FANUC Robot series

R-30*i*A and R-30*i*B Controller

HMI User's Manual

Applies to software version 7.70 and later.

This manual is to be used in conjunction with the
documentation provided by the HMI device manufacturer

MARUCHMID06121E REV A

FANUC Robotics America Corporation
3900 W. Hamlin Road
Rochester Hills, Michigan 48309–3253

Copyrights and Trademarks

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

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

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

WARNING

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

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

FANUC Robotics America Corporation
Training Department
3900 W. Hamlin Road
Rochester Hills, Michigan 48309-3253
www.fanucrobotics.com

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

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

Patents

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

FANUC Robotics America Corporation Patent List

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

FANUC CORPORATION Patent List

4,571,694 4,626,756 4,700,118 4,706,001 4,728,872 4,732,526 4,742,207 4,835,362 4,894,596 4,899,095 4,920,248 4,931,617 4,934,504 4,956,594 4,967,125 4,969,109 4,970,370 4,970,448 4,979,127 5,004,968 5,006,035 5,008,834 5,063,281 5,066,847 5,066,902 5,093,552 5,107,716 5,111,019 5,130,515 5,136,223 5,151,608 5,170,109 5,189,351 5,267,483 5,274,360 5,292,066 5,300,868 5,304,906 5,313,563 5,319,443 5,325,467 5,327,057 5,329,469 5,333,242 5,337,148 5,371,452 5,375,480 5,418,441 5,432,316 5,440,213 5,442,155 5,444,612 5,449,875 5,451,850 5,461,478 5,463,297 5,467,003 5,471,312 5,479,078 5,485,389 5,485,552 5,486,679 5,489,758 5,493,192 5,504,766 5,511,007 5,520,062 5,528,013 5,532,924 5,548,194 5,552,687 5,558,196 5,561,742 5,570,187 5,570,190 5,572,103 5,581,167 5,582,750 5,587,635 5,600,759 5,608,299 5,608,618 5,624,588 5,630,955 5,637,969 5,639,204 5,641,415 5,650,078 5,658,121 5,668,628 5,687,295 5,691,615 5,698,121 5,708,342 5,715,375 5,719,479 5,727,132 5,742,138 5,742,144 5,748,854 5,749,058 5,760,560 5,773,950 5,783,922 5,799,135 5,812,408 5,841,257 5,845,053 5,872,894 5,887,122 5,911,892 5,912,540 5,920,678 5,937,143 5,980,082 5,983,744 5,987,591 5,988,850 6,023,044 6,032,086 6,040,554 6,059,169 6,088,628 6,097,169 6,114,824 6,124,693 6,140,788 6,141,863 6,157,155 6,160,324 6,163,124 6,177,650 6,180,898 6,181,096 6,188,194 6,208,105 6,212,444 6,219,583 6,226,181 6,236,011 6,236,896 6,250,174 6,278,902 6,279,413 6,285,921 6,298,283 6,321,139 6,324,443 6,328,523 6,330,493 6,340,875 6,356,671 6,377,869 6,382,012 6,384,371 6,396,030 6,414,711 6,424,883 6,431,018 6,434,448 6,445,979 6,459,958 6,463,358 6,484,067 6,486,629 6,507,165 6,654,666 6,665,588 6,680,461 6,696,810 6,728,417 6,763,284 6,772,493 6,845,296 6,853,881 6,888,089 6,898,486 6,917,837 6,928,337 6,965,091 6,970,802 7,038,165 7,069,808 7,084,900 7,092,791 7,133,747 7,143,100 7,149,602 7,131,848 7,161,321 7,171,041 7,174,234 7,173,213 7,177,722 7,177,439 7,181,294 7,181,313 7,280,687 7,283,661 7,291,806 7,299,713 7,315,650 7,324,873 7,328,083 7,330,777 7,333,879 7,355,725 7,359,817 7,373,220 7,376,488 7,386,367 7,464,623 7,447,615 7,445,260 7,474,939 7,486,816 7,495,192 7,501,778 7,502,504 7,508,155 7,512,459 7,525,273 7,526,121

Conventions

WARNING

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

CAUTION

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

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

Safety

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

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

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

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

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

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

CONSIDERING SAFETY FOR YOUR ROBOT INSTALLATION

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

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

Keeping People Safe

The safety of people is always of primary importance in any situation. When applying safety measures to your robotic system, consider the following:

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

Using Safety Enhancing Devices

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

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

Setting Up a Safe Workcell

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

- Sponsor your personnel for training in approved FANUC Robotics training course(s) related to your application. Never permit untrained personnel to operate the robots.
- Install a lockout device that uses an access code to prevent unauthorized persons from operating the robot.
- Use anti–tie–down logic to prevent the operator from bypassing safety measures.
- Arrange the workcell so the operator faces the workcell and can see what is going on inside the cell.
- Clearly identify the work envelope of each robot in the system with floor markings, signs, and special barriers. The work envelope is the area defined by the maximum motion range of the robot, including any tooling attached to the wrist flange that extend this range.
- Position all controllers outside the robot work envelope.

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

Staying Safe While Teaching or Manually Operating the Robot

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

- Never wear watches, rings, neckties, scarves, or loose clothing that could get caught in moving machinery.
- Know whether or not you are using an intrinsically safe teach pendant if you are working in a hazardous environment.

- Before teaching, visually inspect the robot and work envelope to make sure that no potentially hazardous conditions exist. The work envelope is the area defined by the maximum motion range of the robot. These include tooling attached to the wrist flange that extends this range.
- The area near the robot must be clean and free of oil, water, or debris. Immediately report unsafe working conditions to the supervisor or safety department.
- FANUC Robotics recommends that no one enter the work envelope of a robot that is on, except for robot teaching operations. However, if you must enter the work envelope, be sure all safeguards are in place, check the teach pendant DEADMAN switch for proper operation, and place the robot in teach mode. Take the teach pendant with you, turn it on, and be prepared to release the DEADMAN switch. Only the person with the teach pendant should be in the work envelope.

WARNING

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

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

WARNING

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

  - Using a low motion speed, single step the program for at least one full cycle.
  - Using a low motion speed, test run the program continuously for at least one full cycle.
  - Using the programmed speed, test run the program continuously for at least one full cycle.
- Make sure all personnel are outside the work envelope before running production.

Staying Safe During Automatic Operation

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

- Make sure all safety provisions are present and active.

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

Staying Safe During Inspection

When inspecting the robot, be sure to

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

Staying Safe During Maintenance

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

- Never enter the work envelope while the robot or a program is in operation.
- Before entering the work envelope, visually inspect the workcell to make sure no potentially hazardous conditions exist.

- Never wear watches, rings, neckties, scarves, or loose clothing that could get caught in moving machinery.
- Consider all or any overlapping work envelopes of adjoining robots when standing in a work envelope.
- Test the teach pendant for proper operation before entering the work envelope.
- If it is necessary for you to enter the robot work envelope while power is turned on, you must be sure that you are in control of the robot. Be sure to take the teach pendant with you, press the DEADMAN switch, and turn the teach pendant on. Be prepared to release the DEADMAN switch to turn off servo power to the robot immediately.
- Whenever possible, perform maintenance with the power turned off. Before you open the controller front panel or enter the work envelope, turn off and lock out the 3–phase power source at the controller.
- Be aware that when you remove a servomotor or brake, the associated robot arm will fall if it is not supported or resting on a hard stop. Support the arm on a solid support before you release the brake.

