at the top of the screen and press ENTER.
13. Type the Slot Number and press ENTER.
14. Press F4, [CHOICE].
15. Select Constant from the list at the top of the screen and press ENTER.
16. Type the Register Number for the result of the device status check and press ENTER.

    The finished line in the teach pendant program should look like the following:

    ```
    CALL EN_STCHK (2,50)
    ```

    Note 50 is the register number for result (R[50]).

B.4 EXAMPLES USING ETHERNET/IP MACROS

B.4.1 Overview

Generally, EtherNet/IP macros are used to support tool change applications. The following examples demonstrate using Auto-Reconnect and connection Offline/Online macros in tool changing applications.

The connection offline/online macros are similar to manually taking the connection offline or online in the teach pendant screens but are called programmatically through a teach pendant program.

A single EtherNet/IP scanner connection can be used to connect to different types of I/O blocks (different electronic keying) that may be used on the various tools if the I/O sizes for these blocks are the same. In these cases, the electronic keying parameters must be set to 0 in the corresponding scanner configuration screen.

Turning on Auto-Reconnect means that the robot will automatically try to reconnect to the target device if the connection is lost. Without Auto-Reconnect enabled, the robot will fault if an EtherNet/IP scanner connection is lost, and reset must be pressed to retry the connection. With auto-reconnect enabled, the robot will not fault and will continuously try to reconnect to the device. Auto- reconnect should be turned off if a tool change is not underway (when you do not expect the connection to be lost) so that unexpected connection problems are not masked.

B.4.2 Individual Examples

The example below turns on Auto-Reconnect for the second connection (the connection corresponding to EtherNet/IP slot 2). This call can be executed just before the tool is to be physically disconnected to prevent the robot from faulting once the disconnection occurs. Alternatively, call EN_OFFLN to disable the EtherNet/IP scanner connection before the tool is to be physically disconnected, and execute this call when the tool is physically reconnected, but before EN_ONLN is called.

```
1:  CALL EN_ARON(2) ;
```

The next example enables, or brings online, the second connection (the connection corresponding to EtherNet/IP slot 2). This call should be executed when the tool is physically reconnected.

```
1:  CALL EN_ONLN(2) ;
```

The example below checks the status of the second connection (the connection corresponding to EtherNet/IP slot 2). The status is placed in register 5, R[5]. The connection is not established and exchanging I/O until the status is equal to the value 6.

```
1:  CALL EN_STCHK(2,5) ;
```

The next example turns off Auto-Reconnect for the second connection (the connection corresponding to EtherNet/IP slot 2). This call should be executed after an EtherNet/IP scanner connection has been established. Any problem with this EtherNet/IP connection at this point would be a valid error and will now fault the robot.

```
1:  CALL EN_AROFF(2) ;
```

The example below disables, or takes offline, the second connection (the connection corresponding to EtherNet/IP slot 2). This call may be executed just before the tool is to be physically disconnected.

```
1:  CALL EN_OFFLN(2) ;
```

B.4.3 Advanced Examples

It can take many seconds for an EtherNet/IP adapter device to power-up and become ready to exchange I/O. The following example can be done after moving away from the tool changer nest to help cycle time by allowing the device power- up and connection time to be done in parallel with the robot motion. The following logic will check for the device to go online for up to 15 seconds. If the device status becomes online in less than 15 seconds, the robot will resume immediately after the online status is obtained. If the device is still not online after 15 seconds a User Alarm is posted and the robot will fault. Note that in line #3, the option argument wait_time is set to 0 for EN_ONLN. Care must be used when setting a device online just after it is reconnected. If it is not fully powered up and available for reconnection by the scanner, an alarm might be generated. To avoid this problem, auto-reconnect is generally enabled while setting the device online, and then disabled once the device comes online. The following logic assumes auto-reconnect has already been enabled.

```
 1:  TIMER[1]=RESET ;
 2:  TIMER[1]=START ;
 3:  CALL EN_ONLN(2,0)
 4:  LBL[1] ;
 5:  CALL EN_STCHK(2,50) ;
 6:  IF R[50]=6,JMP LBL[3] ;
 7:  WAIT    .10(sec) ;
 8:  IF (TIMER[1]<15),JMP LBL[1] ;
 9:  UALM[1] ;
10:  JMP LBL[1] ;
11:  LBL[3] ;
12:  TIMER[1]=STOP ;
13:  CALL EN_AROFF(2) ;
```

The same functionality of the above logic can also be achieved by setting the optional parameter wait_time of the EN_ONLN program to a non-zero value as seen in the logic below. When the wait_time parameter is not set, it will default to 15. In this case, EN_ONLN does not return until an EtherNet/IP scanner connection has been established.

```
1:  CALL EN_ONLN(2,15)
2:  CALL EN_AROFF(2) ;
```

OR

```
1:  CALL EN_ONLN(2)
2:  CALL EN_AROFF(2) ;
```

Below is an outline of how a tool change may occur and a recommended sequence of calls to programmatically handle the tool change.

* Tool is connected and exchanging I/O with EtherNet/IP scanner connection 2.

* A tool change is scheduled to occur.

```
CALL EN_OFLN(2) ;
```

* Physically disconnect the tool.

* Physically connect a new tool.

```
CALL EN_ARON(2) ;
CALL EN_ONLN(2) ;
```

* When EN_ONLN returns, tool is connected and exchanging I/O with EtherNet/IP scanner connection 2.

```
CALL EN_AROFF(2) ;
```

Glossary

A

abort

Abnormal termination of a computer program caused by hardware or software malfunction or operator cancellation.

absolute pulse code system

A positional information system for servomotors that relies on battery-backed RAM to store encoder pulse counts when the robot is turned off. This system is calibrated when it is turned on.