WARNING

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

- Release or block all stored energy. Before working on the pneumatic system, shut off the system air supply and purge the air lines.
- Isolate the robot from all remote control signals. If maintenance must be done when the power is on, make sure the person inside the work envelope has sole control of the robot. The teach pendant must be held by this person.
- Make sure personnel cannot get trapped between the moving robot and other equipment. Know the path that can be used to escape from a moving robot. Make sure the escape route is never blocked.
- Use blocks, mechanical stops, and pins to prevent hazardous movement by the robot. Make sure that such devices do not create pinch points that could trap personnel.

WARNING

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

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

KEEPING MACHINE TOOLS AND EXTERNAL DEVICES SAFE

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

Programming Safety Precautions

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

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

Mechanical Safety Precautions

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

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

KEEPING THE ROBOT SAFE

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

Operating Safety Precautions

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

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

Programming Safety Precautions

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

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

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

ADDITIONAL SAFETY CONSIDERATIONS FOR PAINT ROBOT INSTALLATIONS

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

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

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

CAUTION

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

Paint robots are operated in three modes:

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

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

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

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

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

Paint System Safety Features

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

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

The paint system includes the following safety features:

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

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

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. DCS (Dual Check Safety), 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.

HMI Device communication function

5/11/2012 Version 2.2

This function communicates with a HMI device by that the robot controller pretends to be a GE Fanuc Series 90 PLC. Various data of the robot controller correspond to addresses of Series 90 PLC. HMI device accesses robot controller data by accessing the corresponded PLC address.

1. Accessing to I/O ports (%I, %Q, %M %AI, %AQ)

The robot controller DI[1] can be accessed from HMI device as %Q1. Like this, the robot controller I/O ports are corresponded to PLC address as follows.

Robot controller I/O port		PLC address	Example
Digital input	DI[*x*]	%Q*x*	DI[1] ⇔ %Q1
Digital output	DO[*x*]	%I*x*	DO[1] ⇔ %I1
Robot input	RI[*x*]	%Q(5000+*x*)	RI[1] ⇔ %Q5001
Robot output	RO[*x*]	%I(5000+*x*)	RO[1] ⇔ %I5001
UOP input	UI[*x*]	%Q(6000+*x*)	UI[1] ⇔ %Q6001
UOP output	UO[*x*]	%I(6000+*x*)	UO[1] ⇔ %I6001
SOP input	SI[*x*]	%Q(7000+*x*)	SI[0] ⇔ %Q7000
SOP output	SO[*x*]	%I(7000+*x*)	SO[0] ⇔ %I7000
Weld input	WI[*x*]	%Q(8000+*x*)	WI[1] ⇔ %Q8001
Weld output	WO[*x*]	%I(8000+*x*)	WO[1] ⇔ %I8001
Wire stick input	WSI[*x*]	%Q(8400+*x*)	WSI[1] ⇔ %Q8401
Wire stick output	WSO[*x*]	%I(8400+*x*)	WSO[1] ⇔ %I84001
Group input	GI[*x*]	%AQ*x*	GI[1] ⇔ %AQ1
Group output	GO[*x*]	%AI*x*	GO[1] ⇔ %AI1
Analog input	AI[*x*]	%AQ(1000+*x*)	AI[1] ⇔ %AQ1001
Analog output	AO[*x*]	%AI(1000+*x*)	AO[1] ⇔ %AI1001
PMC keep relay DO[*x*] (*x* : 10001 – 10144) K*a*.*b*		 %I*x* %I((*a**8)+*b*+10001)	 DO[10001] ⇔ %I10001 K2.5 ⇔ %I10022
PMC internal relay DO[*x*] (*x* : 11001 – 23000) R*a*.*b*		 %M(*x*-11000) %M((*a**8)+*b*+1)	 DO[11001] ⇔ %M1 R2.5 ⇔ %M22
PMC data table GO[*x*] (*x* : 10001 – 12000) D(*a**2), D((*a**2)+1)		 %AI(*x*-6000) %AI(*a*+4001)	 GO[10001] ⇔ %AI4001 D4, D5 ⇔ %AI4003

Note: In PLC, %I and %AI are input ports, and %Q and %AQ are output ports. But in this correspondence, %Q and %AQ access the robot controller input ports, and %I and %AI access the robot controller output ports.

Note: If you write input ports of the robot controller (DI, RI, AI), the value is changed in an instant, and the value is recovered to actual input value immediately. But this change in an instant may cause problem for your system. Please don't write to the robot controller input ports.

2. Accessing to Robot Registers (%R)

The default correspondence is the following.

Robot controller data		PLC address	Example
Register	R[*x*]	%R*x*	R[1] ⇔ %R1

Note: Please access %R as 16bits signed integer. The value is rounded off to no decimal place. Range of the value is from -32768 to 32767. If value is out of the range, lower 16bits are accessed.

The correspondence between Robot Registers and PLC address %R is defined by setting of $SNPX_ASG. The default setting of $SNPX_ASG is the following. And the above correspondent is defined. If $SNPX_ASG is changed, the above correspondence is not available. Please check $SNPX_ASG setting before accessing Registers.

System variable	Value
$SNPX_ASG[1].$ADDRESS	1
$SNPX_ASG[1].$SIZE	10000
$SNPX_ASG[1].$VAR_NAME	R[1]@1.1
$SNPX_ASG[1].$MULTIPLY	1

$SNPX_ASG is 80 arrays of $SNPX_ASG[1]-[80], every element has 4 members, $ADDRESS, $SIZE, $VAR_NAME, $MULTIPLY. HMI device can access various robot controller data by setting $SNPX_ASG. As explained later, HMI device can access position registers, system variables and etc by setting $SNPX_ASG.

$SNPX_ASG setting to access to Robot Registers.

$SNPX_ASG	Description
$ADDRESS	Meaning: Start address of %R to assign. Range: 1-16384
$SIZE	Meaning: Number of %R address to assign. One Register uses two %R address. Please set the necessary number of %R address according to accessing number of registers. (You can change the number of %R for one register by adding “@” in $VAR_NAME) Range: 1-16384
$VAR_NAME	Meaning: String to define the assigning data. Please set ”R[1]” to assign Register R[1]. The number in [] is the index of register. Continued registers like R[2]-R[5] can be assigned by one $SNPX_ASG element. In this case, please set $SIZE 8 because the number of registers to assign is 4. And please set $VAR_NAME “R[2]”. The index 2 in this case means the starting index of the continuous assignment. If “@1.1” is added just after the string, the number of %R for one register is changed from 2 to 1. In this case, accessing data size becomes 16bits. Example: R[1]@1.1 “R[1]” means to assign robot registers from index 1. “@1.1” means one register is accessed as 16 bit data.
$MULTIPLY	Meaning: The value accessed by HMI is multiplied by $MULTIPLY. If $MULTIPLY is 0, it means special that HMI can access %R as 32bits REAL data. If it is not 0, HMI accesses %R as 32bits signed integer. And value is rounded off to no decimal place. Range: 0.0001-10000, 0 Example: Register value is 123.45. When $MULTIPLY is 1, %R is 123 When $MULTIPLY is 10, %R is 1235. When $MULTIPLY is 0.1, %R is 12. When $MULTIPLY is 0, %R is 123.45 of REAL

For example, $SNPX_ASG is set as follows.

	$ADDRESS	$SIZE	$VAR_NAME	$MULTIPLY
$SNPX_ASG[1]	1	2	R[1]@1.1	1
$SNPX_ASG[2]	3	4	R[1]	100
$SNPX_ASG[3]	7	4	R[2]	0.1
$SNPX_ASG[4]	11	2	R[1]	0

In this case, %R is corresponded to registers as follows.