A/D value

An analog to digital-value. Converts a multilevel analog electrical system pattern into a digital bit.

AI

Analog input.

AO

Analog output.

alarm

The difference in value between actual response and desired response in the performance of a controlled machine, system or process. Alarm=Error.

algorithm

A fixed step-by-step procedure for accomplishing a given result.

alphanumeric

Data that are both alphabetical and numeric.

AMPS

Amperage amount.

analog

The representation of numerical quantities by measurable quantities such as length, voltage or resistance. Also refers to analog type I/O blocks and distinguishes them from discrete I/O blocks. Numerical data that can vary continuously, for example, voltage levels that can vary within the range of -10 to +10 volts.

AND

An operation that places two contacts or groups of contacts in series. All contacts in series control the resulting status and also mathematical operator.

ANSI

American National Standard Institute, the U.S. government organization with responsibility for the development and announcement of technical data standards.

APC

See absolute pulse code system.

APC motor

See servomotor.

application program

The set of instructions that defines the specific intended tasks of robots and robot systems to make them reprogrammable and multifunctional. You can initiate and change these programs.

arm

A robot component consisting of an interconnecting set of links and powered joints that move and support the wrist socket and end effector.

articulated arm

A robot arm constructed to simulate the human arm, consisting of a series of rotary motions and joints, each powered by a motor.

ASCII

Abbreviation for American Standard Code for Information Interchange. An 8-level code (7 bits plus 1 parity bit) commonly used for the exchange of data.

automatic mode

The robot state in which automatic operation can be initiated.

automatic operation

The time during which robots are performing programmed tasks through unattended program execution.

axis

1. A straight line about which a robot joint rotates or moves. 2. One of the reference lines or a coordinate system. 3. A single joint on the robot arm.

B

backplane

A group of connectors mounted at the back of a controller rack to which printed circuit boards are mated.

BAR

A unit of pressure equal to 100,000 pascals.

barrier

A means of physically separating persons from the restricted work envelope; any physical boundary to a hazard or electrical device/component.

battery low alarm

A programmable value (in engineering units) against which the analog input signal automatically is compared on Genius I/O blocks. A fault is indicated if the input value is equal to or less than the low alarm value.

baud

A unit of transmission speed equal to the number of code elements (bits) per second.

big-endian

The adjectives big-endian and little-endian refer to which bytes are most significant in multi-byte data types and describe the order in which a sequence of bytes is stored in a computer's memory. In a big-endian system, the most significant value in the sequence is stored at the lowest storage address (i.e., first). In a little-endian system, the least significant value in the sequence is stored first.

binary

A numbering system that uses only 0 and 1.

bit

Contraction of binary digit. 1. The smallest unit of information in the binary numbering system, represented by a 0 or 1. 2. The smallest division of a programmable controller word.

bps

Bits per second.

buffer

A storage area in the computer where data is held temporarily until the computer can process it.

bus

A channel along which data can be sent.

bus controller
A Genius bus interface board for a programmable controller.

bus scan
One complete communications cycle on the serial bus.

Bus Switching Module
A device that switches a block cluster to one bus or the other of a dual bus.

byte
A sequence of binary digits that can be used to store a value from 0 to 255 and usually operated upon as a unit. Consists of eight bits used to store two numeric or one alpha character.

C

calibration
The process whereby the joint angle of each axis is calculated from a known reference point.

Cartesian coordinate system
A coordinate system whose axes (x, y, and z) are three intersecting perpendicular straight lines. The origin is the intersection of the axes.

Cartesian coordinates
A set of three numbers that defines the location of a point within a rectilinear coordinate system and consisting of three perpendicular axes (x, y, z).

cathode ray tube
A device, like a television set, for displaying information.

central processing unit
The main computer component that is made up of a control section and an arithmetic-logic section. The other basic units of a computer system are input/output units and primary storage.

channel
The device along which data flow between the input/output units of a computer and primary storage.

character
One of a set of elements that can be arranged in ordered groups to express information. Each character has two forms: 1. a man-intelligible form, the graphic, including the decimal digits 0-9, the letters A-Z, punctuation marks, and other formatting and control symbols; 2. a computer intelligible form, the code, consisting of a group of binary digits (bits).

circular
A MOTYPE option in which the robot tool center point moves in an arc defined by three points. These points can be positions or path nodes.

clear
To replace information in a storage unit by zero (or blank, in some machines).

closed loop
A control system that uses feedback. An open loop control system does not use feedback.

C-MOS RAM
Complementary metal-oxide semiconductor random-access memory. A read/write memory in which the basic memory cell is a pair of MOS (metal-oxide semiconductor) transistors. It is an implementation of S-RAM that has very low power consumption, but might be less dense than other S-RAM implementations.

coaxial cable
A transmission line in which one conductor is centered inside and insulated from an outer metal tube that serves as the second conductor. Also known as coax, coaxial line, coaxial transmission line, concentric cable, concentric line, concentric transmission line.

component
An inclusive term used to identify a raw material, ingredient, part or subassembly that goes into a higher level of assembly, compound or other item.

computer
A device capable of accepting information, applying prescribed processes to the information, and supplying the results of these processes.

configuration
The joint positions of a robot and turn number of wrist that describe the robot at a specified position. Configuration is designated by a STRING value and is included in positional data.