PLC address	Accessed data
%R1	R[1] as 16bits signed integer.
%R2	R[2] as 16bits signed integer.
%R3-4	Multiplied R[1] by 100 as 32bits signed integer.
%R5-6	Multiplied R[2] by 100 as 32bits signed integer.
%R7-8	Divided R[1] by 10 as 32bits signed integer.
%R9-10	Divided R[2] by 10 as 32bits signed integer.
%R11-12	R[1] as 32bits REAL

$SNPX_ASG[1] defines that 2 %R from %R1 to %R2 are assigned to register from R[1] multiplied by 1 as 16bit signed integer. One %R address is 16 bit, and one register uses one %R. So, %R1 is assigned to R[1] as 16bit signed integer and %R2 is assigned to R[2] as 16bits signed integer.

$SNPX_ASG[2] defines that 4 %R from %R3 to %R6 are assigned to register from R[1] multiplied by 100 as 32bit signed integer. One register uses 2 %R. So, %R3-4 are assigned to multiplied R[1] by 100 as 32bit signed integer and %R5-6 are assigned to multiplied R[2] by 100 as 32bits signed integer.

$SNPX_ASG[3] defines that 4 %R from %R7 to %R10 are assigned to register from R[2] divided by 10 as 32bit signed integer. One register uses 2 %R. So, %R7-8 are assigned to divided R[2] by 10 as 32bit signed integer and %R9-10 are assigned to divided R[3] by 10 as 32bits signed integer.

$SNPX_ASG[4] defines that 2 %R from %R11 to %R12 are assigned to register R[1] as 32bit REAL. So, %R11-12 are assigned to R[1] as 32bits REAL

3. Accessing to Position Registers (%R)

To access Position Register from HMI device, please set $SNPX_ASG to assign Position Registers to %R address. In default setting of $SNPX_ASG, position registers are not assigned. You need to set $SNPX_ASG to access position registers.

$SNPX_ASG setting to access Position Registers

$SNPX_ASG	Description
$ADDRESS	Meaning: Start address of %R to assign. Range: 1-16384
$SIZE	Meaning: Number of %R address to assign. One Position Register uses 50 %R address. Please set the necessary number of %R address according to accessing number of position registers. (You can change the number of %R for one position register by adding “@” in $VAR_NAME) Range: 1-16384
$VAR_NAME	Meaning: String to define the assigning data. Please set ”PR[1]” to assign Position Register PR[1]. The number in [] is the index of position register. Continued position registers like PR[2]-PR[5] can be assigned by one $SNPX_ASG element. In this case, please set $SIZE 200 because the number of position registers to assign is 4. And please set $VAR_NAME “PR[2]”. The index 2 in this case means the starting index of the continuous assignment. In multi group system, PR[1] means group 1 data of PR[1]. Please set PR[G2:1] to assign group 2 data of PR[1]. If “@” is added just after the string, the specified part of the position register is assigned. This is explained later.

$MULTIPLY	Meaning: The value accessed by HMI is multiplied by $MULTIPLY. Only REAL values are affected by this setting. If $MULTIPLY is 0, it means special that HMI can access %R as 32bits REAL data. If it is not 0, HMI accesses %R as 32bits signed integer. And value is rounded off to no decimal place. Range: 0.0001-10000, 0 Example: Position register member value is 123.45. When $MULTIPLY is 1, %R is 123 When $MULTIPLY is 10, %R is 1235. When $MULTIPLY is 0.1, %R is 12. When $MULTIPLY is 0, %R is 123.45 of REAL

One position register uses 50 %R address. Contents of the 50 %R are the following.

%R address	Description		Effect of $MULTIPLY
Cartesian data			
1-2	X	32bits signed integer or real (mm)	Yes
3-4	Y	32bits signed integer or real (mm)	Yes
5-6	Z	32bits signed integer or real (mm)	Yes
7-8	W	32bits signed integer or real (deg)	Yes
9-10	P	32bits signed integer or real (deg)	Yes
11-12	R	32bits signed integer or real (deg)	Yes
13-14	E1	32bits signed integer or real (mm, deg)	Yes
15-16	E2	32bits signed integer or real (mm, deg)	Yes
17-18	E3	32bits signed integer or real (mm, deg)	Yes
19	FLIP	16bits signed integer (1:Flip, 0:Non flip)	No
20	LEFT	16bits signed integer (1:Left, 0:Right)	No
21	UP	16bits signed integer (1:Up, 0:Down)	No
22	FRONT	16bits signed integer (1:Front, 0:Back)	No
23	TURN4	16bits signed integer (-128~127)	No
24	TURN5	16bits signed integer (-128~127)	No
25	TURN6	16bits signed integer (-128~127)	No
26	VALIDC	16bits signed integer (→note1)	No
Joint data			
27-28	J1	32bits signed integer or real (mm, deg)	Yes
29-30	J2	32bits signed integer or real (mm, deg)	Yes
31-32	J3	32bits signed integer or real (mm, deg)	Yes
33-34	J4	32bits signed integer or real (mm, deg)	Yes
35-36	J5	32bits signed integer or real (mm, deg)	Yes
37-38	J6	32bits signed integer or real (mm, deg)	Yes
39-40	J7	32bits signed integer or real (mm, deg)	Yes
41-42	J8	32bits signed integer or real (mm, deg)	Yes
43-44	J9	32bits signed integer or real (mm, deg)	Yes
45	VALIDJ	16bits signed integer (→note2)	No
Frame number			
46	UF	16bits signed integer (-1~62) (→note3)	No
47	UT	16bits signed integer (-1~30) (→note4)	No
48-50	Reserve		No

Note1: VALIDC shows whether the position data is valid for Cartesian or not. This becomes 0 at the following situation, and it becomes 1 in the other situation.

– Position data has uninitialized data (displayed "******" on TP).

– Position data is Joint representation and it can not be converted to Cartesian.

If HMI device writes any data to VALIDC, position representation is changed to Cartesian.

Note1: VALIDJ shows whether the position data is valid for Joint or not. This becomes 0 at the following situation, and it becomes 1 in the other situation.

– Position data has uninitialized data (displayed "******" on TP).

– Position data is Cartesian representation and it can not be converted to Joint.

If HMI device writes any data to VALIDJ, position representation is changed to Joint.

Note3: UF is user frame number
If UF is 0, world frame is used.
If UF is -1, the user fame that is selected now is used.
UF of Position Register is always -1.
This value can not be changed.

Note4: UT is tool frame number
If UT is 0, mechanical interface frame is used.
If UT is -1, the tool fame that is selected now is used.
UT of Position Register is always -1.
This value can not be changed.

Position register has 2 representations, Cartesian or Joint. If detail of the position register is displayed as X,Y,Z,W,P,R, it is Cartesian representation. If it is displayed as J1-6, it is Joint representation.
HMI device can always access to any member without regard to current representation. Position representation is changed by reading from HMI. But if the representation of position register and the representation of accessed member from HMI are different, please note the following.

- If the position data is out of stroke limit or there is uninitialized member, this position data can not be converted to another representation. In this case, all members in the part of another representation become 0.
  For example, a position register is Cartesian representation and X is 1000, it is out of stroke limit, J1-J9 is read as 0 by HMI device. In this case, the members of Cartesian part or Joint part can be read correctly.
- If the representation of position register and the representation of accessed member from HMI are different, communication response time is increased. Because position representation conversion takes about several milliseconds or over ten milliseconds. If HMI reads many position registers, this conversion time may have a big effect to response time.