continuous path
A trajectory control system that enables the robot arm to move at a constant tip velocity through a series of predefined locations. A rounding effect of the path is required as the tip tries to pass through these locations.

controller memory
A medium in which data are retained. Primary storage refers to the internal area where the data and program instructions are stored for active use, as opposed to auxiliary or external storage (magnetic tape, disk, diskette, and so forth.)

continuous process control
The use of transducers (sensors) to monitor a process and make automatic changes in operations through the design of appropriate feedback control loops. While such devices historically have been mechanical or electromechanical, microcomputers and centralized control is now used, as well.

continuous production
A production system in which the productive equipment is organized and sequenced according to the steps involved to produce the product. Denotes that material flow is continuous during the production process. The routing of the jobs is fixed and set-ups are seldom changed.

controlled stop

A controlled stop controls robot deceleration until it stops. When a safety stop input such as a safety fence signal is opened, the robot decelerates in a controlled manner and then stops. After the robot stops, the Motor Control Contactor opens and drive power is removed.

controller

A hardware unit that contains the power supply, operator controls, control circuitry, and memory that directs the operation and motion of the robot and communications with external devices. See control unit.

control, open-loop

An operation where the computer applies control directly to the process without manual intervention.

control unit

The portion of a computer that directs the automatic operation of the computer, interprets computer instructions, and initiates the proper signals to the other computer circuits to execute instructions.

coordinate system

See Cartesian coordinate system.

CPU

See central processing unit.

CRT

See cathode ray tube.

cps (viscosity)

Centipoises per second.

CRT/KB

Cathode ray tube/keyboard. An optional interface device for the robot system. The CRT/KB is used for some robot operations and for entering programs. It can be a remote device that attaches to the robot via a cable.

cycle

1. A sequence of operations that is repeated regularly. The time it takes for one such sequence to occur. 2. The interval of time during which a system or process, such as seasonal demand or a manufacturing operation, periodically returns to similar initial conditions. 3. The interval of time during which an event or set of events is completed. In production control, a cycle is the length of time between the release of a manufacturing order and shipment to the customer or inventory.

cycle time

1. In industrial engineering, the time between completion of two discrete units of production. 2. In materials management, the length of time from when material enters a production facility until it exits. See throughput.

cursor

An indicator on a teach pendant or CRT display screen at which command entry or editing occurs. The indicator can be a highlighted field or an arrow (> or ^).

cylindrical

Type of work envelope that has two linear major axes and one rotational major axis. Robotic device that has a predominantly cylindrical work envelope due to its design. Typically has fewer than 6 joints and typically has only 1 linear axis.

D

D/A converter

A digital-to-analog converter. A device that transforms digital data into analog data.

D/A value

A digital-to-analog value. Converts a digital bit pattern into a multilevel analog electrical system.

daisy chain

A means of connecting devices (readers, printers, etc.) to a central processor by party-line input/output buses that join these devices by male and female connectors. The last female connector is shorted by a suitable line termination.

daisy chain configuration

A communications link formed by daisy chain connection of twisted pair wire.

data

A collection of facts, numeric and alphabetical characters, or any representation of information that is suitable for communication and processing.

data base

A data file philosophy designed to establish the independence of computer program from data files. Redundancy is minimized and data elements can be added to, or deleted from, the file designs without changing the existing computer programs.

DC

Abbreviation for direct current.

DEADMAN switch

A control switch on the teach pendant that is used to enable servo power. Pressing the DEADMAN switch while the teach pendant is on activates servo power and releases the robot brakes; releasing the switch deactivates servo power and applies the robot brakes.

debugging

The process of detecting, locating and removing mistakes from a computer program, or manufacturing control system. See diagnostic routine.

deceleration tolerance
The specification of the percentage of deceleration that must be completed before a motion is considered finished and another motion can begin.

default
The value, display, function or program automatically selected if you have not specified a choice.

deviation
Usually, the absolute difference between a number and the mean of a set of numbers, or between a forecast value and the actual data.

device
Any type of control hardware, such as an emergency-stop button, selector switch, control pendant, relay, solenoid valve, or sensor.

diagnostic routine
A test program used to detect and identify hardware/software malfunctions in the controller and its associated I/O equipment. See debugging.

diagnostics
Information that permits the identification and evaluation of robot and peripheral device conditions.

digital
A description of any data that is expressed in numerical format. Also, having the states On and Off only.

digital control
The use of a digital computer to perform processing and control tasks in a manner that is more accurate and less expensive than an analog control system.

digital signal
A single point control signal sent to or from the controller. The signal represents one of two states: ON (TRUE, 1. or OFF (FALSE, 0).

directory
A listing of the files stored on a device.

discrete
Consisting of individual, distinct entities such as bits, characters, circuits, or circuit components. Also refers to ON/OFF type I/O blocks.

disk
A secondary memory device in which information is stored on a magnetically sensitive, rotating disk.

disk memory
A non-programmable, bulk-storage, random-access memory consisting of a magnetized coating on one or both sides of a rotating thin circular plate.

drive power

The energy source or sources for the robot servomotors that produce motion.

DRAM

Dynamic Random Access Memory. A read/write memory in which the basic memory cell is a capacitor. DRAM (or D-RAM) tends to have a higher density than SRAM (or S-RAM). Due to the support circuitry required, and power consumption needs, it is generally impractical to use. A battery can be used to retain the content upon loss of power.

E

edit

1. A software mode that allows creation or alteration of a program. 2. To modify the form or format of data, for example, to insert or delete characters.

emergency stop

The operation of a circuit using hardware-based components that overrides all other robot controls, removes drive power from the actuators, and causes all moving parts of to stop. The operator panel and teach pendant are each equipped with EMERGENCY STOP buttons.

enabling device

A manually operated device that, when continuously activated, permits motion. Releasing the device stops the motion of the robot and associated equipment that might present a hazard.

encoder

1. A device within the robot that sends the controller information about where the robot is. 2. A transducer used to convert position data into electrical signals. The robot system uses an incremental optical encoder to provide position feedback for each joint. Velocity data is computed from the encoder signals and used as an additional feedback signal to assure servo stability.

end effector

An accessory device or tool specifically designed for attachment to the robot wrist or tool mounting plate to enable the robot to perform its intended tasks. Examples include gripper, spot weld gun, arc weld gun, spray paint gun, etc.

end-of-arm tooling

Any of a number of tools, such as welding guns, torches, bells, paint spraying devices, attached to the faceplate of the robot wrist. Also called end effector or EOAT.

engineering units

Units of measure as applied to a process variable, for example, psi, Degrees F., etc.

envelope, maximum

The volume of space encompassing the maximum designed movements of all robot parts including the end effector, workpiece, and attachments.

EOAT

See end of arm tooling, tool.

EPROM

Erasable Programmable Read Only Memory. Semiconductor memory that can be erased and reprogrammed. A non-volatile storage memory.

error

The difference in value between actual response and desired response in the performance of a controlled machine, system or process. Alarm=Error.

error message

A numbered message, displayed on the CRT/KB and teach pendant, that indicates a system problem or warns of a potential problem.

Ethernet

A Local Area Network (LAN) bus-oriented, hardware technology that is used to connect computers, printers, terminal concentrators (servers), and many other devices together. It consists of a master cable and connection devices at each machine on the cable that allow the various devices to "talk" to each other. Software that can access the Ethernet and cooperate with machines connected to the cable is necessary. Ethernets come in varieties such as baseband and broadband and can run on different media, such as coax, twisted pair and fiber. Ethernet is a trademark of Xerox Corporation.

execute

To perform a specific operation, such as one that would be accomplished through processing one statement or command, a series of statements or commands, or a complete program or command procedure.

extended axis

An optional, servo-controlled axis that provides extended reach capability for a robot, including in-booth rail, single- or double-link arm, also used to control motion of positioning devices.

F

faceplate

The tool mounting plate of the robot.

feedback

1. The signal or data fed back to a commanding unit from a controlled machine or process to denote its response to the command signal. The signal representing the difference between actual response and desired response that is used by the commanding unit to improve performance of the controlled machine or process. 2. The flow of information back into the control system so that actual performance can be compared with planned performance, for instance in a servo system.

field

A specified area of a record used for a particular category of data. 2. A group of related items that occupy the same space on a CRT/KB screen or teach pendant LCD screen. Field name is the name of the field; field items are the members of the group.

field devices

User-supplied devices that provide information to the PLC (inputs: push buttons, limit switches, relay contacts, an so forth) or perform PLC tasks (outputs: motor starters, solenoids, indicator lights, and so forth.)

file

1. An organized collection of records that can be stored or retrieved by name. 2. The storage device on which these records are kept, such as bubble memory or disk.

filter

A device to suppress interference that would appear as noise.

Flash File Storage

A portion of FROM memory that functions as a separate storage device. Any file can be stored on the FROM disk.

Flash ROM

Flash Read Only Memory. Flash ROM is not battery-backed memory but it is non-volatile. All data in Flash ROM is saved even after you turn off and turn on the robot.

flow chart

A systems analysis tool to graphically show a procedure in which symbols are used to represent operations, data, flow, and equipment. See block diagram, process chart.

flow control

A specific production control system that is based primarily on setting production rates and feeding work into production to meet the planned rates, then following it through production to make sure that it is moving. This concept is most successful in repetitive production.

format

To set up or prepare a memory card or floppy disk (not supported with version 7.20 and later) so it can be used to store data in a specific system.

FR

See Flash ROM.

F-ROM

See Flash ROM.

FROM disk

See Flash ROM.

G

general override stat

A percentage value that governs the maximum robot jog speed and program run speed.

Genius I/O bus

The serial bus that provides communications between blocks, controllers, and other devices in the system especially with respect to GE FANUC Genius I/O.

gripper

The "hand" of a robot that picks up, holds and releases the part or object being handled. Sometimes referred to as a manipulator. See EOAT, tool.

group signal

An input/output signal that has a variable number of digital signals, recognized and taken as a group.

gun

See applicator.

H

Hand Model.

Used in Interference Checking, the Hand Model is the set of virtual model elements (spheres and cylinders) that are used to represent the location and shape of the end of arm tooling with respect to the robot's faceplate.

hardware

1. In data processing, the mechanical, magnetic, electrical and electronic devices of which a computer, controller, robot, or panel is built. 2. In manufacturing, relatively standard items such as nuts, bolts, washers, clips, and so forth.

hard-wire

To connect electric components with solid metallic wires.

hard-wired

1. Having a fixed wired program or control system built in by the manufacturer and not subject to change by programming. 2. Interconnection of electrical and electronic devices directly through physical wiring.

hazardous motion

Unintended or unexpected robot motion that can cause injury.

hexadecimal

A numbering system having 16 as the base and represented by the digits 0 through 9, and A through F.

hold

A smoothly decelerated stopping of all robot movement and a pause of program execution. Power is maintained on the robot and program execution generally can be resumed from a hold.

HTML.