Uninitialized member of position data is displayed as "******" on position register menu on TP. These members are read as 0 from HMI device. Please read VALIDC or VALIDJ to check whether the value is 0 or uninitialized. If position data has uninitialized member, VALIDJ and VALIDC are 0.

If you write data to the member in Cartesian part from HMI device, the position representation becomes Cartesian. If you write data to member in Joint part from HMI, the position representation becomes Joint. If you write data to UF or UT from HMI device, position representation is not changed.

Extract a part by adding "@" in $VAR_NAME

If "@" is added to $VAR_NAME, the specified part of data structure is extracted. It is also used for Register assignment (R[1]@1.1).
For example, you need access to only X, Y and Z of PR[1-3]. In this case, the setting of $SNPX_ASG should be the following.

	$ADDRESS	$SIZE	$VAR_NAME	$MULTIPLY
$SNPX_ASG[1]	1	150	PR[1]	1

The following is the correspondence between position registers and %R.

PLC address	Accessed data
%R1-2	X of PR[1] as 32bits signed integer
%R3-4	Y of PR[1] as 32bits signed integer
%R5-6	Z of PR[1] as 32bits signed integer
%R51-52	X of PR[2] as 32bits signed integer
%R53-54	Y of PR[2] as 32bits signed integer
%R55-56	Z of PR[2] as 32bits signed integer
%R101-102	X of PR[3] as 32bits signed integer
%R103-104	Y of PR[3] as 32bits signed integer
%R105-106	Z of PR[3] as 32bits signed integer

Actual read data are only 18 %R, but this assignment occupies 150 %R. And, some HMI accesses %R7-50 that is unnecessary to read, because reading one big data is more efficient than reading several small data for communicate with Series 90 PLC. But the address %R27-45 includes Joint representation part, and if position data is Cartesian representation, representation conversion is needed to read this unnecessary data.

To communicate efficiently, please set $SNPX_ASG as follows.

	$ADDRESS	$SIZE	$VAR_NAME	$MULTIPLY
$SNPX_ASG[1]	1	18	PR[1]@1.6	1

The following is the correspondence between position registers and %R.

PLC address	Accessed data
%R1-2	X of PR[1] as 32bits signed integer
%R3-4	Y of PR[1] as 32bits signed integer
%R5-6	Z of PR[1] as 32bits signed integer
%R7-8	X of PR[2] as 32bits signed integer
%R9-10	Y of PR[2] as 32bits signed integer
%R11-12	Z of PR[2] as 32bits signed integer
%R13-14	X of PR[3] as 32bits signed integer

%R15-16	Y of PR[3] as 32bits signed integer
%R17-18	Z of PR[3] as 32bits signed integer

The change of $SNPX_ASG is that "@1.6" is added in $VAR_NAME and $SIZE is changed from 150 to 18. By this change, the number of %R for one position register is changed from 50 to 6. The "6" in "@1.6" means the number of %R for one position register. And the "1" in "@1.6" means the starting address of extracting part in position data structure.

By specifying "@" in $VAR_NAME, you can extract the specified part of position data structure.

@1.6

The size of extracting part (The number of %R for one position)

The starting address of extracting part in position data structure.

You can specify "@" not only for position register, you can specify it for all data that is assigned by $SNPX_ASG.

The following is another example to assign J1-J6 of PR[1-3].

	$ADDRESS	$SIZE	$VAR_NAME	$MULTIPLY
$SNPX_ASG[1]	1	96	PR[3]@27.12	1

4. Accessing to String Registers (%R)

To access String Register from HMI device, please set $SNPX_ASG to assign String Registers to %R address. In default setting of $SNPX_ASG, string registers are not assigned. You need to set $SNPX_ASG to access string registers. It is possible to use accessing to String Register from 7DA7/09 of system software of robot controller.

$SNPX_ASG setting to access String Registers

$SNPX_ASG	Description
$ADDRESS	Meaning: Start address of %R to assign. Range: 1-16384
$SIZE	Meaning: Number of %R address to assign. One String Register uses 40 %R address. Please set the necessary number of %R address according to accessing number of string registers. It is possible to deal with first 80 characters of one string register in default setting. (You can change the number of %R for one string register by adding "@" in $VAR_NAME) Range: 1-16384
$VAR_NAME	Meaning: String to define the assigning data. Please set "SR[1]" to assign String Register SR[1]. The number in [] is the index of string register. Continued string registers like SR[2]-SR[5] can be assigned by one $SNPX_ASG element. In this case, please set $SIZE 160 because the number of string registers to assign is 4. And please set $VAR_NAME "SR[2]". The index 2 in this case means the starting index of the continuous assignment. If "@" is added just after the string, the specified part of the String Register is assigned. Example: SR[1]@1.5 "SR[1]" means to assign String Registers from index 1. "@1.5" means the first 10 characters of String Register data are assigned.

$MULTIPLY	Meaning: This setting defines string data format. Please set 1 or -1 according to HMI device. GE Fanuc CIMPLICITY 1 Total Control Quick Panel -1

5. Reading Current Position (%R)

HMI can read current position. To read current position, please assign current position to %R by $SNPX_ASG.

$SNPX_ASG setting to read Current Position

$SNPX_ASG	Description
$ADDRESS	Meaning: Start address of %R to assign. Range: 1-16384
$SIZE	Meaning: Number of %R address to assign. Current Position uses 50 %R address. (You can change the number of %R for Current Position by adding “@” in $VAR_NAME) Range: 1-16384
$VAR_NAME	Meaning: String to define the assigning data. Please set ”POS[1]” to assign Current Position. The number in [] is user frame number.. When user frame number is 0, HMI can read Current Position on World frame. When user frame number is -1, HMI can read Current Position on the User frame that is selected now. When user frame number is 1-61, HMI can read Current Position on the specified user frame.. Data structure is the same as position register. In multi group system, POS[1] means current position group 1 robot. Please set POS[G2:1] to assign current position of group 2 robot. If “@” is added just after the string, the specified part of current position is assigned.

$MULTIPLY	Meaning: The value accessed by HMI is multiplied by $MULTIPLY. Only REAL values are affected by this setting. If $MULTIPLY is 0, it means special that HMI can access %R as 32bits REAL data. If it is not 0, HMI accesses %R as 32bits signed integer. And value is rounded off to no decimal place. Range: 0.0001-10000, 0 Example: Current position member value is 123.45. When $MULTIPLY is 1, %R is 123 When $MULTIPLY is 10, %R is 1235. When $MULTIPLY is 0.1, %R is 12. When $MULTIPLY is 0, %R is 123.45 of REAL

Note: Current position is read only. If you write it, nothing occurs.

Current position uses 50 %R, and data structure is the same as position register.
UT of current position is always -1.
UF of current position is the specified user frame number in $VAR_NAME.

Current position is not assigned continuously. For example, the following setting of $SNPX_ASG assigns current position of world frame to %R1-50, but current position of user frame 1 is not assigned to %R51-100. Please set “POS[1]” in another $SNPX_ASG.

	$ADDRESS	$SIZE	$VAR_NAME	$MULTIPLY
$SNPX_ASG[1]	1	100	POS[0]	1

6. Reading Alarm History (%R)

HMI can read alarm history. To read alarm history, please assign alarm history to %R by $SNPX_ASG.

$SNPX_ASG setting to read Alarm history.