Hypertext Markup Language. A markup language that is used to create hypertext and hypermedia documents incorporating text, graphics, sound, video, and hyperlinks.

http.

Hypertext transfer protocol. The protocol used to transfer HTML files between web servers.

I

impedance

A measure of the total opposition to current flow in an electrical circuit.

incremental encoder system

A positional information system for servomotors that requires calibrating the robot by moving it to a known reference position (indicated by limit switches) each time the robot is turned on or calibration is lost due to an error condition.

index

An integer used to specify the location of information within a table or program.

index register

A memory device containing an index.

industrial robot

A reprogrammable multifunctional manipulator designed to move material, parts, tools, or specialized devices through variable programmed motions in order to perform a variety of tasks.

industrial robot system

A system that includes industrial robots, end effectors, any equipment devices and sensors required for the robot to perform its tasks, as well as communication interfaces for interlocking, sequencing, or monitoring the robot.

information

The meaning derived from data that have been arranged and displayed in a way that they relate to that which is already known. See data.

initialize

1. Setting all variable areas of a computer program or routine to their desired initial status, generally done the first time the code is executed during each run. 2. A program or hardware circuit that returns a program a system, or hardware device to an original state. See startup, initial.

input

The data supplied from an external device to a computer for processing. The device used to accomplish this transfer of data.

input device

A device such as a terminal keyboard that, through mechanical or electrical action, converts data from the form in which it has been received into electronic signals that can be interpreted by the CPU or programmable controller. Examples are limit switches, push buttons, pressure switches, digital encoders, and analog devices.

input processing time

The time required for input data to reach the microprocessor.

input/output

Information or signals transferred between devices, discreet electrical signals for external control.

input/output control

A technique for controlling capacity where the actual output from a work center is compared with the planned output developed by CRP. The input is also monitored to see if it corresponds with plans so that work centers will not be expected to generate output when jobs are not available to work on.

integrated circuit

A solid-state micro-circuit contained entirely within a chip of semiconductor material, generally silicon. Also called chip.

interactive

Refers to applications where you communicate with a computer program via a terminal by entering data and receiving responses from the computer.

interface

1. A concept that involves the specifications of the inter-connection between two equipments having different functions. 2. To connects a PLC with the application device, communications channel, and peripherals through various modules and cables. 3. The method or equipment used to communicate between devices.

interference zone

An area that falls within the work envelope of a robot, in which there is the potential for the robot motion to coincide with the motion of another robot or machine, and for a collision to occur.

interlock

An arrangement whereby the operation of one control or mechanism brings about, or prevents, the operations of another.

interrupt

A break in the normal flow of a system or program that occurs in a way that the flow can be resumed from that point at a later time. Interrupts are initiated by two types of signals: 1. signals originating within the computer system to synchronize the operation of the computer system with the outside world; 2. signals originating exterior to the computer system to synchronize the operation of the computer system with the outside world.

I/O

Abbreviation for input/output or input/output control.

I/O block

A microprocessor-based, configurable, rugged solid state device to which field I/O devices are attached.

I/O electrical isolation

A method of separating field wiring from logic level circuitry. This is typically done through optical isolation devices.

I/O module

A printed circuit assembly that is the interface between user devices and the Series Six PLC.

I/O scan

A method by which the CPU monitors all inputs and controls all outputs within a prescribed time. A period during which each device on the bus is given a turn to send information and listen to all of the broadcast data on the bus.

ISO

The International Standards Organization that establishes the ISO interface standards.

isolation

1. The ability of a logic circuit having more than one inputs to ensure that each input signal is not affected by any of the others. 2. A method of separating field wiring circuitry from logic level circuitry, typically done optically.

item

1. A category displayed on the teach pendant on a menu. 2. A set of adjacent digits, bits, or characters that is treated as a unit and conveys a single unit of information. 3. Any unique manufactured or purchased part or assembly: end product, assembly, subassembly, component, or raw material.

J

jog coordinate systems

Coordinate systems that help you to move the robot more effectively for a specific application. These systems include JOINT, WORLD, TOOL, and USER.

JOG FRAME

A jog coordinate system you define to make the robot jog the best way possible for a specific application. This can be different from world coordinate frame.

jogging

Pressing special keys on the teach pendant to move the robot.

jog speed
Is a percentage of the maximum speed at which you can jog the robot.

joint
1. A single axis of rotation. There are up to six joints in a robot arm (P-155 swing arm has 8). 2. A jog coordinate system in which one axis is moved at a time.

JOINT
A motion type in which the robot moves the appropriate combination of axes independently to reach a point most efficiently. (Point to point, non-linear motion).

joint interpolated motion
A method of coordinating the movement of the joints so all joints arrive at the desired location at the same time. This method of servo control produces a predictable path regardless of speed and results in the fastest cycle time for a particular move. Also called joint motion.

K

K
Abbreviation for kilo, or exactly 1024 in computer jargon. Related to 1024 words of memory.

KAREL
The programming language developed for robots by the FANUC Robotics America, Inc.

L

label
An ordered set of characters used to symbolically identify an instruction, a program, a quantity, or a data area.

LCD
See liquid crystal display.

lead time
The span of time needed to perform an activity. In the production and inventory control context, this activity is normally the procurement of materials and/or products either from an outside supplier or from one's own manufacturing facility. Components of lead time can include order preparation time, queue time, move or transportation time, receiving and inspection time.

LED
See Light Emitting Diode.

LED display
An alphanumeric display that consists of an array of LEDs.

Light Emitting Diode
A solid-state device that lights to indicate a signal on electronic equipment.

limiting device

A device that restricts the work envelope by stopping or causing to stop all robot motion and that is independent of the control program and the application programs.

limit switch

A switch that is actuated by some part or motion of a machine or equipment to alter the electrical circuit associated with it. It can be used for position detection.

linear

A motion type in which the appropriate combination of axes move in order to move the robot TCP in a straight line while maintaining tool center point orientation.

liquid crystal display

A digital display on the teach pendant that consists of two sheets of glass separated by a sealed-in, normally transparent, liquid crystal material. Abbreviated LCD.

little-endian

The adjectives big-endian and little-endian refer to which bytes are most significant in multi-byte data types and describe the order in which a sequence of bytes is stored in a computer’s memory. In a big-endian system, the most significant value in the sequence is stored at the lowest storage address (i.e., first). In a little-endian system, the least significant value in the sequence is stored first.

load

1. The weight (force) applied to the end of the robot arm. 2. A device intentionally placed in a circuit or connected to a machine or apparatus to absorb power and convert it into the desired useful form. 3. To copy programs or data into memory storage.

location

1. A storage position in memory uniquely specified by an address. 2. The coordinates of an object used in describing its x, y, and z position in a Cartesian coordinate system.

lockout/tagout

The placement of a lock and/or tag on the energy isolating device (power disconnecting device) in the off or open position. This indicates that the energy isolating device or the equipment being controlled will not be operated until the lock/tag is removed.

log

A record of values and/or action for a given function.

logic

A fixed set of responses (outputs) to various external conditions (inputs). Also referred to as the program.

loop

The repeated execution of a series of instructions for a fixed number of times, or until interrupted by the operator.

M

mA
See milliampere.

machine language
A language written in a series of bits that are understandable by, and therefore instruct, a computer. This is a "first level" computer language, as compared to a "second level" assembly language, or a "third level" compiler language.

machine lock
A test run option that allows the operator to run a program without having the robot move.

macro
A source language instruction from which many machine-language instructions can be generated.

magnetic disk
A metal or plastic floppy disk (not supported on version 7.10 and later) that looks like a phonograph record whose surface can store data in the form of magnetized spots.

magnetic disk storage
A storage device or system consisting of magnetically coated metal disks.

magnetic tape
Plastic tape, like that used in tape recorder, on which data is stored in the form of magnetized spots.

maintenance
Keeping the robots and system in their proper operating condition.

MC
See memory card.

mechanical unit
The robot arm, including auxiliary axis, and hood/deck and door openers.

medium
plural **media** . The physical substance upon which data is recorded, such as a memory card (or floppy disk which is not supported on version 7.10 and later).

memory
A device or media used to store information in a form that can be retrieved and is understood by the computer or controller hardware. Memory on the controller includes C-MOS RAM, Flash ROM and D-RAM.

memory card
A C-MOS RAM memory card or a flash disk-based PC card.

menu

A list of options displayed on the teach pendant screen.

message

A group of words, variable in length, transporting an item of information.

microprocessor

A single integrated circuit that contains the arithmetic, logic, register, control and memory elements of a computer.

microsecond

One millionth (0.000001) of a second

milliampere

One one-thousandth of an ampere. Abbreviated mA.

millisecond

One thousandth of a second. Abbreviated msec.

module

A distinct and identifiable unit of computer program for such purposes as compiling, loading, and linkage editing. It is eventually combined with other units to form a complete program.

motion type

A feature that allows you to select how you want the robot to move from one point to the next. MOTYPES include joint, linear, and circular.

mode

1. One of several alternative conditions or methods of operation of a device. 2. The most common or frequent value in a group of values.

N

network

1. The interconnection of a number of devices by data communication facilities. "Local networking" is the communications network internal to a robot."Global networking" is the ability to provide communications connections outside of the robot's internal system. 2. Connection of geographically separated computers and/or terminals over communications lines. The control of transmission is managed by a standard protocol.

non-volatile memory

Memory capable of retaining its stored information when power is turned off.

O

Obstacle Model.

Used in Interference Checking, the Obstacle Model is the set of virtual model elements (spheres, cylinders, and planes) that are used to represent the shape and the location of a given obstacle in space.

off-line

Equipment or devices that are not directly connected to a communications line.

off-line operations

Data processing operations that are handled outside of the regular computer program. For example, the computer might generate a report off-line while the computer was doing another job.

off-line programming

The development of programs on a computer system that is independent of the "on-board" control of the robot. The resulting programs can be copied into the robot controller memory.

offset

The count value output from a A/D converter resulting from a zero input analog voltage. Used to correct subsequent non-zero measurements also incremental position or frame adjustment value.

on-line

A term to describe equipment or devices that are connected to the communications line.

on-line processing

A data processing approach where transactions are entered into the computer directly, as they occur.

operating system

Lowest level system monitor program.

operating work envelope

The portion of the restricted work envelope that is actually used by the robot while it is performing its programmed motion. This includes the maximum the end-effector, the workpiece, and the robot itself.

operator

A person designated to start, monitor, and stop the intended productive operation of a robot or robot system.

operator box

A control panel that is separate from the robot and is designed as part of the robot system. It consists of the buttons, switches, and indicator lights needed to operate the system.

operator panel

A control panel designed as part of the robot system and consisting of the buttons, switches, and indicator lights needed to operate the system.

optional features

Additional capabilities available at a cost above the base price.

OR

An operation that places two contacts or groups of contacts in parallel. Any of the contacts can control the resultant status, also a mathematical operation.

orientation

The attitude of an object in space. Commonly described by three angles: rotation about x (w), rotation about y (p), and rotation about z (r).

origin

The point in a Cartesian coordinate system where axes intersect; the reference point that defines the location of a frame.