$SNPX_ASG	Description
$ADDRESS	Meaning: Start address of %R to assign. Range: 1-16384
$SIZE	Meaning: Number of %R address to assign. One alarm uses 100 %R address. Please set the necessary number of %R address according to accessing number of alarms. (You can change the number of %R for one alarm by adding “@” in $VAR_NAME) Range: 1-16384
$VAR_NAME	Meaning: String to define the assigning data. Please set ”ALM[1]” to assign alarm. The number in [] is line number in alarm menu. “ALM[1]” assigns active alarm. Alarms displayed in active alarm menu can be read. “ALM[E1]” assigns alarm history. Alarms displayed in alarm history menu can be read. “ALM[P1]” assigns password log. Password log displayed in password log menu can be read. “ALM[M1]” assigns motion alarm history. Alarm displayed in motion alarm menu can be read. “ALM[A1]” assigns application alarm history. Alarm displayed in application alarm menu can be read. “ALM[S1]” assigns system alarm history. Alarm displayed in system alarm menu can be read.
$MULTIPLY	Meaning: This setting defines string data format. Please set 1 or -1 according to HMI device. GE Fanuc CIMPLICITY 1 Total Control Quick Panel -1 Range: 1,-1

One alarm uses 100 %R address. Contents of the 100 %R are the following.

%R address	Description
1	Alarm ID 16bit signed integer When alarm is “SRVO-001”, alarm ID is 11 that mean “SRVO”. Please refer operation manual for the value of alarm ID
2	Alarm number 16bits signed integer When alarm is “SRVO-001”, alarm number is 1.
3	Alarm ID of Cause Code 16bits signed integer When alarm occurs, alarm message is displayed on top of teach pendant. Sometimes a message is also displayed on the second line, it is cause code. This member shows alarm ID of cause code. If the alarm does not have cause code, this becomes 0.
4	Alarm number of cause code 16bit signed integer Alarm number of cause code. If the alarm does not have cause code, this becomes 0.
5	Alarm severity 16bit signed integer The value shows alarm severity. NONE 128 WARN 0 PAUSE.L 2 PAUSE.G 34 STOP.L 6 STOP.G 38 SERVO 54 ABORT.L 11 ABORT.G 43 SERVO2 58 SYSTEM 123
6	Occurred Time (year) 16bits signed integer
7	Occurred Time (month) 16bits signed integer
8	Occurred Time (day) 16bits signed integer
9	Occurred Time (hour) 16bits signed integer
10	Occurred Time (minutes) 16bits signed integer
11	Occurred Time (second) 16bits signed integer
12-51	Alarm message 80 characters string Alarm message string. It shows the string as the same as that is displayed on top line of teach pendant.
52-91	Cause code alarm message 80 characters string Alarm message of cause code.
92-100	Alarm severity word 18- characters string String of alarm severity like a “WARN”.

Alarm ID and alarm number of “RESET” is read as 0. Alarm message is “RESET”.

When there are 2 lines on active alarm menu, all members of alarms after ALM[3] are 0.

In string item, address after string are 0.

One alarm uses 100 %R, if you would like to read only alarm ID and alarm number, most of 100 %R area is not used. To communicate efficiently, please use “@”.

For example

	$ADDRESS	$SIZE	$VAR_NAME	$MULTIPLY
$SNPX_ASG[1]	1	12	ALM[E1]@1.4	1

This assigns alarm history as follows.

PLC address	Accessed data
%R1	Alarm ID of alarm1
%R2	Alarm number of alarm1
%R3	Alarm ID of cause code of alarm1
%R4	Alarm number of cause code of alarm1
%R5	Alarm ID of alarm2
%R6	Alarm number of alarm2
%R7	Alarm ID of cause code of alarm2
%R8	Alarm number of cause code of alarm2
%R9	Alarm ID of alarm3
%R10	Alarm number of alarm3
%R11	Alarm ID of cause code of alarm3
%R12	Alarm number of cause code of alarm3

7. Reading Program Execution Status (%R)

HMI device can read Program Execution Status. To read program execution status, please assign it to %R by $SNPX_ASG.

$SNPX_ASG setting to read program execution status.

<table>
<tr><th>$SNPX_ASG</th><th colspan="2">Description</th></tr>
<tr><td>$ADDRESS</td><td>Meaning:
Range:</td><td>Start address of %R to assign.
1-16384</td></tr>
<tr><td>$SIZE</td><td>Meaning:

Range:</td><td>Number of %R address to assign.
One task uses 18 %R address. Please set the necessary number of %R address according to accessing number of tasks.
(You can change the number of %R for one task by adding “@” in $VAR_NAME)
1-16384</td></tr>
<tr><td>$VAR_NAME</td><td>Meaning:</td><td>String to define the assigning data.
Please set ”PRG[1]” to assign program execution status. The number in [] is task number.
In single task system, program execution status always can be read by PRG[1].

In multi task system, when 2 tasks are running at the same time, PRG[1] and PRG[2] shows the execution status of every task.
Which task is PRG[1] is decided by timing of execution and communication. But the task that is read by PRG[1] is always read by PRG[1] until it is aborted.</td></tr>
<tr><td>$MULTIPLY</td><td>Meaning:

Range:</td><td>This setting defines string data format. Please set 1 or -1 according to HMI device.
GE Fanuc CIMPLICITY 1
Total Control Quick Panel -1
1,-1</td></tr>
</table>

One task uses 18 %R address. Contents of the 18 %R are the following.

%R address	Description
1-8	Program name 16 characters string Name of running program. When sub program is called, it shows sub program name.
9	Line number 16bits signed integer. Execution line number. When sub program is called, it shows line number of sub program.
10	Execution status 16bits signed integer Aborted 0 Paused 1 Running 2
11-18	Parent program name 16 characters string Name of the started program When sub program is not called, it is the same as program name.

When program is aborted, all members become 0.

By assigning each strings to $VAR_NAME, the type of read program name can be selected. And the meaning of line number is changed depending on the situation.

$VAR_NAME	Program name and line number
PRG[1]	Program name Name of running program. Line number Execution line number.
PRG[M1]	Program name When running program is MACRO program, it shows name of parent program that calls the running program. When the parent program is also MACRO program, it shows name of parent program that calls the parent program. Line number Line number of CALL instruction that calls the running program.
PRG[K1]	Program name When running program is KAREL program, it shows name of parent program that calls the running program. When the parent program is also KAREL program, it shows name of parent program that calls the parent program. Line number Line number of CALL instruction that calls the running program.
PRG[MK1] or PRG[KM1]	Program name When running program is MACRO or KAREL program, it shows name of parent program that calls the running program. When the parent program is also MACRO or KAREL program, it shows name of parent program that calls the

	parent program. Line number Line number of CALL instruction that calls the running program.

By adding "@", a part of this structure is assigned to %R.

For example.

	$ADDRESS	$SIZE	$VAR_NAME	$MULTIPLY
$SNPX_ASG[1]	1	4	PRG[1]@9.2	1

This assigns program execution status as follows.

PLC address	Accessed data
%R1	Line number of task1
%R2	Execution status of task1
%R3	Line number of task 2.
%R4	Execution status of task2.

8. Accessing to System Variables (%R)

HMI can access system variables. To access system variables, please assign it to %R by $SNPX_ASG.

$SNPX_ASG setting to access system variables.

$SNPX_ASG	Description
$ADDRESS	Meaning: Start address of %R to assign. Range: 1-16384
$SIZE	Meaning: Number of %R address to assign. The number of %R for one system variable is defined by data type of the system variable. Please refer to the following data type list. (You can change the number of %R for one system variable by adding "@" in $VAR_NAME) Range: 1-16384
$VAR_NAME	Meaning: String to define the assigning data. Please set system variable name, for example "WAITTMOUT". If array system variable like a "UALRM_SEV[1]" is set, array elements are assigned continuously like a Registers. From version 2, KAREL string can be specified by the following format. $[KAREL program name]variable name.
$MULTIPLY	Meaning: The meaning of $MULTIPLY is defined according to the data type of system variable. Please refer the following data type list.

To access system variable, data type access by HMI device and meaning of $MULTIPLY are changed according to data type of system variable. The following is the list that can be accessed by HMI device, and it also explain the number of %R to use and meaning of $MULTIPLY.

Data type of system variables	The number of %R for one system variable	Meaning of $MULTIPLY and data type accessed by HMI
INTEGER 32bits signed integer	2	The value accessed by HMI is multiplied by $MULTIPLY. If $MULTIPLY is 0, it is same as $MULTIPLY is 1. If it is not 0, HMI accesses %R as 32bits signed integer. And value is rounded off to no decimal place
SHORT 16bits signed integer	2	
BYTE 8bits signed integer	2	
REAL 32bits real	2	The value accessed by HMI is multiplied by $MULTIPLY. If $MULTIPLY is 0, it means special that HMI can access %R as 32bits REAL data. If it is not 0, HMI accesses %R as 32bits signed integer. And value is rounded off to no decimal place
BOOLEAN TRUE/FALSE	2	HMI accesses %R as 32bit signed integer. If value is TRUE %R is 1. If it is 0, %R is 0. If HMI writes 0, the system variable becomes FALSE. If HMI writes not 0, the system variable becomes TRUE. $MULTIPLY is not used for this data type.
POSITION Position data	50	Data structure is the same as position register. Meaning of $MULTIPLY is also same as position register.
STRING String data	40	This defines string data format. GE Fanuc CIMPLICITY 1 Total Control Quick Panel -1

INTEGER, SHORT and BYTE are access by the same way from HMI device. The system variable that integer value is displayed in system variable menu on teach pendant is INTEGER, SHORT or BYTE.

The variable that is a real value is displayed in system variable menu on teach pendant is REAL.

The variable that is TRUE or FALSE is displayed in system variable menu on teach pendant is BOOLEAN.

The variable that is a string is displayed on system variable menu on teach pendant is STRING.

The variable that is displayed as POSITION in system variable menu on teach pendant is POSITION.

Note: The protected variable in system variable menu on teach pendant can be changed by HMI device. If illegal value is set to system variable, system may be harmed seriously. Please check value enough to write system variables.

By adding “@”, a part of this structure is assigned to %R.
For example. The following $SNPX_ASG assigns X, Y and Z of user frame 1 and 2 of group 1.

	$ADDRESS	$SIZE	$VAR_NAME	$MULTIPLY
$SNPX_ASG[1]	1	12	$MNUFRAME[1,1]@1.6	1

This assigns system variables to %R as follows.

PLC address	Accessed data
%R1-2	X of user frame 1 of Group 1 (X of $MNUFRAME[1,1])
%R3-4	Y of user frame 1 of Group 1 (Y of $MNUFRAME[1,1])
%R5-6	Z of user frame 1 of Group 1 (Z of $MNUFRAME[1,1])
%R7-8	X of user frame 2 of Group 1 (X of $MNUFRAME[1,2])
%R9-10	Y of user frame 2 of Group 1 (Y of $MNUFRAME[1,2])
%R11-12	Z of user frame 2 of Group 1 (Z of $MNUFRAME[1,2])

9. Accessing to comment of R[ ], PR[ ] , SR[ ] and I/O (%R)

HMI device can access comment of register, position register, string register and I/O. To access comment, please assign it to %R by $SNPX_ASG.

$SNPX_ASG setting to access register, position register, string register and I/O.

$SNPX_ASG	Description
$ADDRESS	Meaning: Start address of %R to assign. Range: 1-16384
$SIZE	Meaning: Number of %R address to assign. One comment uses 40 %R address. Please set the necessary number of %R address according to accessing number of comment. (You can change the number of %R for one comment by adding "@" in $VAR_NAME) Range: 1-16384
$VAR_NAME	Meaning: String to define the assigning data. Please set "R[C1]" to assign comment of register. The number in [] is index of register. To assign comment of position register, string register and I/O, please set the following string. Register R[C1] Position register PR[C1] String register SR[C1] SDI DI[C1] SDO DO[C1] RDI RI[C1] RDO RO[C1] UI UI[C1] UO UO[C1] SI SI[C1] SO SO[C1] WI WI[C1] WO WO[C1] WSI WSI[C1] WSO WSO[C1] GI GI[C1] GO GO[C1] AI AI[C1] AO AO[C1] Continued registers like P[2]-P[5] can be assigned by one $SNPX_ASG element. In this case, please set $SIZE 160 because the number for one comment to assign is 40. And please set $VAR_NAME "P[C2]". The index 2 in this case means the starting index of the continuous assignment. If "@" is added just after the string, the specified

	part of the comment is assigned. Example: R[C1]@1.5 "R[C1]" means to assign comment of registers from index 1. "@1.5" means the first 10 characters of comment are assigned.
$MULTIPLY	Meaning: This setting defines string data format. Please set 1 or -1 according to HMI device. GE Fanuc CIMPLICITY 1 Total Control Quick Panel -1 Range: 1,-1

10. Accessing I/O value and simulation status (%R)

HMI device can access values and simulation status of I/O. To access values and simulation status, please assign it to %R by $SNPX_ASG.

I/O value can be accessed by %I, %Q, %AI and %AQ, they also can be accessed by %R.
And I/O simulation status also can be accessed by %R.
Please assign it to %R by $SNPX_ASG.

This function is available from Version 2.

$SNPX_ASG setting to access I/O value and simulation status.

$SNPX_ASG	Description
$ADDRESS	Meaning: Start address of %R to assign. Range: 1-16384
$SIZE	Meaning: Number of %R address to assign. The number of %R for one I/O value or simulation status is changed by $MULTIPLY setting. When $MULTIPLY is 1, one I/O value or simulation status uses one %R. When $MULTIPLY is 0, one %R includes 16 I/O value or simulation status as bit image. (Note: Value of GI/O and AI/O use one %R even if $MULTIPLY is 0.) Range: 1-16384
$VAR_NAME	Meaning: String to define the assigning data. Please set "DI[1]" to assign value of DI[1]. The number in [] is index of DI. To assign value or simulation status of the other type of I/O, please set the following string. Value / Simulation SDI DI[1] DI[S1] SDO DO[1] DO[S1] RDI RI[1] RI[S1] RDO RO[1] RO[S1] UI UI[1] UO UO[1] SI SI[1]

	SO SO[1] WI WI[1] WI[S1] WO WO[1] WO[S1] WSI WSI[1] WSI[S1] WSO WSO[1] WSO[S1] GI GI[1] GI[S1] GO GO[1] GO[S1] AI AI[1] AI[S1] AO AO[1] AO[S1] Example: Assign SDI[11]-SDI[42]: When $MULTIPLY is 1, please set 32 to size because the number of DI is 32, and set “DI[11]” to $VAR_NAME. In this case, index 11 means that %R are assigned from SDI[11]. When $MULTIPLY is 0, 16 DI values are assigned to one %R, so please set 2 to $SIZE, and set “DI[11]” to $VAR_NAME.
$MULTIPLY	Meaning: One %R has one I/O data or 16 I/O data as bit image. One I/O data is assigned to one %R 1 16 I/O data are assigned to one %R 0

By setting $MULTIPLY you can select whether one I/O data is assigned to one %R or 16 I/O data are assigned to one %R.