OT

See overtravel.

output

Information that is transferred from the CPU for control of external devices or processes.

output device

A device, such as starter motors, solenoids, that receive data from the programmable controller.

output module

An I/O module that converts logic levels within the CPU to a usable output signal for controlling a machine or process .

outputs

Signals, typically on or off, that controls external devices based upon commands from the CPU.

override

See general override.

overtravel

A condition that occurs when the motion of a robot axis exceeds its prescribed limits.

overwrite

To replace the contents of one file with the contents of another file when copying.

P

parity

The anticipated state, odd or even, of a set of binary digits.

parity bit

A binary digit added to an array of bits to make the sum of all bits always odd or always even.

parity check

A check that tests whether the number of ones (or zeros) in an array of binary digits is odd or even.

parity error
A condition that occurs when a computed parity check does not agree with the parity bit.

part
A material item that is used as a component and is not an assembly or subassembly.

pascal
A unit of pressure in the meter-kilogram-second system equivalent to one newton per square meter.

path
1. A variable type available in the KAREL system that consists of a list of positions. Each node includes positional information and associated data. 2. The trajectory followed by the TCP in a move.

PCB
See printed circuit board.

PC Interface
The PC Interface software uses Ethernet connections to provide file transfer protocol (FTP) functions, PC send macros, telnet interface, TCP/IP interface web server functions, and host communications.

pendant
See teach pendant.

PLC
See programmable logic controller or cell controller.

PMC
The programmable machine controller (PMC) functions provide a ladder logic programming environment to create PMC functions. This provides the capability to use the robot I/O system to run PLC programs in the background of normal robot operations. This function can be used to control bulk supply systems, fixed automation that is part of the robot workcell, or other devices that would normally require basic PLC controls.

printed circuit board
A flat board whose front contains slots for integrated circuit chips and connections for a variety of electronic components, and whose back is printed with electrically conductive pathways between the components.

production mode
See automatic mode.

program
1. A plan for the solution of a problem. A complete program includes plans for the transcription of data, coding for the computer, and plans for the absorption of the results into the system. 2. A sequence of instructions to be executed by the computer or controller to control a robot/robot system. 3. To furnish a computer with a code of instructions. 4. To teach a robot system a specific set of movements and instructions to do a task.

programmable controller
See programmable logic controller or cell controller.

programmable logic controller
A solid-state industrial control device that receives inputs from user-supplied control devices, such as switches and sensors, implements them in a precise pattern determined by ladder diagram-based programs stored in the user memory, and provides outputs for control of processes or user-supplied devices such as relays and motor starters.

Program ToolBox
The Program ToolBox software provides programming utilities such as mirror image and flip wrist editing capabilities.

protocol
A set of hardware and software interfaces in a terminal or computer that allows it to transmit over a communications network, and that collectively forms a communications language.

psi
Pounds per square inch.

Q

queue.
1. Waiting lines resulting from temporary delays in providing service. 2. The amount of time a job waits at a work center before set-up or work is performed on the job. See also job queue.

R

RAM
See Random Access Memory.

random access
A term that describes files that do not have to be searched sequentially to find a particular record but can be addressed directly.

Random Access Memory
1. Volatile, solid-state memory used for storage of programs and locations; battery backup is required. 2. The working memory of the controller. Programs and variable data must be loaded into RAM before the program can execute or the data can be accessed by the program.

range
1. A characterization of a variable or function. All the values that a function can possess. 2. In statistics, the spread in a series of observations. 3. A programmable voltage or current spectrum of values to which input or output analog signals can be limited.

RI
Robot input.

RO

Robot output.

read

To copy, usually from one form of storage to another, particularly from external or secondary storage to internal storage. To sense the meaning of arrangements of hardware. To sense the presence of information on a recording medium.

Read Only Memory

A digital memory containing a fixed pattern of bits that you cannot alter.

record

To store the current set or sets of information on a storage device.

recovery

The restoration of normal processing after a hardware or software malfunction through detailed procedures for file backup, file restoration, and transaction logging.

register

1. A special section of primary storage in a computer where data is held while it is being worked on.
2. A memory device capable of containing one or more computer bits or words.

remote/local

A device connection to a given computer, with remote devices being attached over communications lines and local devices attached directly to a computer channel; in a network, the computer can be a remote device to the CPU controlling the network.

repair

To restore robots and robot systems to operating condition after damage, malfunction, or wear.

repeatability

The closeness of agreement among the number of consecutive movements made by the robot arm to a specific point.

reset

To return a register or storage location to zero or to a specified initial condition.

restricted work envelope

That portion of the work envelope to which a robot is restricted by limiting devices that establish limits that will not be exceeded in the event of any reasonably foreseeable failure of the robot or its controls. The maximum distance the robot can travel after the limited device is actuated defines the restricted work envelope of the robot.

RIA

Robotic Industries Association Subcommittee of the American National Standards Institute, Inc.

robot

A reprogrammable multifunctional manipulator designed to move material, parts, tools, or specialized devices, through variable programmed motions for the performance of a variety of tasks.

Robot Model.

Used in Interference Checking, the Robot Model is the set of virtual model elements (sphere and cylinders) that are used to represent the location and shape of the robot arm with respect to the robot's base. Generally, the structure of a six axes robot can be accurately modeled as a series of cylinders and spheres. Each model element represents a link or part of the robot arm.