When $MULTIPLY is 1, one I/O values or one simulation status are assigned to one %R. When the value is ON, the corresponded %R is set to 1. When the value is OFF, the corresponded %R is set to 0.

When $MULTIPLY is 0, 16 I/O values or 16 simulation status are assigned to one %R. When the value is ON, the corresponded bit of %R is set to 1. When the value is OFF, the corresponded bit of %R is set to 0. I/O of lower index is assigned to lower bit of %R. Example: When DI[1-16] is assigned to %R1, if only DI[1] is ON and the others are OFF, %R1 is set to 1. If only SDI[16] is ON and the others are OFF, %R1 is set to 32768 (in case of unsigned 16 bit integer).

11. Set $SNPX_ASG from HMI device (%G)

You need to set $SNPX_ASG to access data except I/O ports. You can set $SNPX_ASG in system variable. And you can also set it by HMI device.
Please note HMI device must support writing string to use this function.

If HMI device can write a string to PLC address %G, the string is interpreted as a command. Any address in %G can accept command string. The following commands are supported.

CLRASG Clear all $SNPX_ASG

Format: CLRASG

Function: All $SNPX_ASG members in all elements are cleared.
Please execute this command once before use SETASG commands.

SETASG Set $SNPX_ASG

Format: SETASG ($ADDRESS) ($SIZE) ($VAR_NAME) [($MULTIPLY)]

Function: Set specified value to $SNPX_ASG. The unused element of $SNPX_ASG array is selected automatically.
$MULTIPLY can be omitted. If $MULTIPLY is not specified, $MULTIPLY becomes 1.
Please execute CLRASG once before execute SETASG.

SETVAR Set system variable

Format: SETVAR (System variable name) (value)

Function: The specified value is set in the specified system variable.
Available data type of system variable is INTEGER, SHORT, BYTE, REAL, BOOLEAN and STRING.
To set BOOLEN type variable, set 1 for TRUE, set 0 for FALSE.
To set STRING type variable, please use “” to set string including space character.

CLRALM Clear alarm history

Format: CLRALM

Function: All alarm histories are cleared
This function is available from version 2.

Example: If you would like to set $SNPX_ASG as follows.

	$ADDRESS	$SIZE	$VAR_NAME	$MULTIPLY
$SNPX_ASG[1]	1	2	R[1]@1.1	1
$SNPX_ASG[2]	3	4	R[1]	100
$SNPX_ASG[3]	7	4	R[2]	0.1
$SNPX_ASG[4]	11	2	R[1]	0

Please write these strings as this order.

```
CLRASG
SETASG 1 2 R[1]@1.1 1
SETASG 3 4 R[1] 100
SETASG 7 4 R[2] 0.1
SETASG 11 2 R[1] 0
```

From version2, multiple commands can be sent at one request. Please connect command strings by carriage return (13) or line feed (10).

* Processing time may be reduced by adding carriage return (13) or line feed (10) to the last command when multiple commands are sent.

12. Hints

12.1. Assigned data is not read correctly

If HMI device reads %R that is not assigned to any data, this %R is always read as 0.

If %R that the problem occurs is not 0, this %R is assigned to the other data.

For example, when $SNPX_ASG is set as follows, %R101-%R150 is used by both $SNPX_ASG[1] and $SNPX_ASG[2].

	$ADDRESS	$SIZE	$VAR_NAME	$MULTIPLY
$SNPX_ASG[1]	1	1000	R[1]@1.1	1
$SNPX_ASG[2]	101	50	PR[1]	100

In this case, the $SNPX_ASG whose index is smaller is used.

So, %R101-%R150 is assigned to R[101]-R[150], and PR[1] can not be accessed by HMI device.

If %R that the problem occurs is 0, there may also be duplicated assignment as above.

Please check duplication of $SNPX_ASG.

If assignment is not duplicated, $SNPX_ASG setting has problem. Please check $SNPX_ASG. The following are the problems that occur frequently.

- Correct format of $VAR_NAME setting is the following. If set string is not matched to them, the data is not assigned.

“R[n]”	
“PR[n]” “PR[Gn:n]	If you specify group number, “:” must be specified between group number and index.
“SR[n]”	
“POS[n]” “POS[Gn:n]	If you specify group number, “:” must be specified between group number and index.
“ALM[n]” “ALM[En]” “ALM[Pn]	Alarm line number must be specified just after E, M, A, S or P.
“PRG[n]”	
System variable name	The first character of system variable must be “$”.
$[KAREL program name]variable name	Specify KAREL program name just after ‘$[‘, then specify ‘]’ and specify KAREL variable name just after ‘]’.
“DI[n]” “DI[Sn]” “DI[Cn]” “DO[n]” “DO[Sn]” “DO[Cn]” “RI[n]” “RI[Sn]” “RI[Cn]”, “RO[n]” “RO[Sn]” “RO[Cn]” “UI[n]” “UI[Sn]” “UI[Cn]” “UO[n]” “UO[Sn]” “UO[Cn]” “SI[n]” “SI[Sn]” “SI[Cn]” “SO[n]” “SO[Sn]” “SO[Cn]” “WI[n]” “WI[Sn]” “WI[Cn]”,	I/O index number must be specified just after S or C

"WO[n]" "WO[Sn]" "WO[Cn]" "WSI[n]" "WSI[Sn]" "WSI[Cn]" "WSO[n]" "WSO[Sn]" "WSO[Cn]" "GI[n]" "GI[Sn]" "GI[Cn]" "GO[n]" "GO[Sn]" "GO[Cn]" "AI[n]" "AI[Sn]" "AI[Cn]" "AO[n]" "AO[Sn]" "AO[Cn]"	

- If there is space character in $VAR_NAME, data is not assigned. If you specify "@", please do not add any space before "@".
- If you use continuous array assignment, the number of %R for one element is defined according to data type, please check explanation of assigning data.
- The format of "@" is "@n.n". A "." must be specified between starting address and the number of %R. A "@" must be specified just after variable name.

12.2. Communicate efficiently

- Accessing I/O signal is faster than accessing %R that various data is assigned by $SNPX_ASG. And accessing to %R may be slow when program is running, but accessing to I/O ports are not effected by program execution. To communicate efficiently, please reduce accessing to %R area.
- Position representation conversion takes much time. If you access position register, please assign only necessary members of position structure by using "@". And please change position representation of the position registers to the representation that is accessed by HMI device.
- Some HMI device read the address that is not necessary. In this case, please eliminate unassigned %R between assigned %R.

12.3. Version of HMI device communication function

The version of HMI device communication function is in system variable $SNPX_PARAM.$VERSION. If the variable does not exist, the system is version 1.
The following functions are supported from Version 2.

- Access to comment of register, position register and I/O
- Access WI/O and WSI/O.
- Access I/O simulation status
- CLRALM command
- Send multiple commands at one request.
- Access I/O value via %R
- Access KAREL variables
- Multi connection and private $SNPX_ASG

12.4. Multi connection and private $SNPX_ASG

From version 2, multiple HMI devices can be connected to one robot controller via Ethernet. (Only one HMI device can be connected via RS-232-C.)
By the default setting, multi connection is disabled. To enable multi connection, please set the number of connections to system variable $SNPX_PARAM.$NUM_CIMP.
(Note: When a large number is set in $SNPX_PARAM.$NUM_CIMP, a lot of memory is used and robot system may have some problem.)