ROM

See Read Only Memory.

routine

1. A list of coded instructions in a program. 2. A series of computer instructions that performs a specific task and can be executed as often as needed during program execution.

S

saving data.

Storing program data in Flash ROM, to a floppy disk (not supported on version 7.10 and later), or memory card.

scfm

Standard cubic feet per minute.

scratch start

Allows you to enable and disable the automatic recovery function.

sensor

A device that responds to physical stimuli, such as heat, light, sound pressure, magnetism, or motion, and transmits the resulting signal or data for providing a measurement, operating a control or both. Also a device that is used to measure or adjust differences in voltage in order to control sophisticated machinery dynamically.

serial communication

A method of data transfer within a PLC whereby the bits are handled sequentially rather than simultaneously as in parallel transmission.

serial interface

A method of data transmission that permits transmitting a single bit at a time through a single line. Used where high speed input is not necessary.

Server Side Include (SSI)

A method of calling or "including" code into a web page.

servomotor

An electric motor that is controlled to produce precision motion. Also called a "smart" motor.

SI

System input.

signal

The event, phenomenon, or electrical quantity that conveys information from one point to another.

significant bit

A bit that contributes to the precision of a number. These are counted starting with the bit that contributes the most value, of "most significant bit", and ending with the bit that contributes the least value, or "least significant bit".

slip sheet

A sheet of material placed between certain layers of a unit load. Also known as tier sheet.

SO

System output.

specific gravity

The ratio of a mass of solid or liquid to the mass of an equal volume of water at 45C. You must know the specific gravity of the dispensing material to perform volume signal calibration. The specific gravity of a dispensing material is listed on the MSDS for that material.

SRAM

A read/write memory in which the basic memory cell is a transistor. SRAM (or S-RAM) tends to have a lower density than DRAM. A battery can be used to retain the content upon loss of power.

slpm

Standard liters per minute.

Standard Operator Panel (SOP).

A panel that is made up of buttons, keyswitches, and connector ports.

state

The on or off condition of current to and from and input or output device.

statement

See instruction.

storage device

Any device that can accept, retain, and read back one or more times. The available storage devices are SRAM, Flash ROM (FROM or F-ROM), floppy disks (not available on version 7.10 and later), memory cards, or a USB memory stick.

system variable

An element that stores data used by the controller to indicate such things as robot specifications, application requirements, and the current status of the system.

T

Tare

The difference between the gross weight of an object and its contents, and the object itself. The weight of an object without its contents.

TCP

See tool center point.

teaching

Generating and storing a series of positional data points effected by moving the robot arm through a path of intended motions.

teach mode

1. The mode of operation in which a robot is instructed in its motions, usually by guiding it through these motions using a teach pendant. 2. The generation and storage of positional data. Positional data can be taught using the teach pendant to move the robot through a series of positions and recording those positions for use by an application program.

teach pendant

1. A hand-held device used to instruct a robot, specifying the character and types of motions it is to undertake. Also known as teach box, teach gun. 2. A portable device, consisting of an LCD display and a keypad, that serves as a user interface to the KAREL system and attaches to the operator box or operator panel via a cable. The teach pendant is used for robot operations such as jogging the robot, teaching and recording positions, and testing and debugging programs.

telemetry

The method of transmission of measurements made by an instrument or a sensor to a remote location.

termination type

Feature that controls the blending of robot motion between segments.

tool

A term used loosely to define something mounted on the end of the robot arm, for example, a hand, gripper, or an arc welding torch.

tool center point

1. The location on the end-effector or tool of a robot hand whose position and orientation define the coordinates of the controlled object. 2. Reference point for position control, that is, the point on the tool that is used to teach positions. Abbreviated TCP.

TOOL Frame

The Cartesian coordinate system that has the position of the TCP as its origin to stet. The z-axis of the tool frame indicates the approach vector for the tool.

TP.

See teach pendant.

transducer

A device for converting energy from one form to another.

U

UOP

See user operator panel.

URL

Universal Resource Locator. A standard addressing scheme used to locate or reference files on web servers.

USB memory stick

The controller USB memory stick interface supports a USB 1.1 interface. The USB Organization specifies standards for USB 1.1 and 2.0. Most memory stick devices conform to the USB 2.0 specification for operation and electrical standards. USB 2.0 devices as defined by the USB Specification must be backward compatible with USB 1.1 devices. However, FANUC Robotics does not support any security or encryption features on USB memory sticks. The controller supports most widely-available USB Flash memory sticks from 32MB up to 1GB in size.

USER Frame

The Cartesian coordinate system that you can define for a specific application. The default value of the User Frame is the World Frame. All positional data is recorded relative to User Frame.

User Operator Panel

User-supplied control device used in place of or in parallel with the operator panel or operator box supplied with the controller. Abbreviated UOP .

V

variable

A quantity that can assume any of a given set of values.

variance

The difference between the expected (or planned) and the actual, also statistics definitions.

vision system

A device that collects data and forms an image that can be interpreted by a robot computer to determine the position or to “see” an object.

volatile memory
Memory that will lose the information stored in it if power is removed from the memory circuit device.

W

web server
An application that allows you to access files on the robot using a standard web browser.

warning device
An audible or visible device used to alert personnel to potential safety hazards.

work envelope
The volume of space that encloses the maximum designed reach of the robot manipulator including the end effector, the workpiece, and the robot itself. The work envelope can be reduced or restricted by limiting devices. The maximum distance the robot can travel after the limit device is actuated is considered the basis for defining the restricted work envelope.

write
To deliver data to a medium such as storage.

Index

P

R

S