When multiple HMI devices are connected to one robot controller, all HMI devices use the same $SNPX_ASG. The setting of $SNPX_ASG will be conflicted, and proper data can not be accessed.
To solve this problem, private $SNPX_ASG is supported from Version 2.
In Version 1, CLRASG command clears all $SNPX_ASG system variable. In Version 2, CLRASG command creates private $SNPX_ASG. The private $SNPX_ASG is used by the connection that execute CLRASG command only.
After CLRASG command, SETASG command and accessing to %R by the connection uses the private $SNPX_ASG.
When the connection is closed, the private $SNPX_ASG is deleted.
After CLRASG command, the result of SETASG command is not reflected to the system variable $SNPX_ASG.
If CLRASG command is not executed, SETASG command and accessing %R of the connection uses system variable $SNPX_ASG.

To inhibit private $SNPX_ASG in Version 2 system, please set 0 to $SNPX_PARAM.$NUM_CIMP. (Default)
When more than $SNPX_PARAM.$NUM_CIMP connections are requested, the connection that the last communication is the oldest is closed.

Quick Panel Supplement

1. Introduction

This manual is meant to be a supplement to the Human to Machine Interface (HMI) Device communication function manual. This manual explains the address correspondence between the R-J3 controller and the Total Control Products HMI.

2. Accessing I/O ports (%IBI, %QBI, %MBI %AIUI, %AQUI)

R-J3 I/O can be accessed from a Total Control Products HMI device as %IBI, %QBI, %MBI, %AIUI, or %AQUI. R-J3 I/O ports correspond to Total Control Products address as follows:

R-J3 I/O port		Total Control Products address	Example
Digital input	DI[*x*]	%QBI*x*	DI[1] ⇔ %QBI1
Digital output	DO[*x*]	%IBI*x*	DO[1] ⇔ %IBI1
Robot input	RI[*x*]	%QBI(5000+*x*)	RI[1] ⇔ %QBI5001
Robot output	RO[*x*]	%IBI(5000+*x*)	RO[1] ⇔ %IBI5001
UOP input	UI[*x*]	%QBI(6000+*x*)	UI[1] ⇔ %QBI6001
UOP output	UO[*x*]	%IBI(6000+*x*)	UO[1] ⇔ %IBI6001
SOP input	SI[*x*]	%QBI(7000+*x*)	SI[0] ⇔ %QBI7000
SOP output	SO[*x*]	%IBI(7000+*x*)	SO[0] ⇔ %IBI7000
Weld input	WI[*x*]	%QBI(8000+*x*)	WI[1] ⇔ %QBI8001
Weld output	WO[*x*]	%IBI(8000+*x*)	WO[1] ⇔ %IBI8001
Wire stick input	WSI[*x*]	%QBI(8400+*x*)	WSI[1] ⇔ %QBI8401
Wire stick output	WSO[*x*]	%IBI(8400+*x*)	WSO[1] ⇔ %IBI84001
Group input	GI[*x*]	%AQUI*x*	GI[1] ⇔ %AQUI1
Group output	GO[*x*]	%AIUI*x*	GO[1] ⇔ %AIUI1
Analog input	AI[*x*]	%AQBI(1000+*x*)	AI[1] ⇔ %AQUI1001
Analog output	AO[*x*]	%AIUI(1000+*x*)	AO[1] ⇔ %AIUI1001
PMC keep relay DO[*x*] (*x* : 10001 – 10160) K*a*.*b*		 %IBI*x* %IBI((*a**8)+*b*+10001)	 DO[10001] ⇔ %IBI10001 K2.5 ⇔ %IBI10022
PMC internal relay DO[*x*] (*x* : 11001 – 23000) R*a*.*b*		 %MBI(*x*-11000) %MBI((*a**8)+*b*+1)	 DO[11001] ⇔ %MBI1 R2.5 ⇔ %MBI22
PMC data table GO[*x*] (*x* : 10001 – 11500) D(*a**2), D((*a**2)+1)		 %AIUI(*x*-6000) %AIUI(*a*+4001)	 GO[10001] ⇔ %AIUI4001 D4, D5 ⇔ %AIUI4003

3. Accessing Robot Registers (%R)

The R-J3 to Total Control Products correlation is as follows:

R-J3 data	Total Control Products address	Example
Robot Register R[*x*]	%RA*x* Register ASCII	Robot register can not be set to ASCII, Do not use %RA for Robot registers.
	%RBT*x* Register BIT	R[1] ⇔ %RBT1
	%RBD4*x* Register BCD	R[1] ⇔ %RBD41
	%RI*x* Register Integer	R[1] ⇔ %RI1
	%RUI*x* Register Unsigned Integer	R[1] ⇔ %RUI1
	%RLI*x* Register Long Integer	R[1] ⇔ %RLI1
	%RLUI*x* Register Long Unsigned Integer	R[1] ⇔ %RLUI1
	%RR*x* Register Real	R[1] ⇔ %RR1

4. Accessing Position Registers (%R)

When accessing Position Register data you must use the appropriate %R address for the different values of $MULTIPLY.

$MULTIPLY value	Total Control Products address	Explanation
0	%RR*x* Register Real	Robot positional data is accessed as Real number.
Any other value	%RBT*x* Register BIT	Robot positional data is accessed as BIT number.
	%RBD4*x* Register BCD	Robot positional data is accessed as BCD number.
	%RI*x* Register Integer	Robot positional data is accessed as Integer number.
	%RUI*x* Register Unsigned Integer	Robot positional data is accessed as Unsigned Integer number.
	%RLI*x* Register Long Integer	Robot positional data is accessed as Long Integer number.
	%RLUI*x* Register Long Unsigned Integer	Robot positional data is accessed as Unsigned Integer number.

5. Reading Current Position (%R)

When accessing Current Position data you must use the appropriate %R address for the different values of $MULTIPLY. This is explained in “3. Accessing Position Registers (%R).”

6. Reading Alarm History (%R)

When accessing Alarm History data you must use the appropriate %R for the different pieces of data.

Data Description	Total Control Products address
Alarm ID Alarm Number Alarm ID of Cause Code Alarm number of Cause Code Alarm Severity Occurred Time (year) Occurred Time (month) Occurred Time (day) Occurred Time (hour) Occurred Time (minute) Occurred Time (second)	%RI*x* Register Integer
Alarm Message Cause Code Alarm Message Alarm Severity Word	%RA*x* Register ASCII

7. Reading Program Execution Status (%R)

When accessing Program Execution Status data you must use the appropriate %R for the different pieces of data.

Data Description	Total Control Products address
Line number Execution status	%RI*x* Register Integer
Program name Parent program name	%RA*x* Register ASCII

8. Accessing System Variables (%R)

When accessing System Variable data you must use the appropriate %R for the different pieces of data.

System Variable Data Type	Total Control Products address
Integer Variables	%RLI*x* Register Long Integer %RLUI*x* Register Long Unsigned Integer
Short	%RI*x* Register Integer %RUI*x* Register Unsigned Integer
Byte	%RI*x* Register Integer %RUI*x* Register Unsigned Integer
Real	%RR*x* Register Real
Boolean	%RBT*x* Register BIT
Position (Positional Data)	%RR*x* Register Real
String Variables	%RA*x* Register ASCII

9. Accessing comments of R[ ], PR[ ] and I/O (%R)

Data Description	Total Control Products address
Comment String	%RA*x* Register ASCII

10. Accessing I/O value and simulation status (%R)

When accessing I/O simulation status you must use the appropriate %R for the different pieces of data.

Data Description	Total Control Products address
Integer Variables when $MULTIPLY = 0	%RLI*x* Register Long Integer
Boolean when $MULTIPLY = 1	%RBT*x* Register BIT

11. Set $SNPX_ASG from by HMI device (%G)

Total Control Products devices do not support ASCII entry.