FANUC Robotics
R–J3iB Mate Controller
LR Handling Tool
Operator's Manual

MAROIBLRH02021E REV. A

B–81524EN/01

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

The descriptions and specifications contained in this manual were in effect at the time this manual was approved for printing. FANUC Robotics North 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 North America, Inc.
Training Department
3900 W. Hamlin Road
web site: www.fanucrobotics.com

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

Conventions Used in this Manual

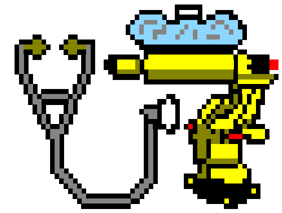
This manual includes information essential to the safety of personnel, equipment, software, and data. This information is indicated by headings and boxes in the text.

> **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.

FANUC Robotics cRc Customer Resource Center



CUSTOMER RESOURCE CENTER

1-800-47-ROBOT

www.fanucrobotics.com

(Canada & United States: 1-800-477-6268) or (International: 011-1-248-377-7159)

TECHNICAL SUPPORT, FIELD SERVICES & ROBOT REPAIR

Press 1 (24 Hours/365 Days)

Facsimile = 248-377-7463

- Technical Support "Hot-Line"
- Service personnel dispatch
- After-hours parts support (8:00 p.m. to 7:00 a.m.)

For best call results have:

- Customer number (if known)
- Company name
- Your name
- Your phone & fax numbers
- Robot & controller type
- "F#" or serial number of robot
- Hour meter reading (if available
- Software type and edition
- Any error messages and LED displays (if applicable)
- Your P.O., credit card, or receiving # for warranty, down robot, or preventive maintenance service orders

PARTS & PART REPAIR

Press 2 ("After Hours" Press 1)

Facsimile = (248) 377-7832

- Parts for down robots
- Replenishment & repair orders
- Warranty part replacement
- Robot software and PACs

For best call results have:

- Customer number (if known)
- Company name
- Your name
- Your phone & fax numbers
- Part name & number (if known)
- "F#" or serial number of robot, if available (req'd for warranty)
- P.O., credit card, or receiving # for warranty, down units, or software
- Shipping & billing addresses
- Reason for repair (any symptoms, error codes, or diagnostic LEDs that were identified)

PRODUCT & SYSTEMS TRAINING

Press 3 (8am-5pm EST; M-F)

Facsimile = 248-377-7367

- Training class registration
- Consultation for non-standard training classes or on-site requests

For best call results have:

- Customer number (if known)
- Company name
- Your name
- Your phone & fax numbers
- Your billing address
- Types of courses needed
- Special requirements
- Robot and controller type
- Proposed schedules
- Number of people attending
- Student names (if available)
- Method of payment (P.O., credit card, etc.)

MARKETING, SALES, & LITERATURE REQUESTS

Press 4 (8am-5pm EST; M-F)

Facsimile = 248-377-7366

- Marketing information
- Application review
- New robot product sales
- Systems solution sales

For best call results have:

- Company name
- Company address
- Your name
- Your phone & fax numbers
- Description of your need

NOTE: A RETURN AUTHORIZATION ("RA") FROM "PARTS" IS REQUIRED BEFORE SHIPPING ANY MATERIAL BACK TO FANUC ROBOTICS FOR PROPER RECEIVING & TRACKING. F# IS LOCATED ON THE ROBOT BASE OR OPERATION PANEL.

Revised:11/20/01

FANUC Robotics North America, Inc.
3900 West Hamlin Road
Rochester Hills, Michigan 48309-3253
www.fanucrobotics.com

FANUC Robotics North America Customer Resource Center (cRc) Aftermarket Support Solutions

- **Parts Solutions-** customized spare part packages, individual spare part pricing and availability, warranty parts, refurbished parts inventory, core part credit, 24 hour emergency service on down robots

- **Certified Repair/Rebuild Solutions-** certified repair/rebuild capabilities for over 500 different components and multiple controller product platforms completed in the Rochester Hills facility, authorized repair for North America, repaired with authentic FANUC parts, certified repair/rebuild serial identification, warranty identification tracked, core part credits on certified repairs
  - *All certified repairs are backed by the same one-year warranty as a new FANUC Robotics part

- **Rebuild Services Levels**
  - **Level 1 Refurbishment Service**- preventive maintenance service for robots with less than 10,000 servo hours
  - **Level 2 Refurbishment Service-**thorough diagnostic and maintenance service 10,000 - 30,000 servo hours
  - **Rebuild Service-** major overhaul service greater than 30,000 servo hours
  - **Rebuilt Robots**- sale of certified rebuilt robots
  - *All certified repairs are backed by the same one-year warranty as a new FANUC Robotics part

- **Training Solutions-** Robot Operations Computer Based Training CD-Rom, over 50 courses available providing standard & customized training solutions that improve your ability to optimize and troubleshoot robot and system performance

- **NO Charge Web Solutions (Requires Registration)**
  - **cRc Parts Solutions**- parts search engine, create/view parts quotes, submit order online, order tracking, multiple tracking options, order status (including backorder information), detailed shipment breakdowns, and a direct link to carrier's website
  - **Return Solutions-** new or refurbished parts, certified component repairs, warranty orders by F numbers, core credit, printable return labels, online return tracking, authorization tracking number online, online credit value viewing

- **Customized Web Solutions**
  - **Electronic Documentation Solutions-** engineering bulletins, products, operations, and maintenance manuals, multi-platform eDiagnostic tools 'DRC', minimize mean time to repair
  - **Application Back-Up Solutions-** scheduled on-site manual back-up, preventive maintenance schedules, customer direct uploads, cataloging by F numbers, automated network solutions, minimize mean time to repair

- **Technical Solutions-** local / regional field service scheduling and dispatch to minimize response time, reliability and maintainability studies, preventive maintenance, robot evaluations, system optimization, enhancement services / upgrades, in-plant resident service engineer

Customer Service Hotline
1-800-47-ROBOT
(248) 377-7463

DNV Certification, Inc.
ISO 9001/QS-9000 REGISTERED FIRM

For more information, visit our website at: http://www.fanucrobotics.com/crc.asp

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.

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 as the primary safety element.
- 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 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.

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.

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 SCC 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.

- 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.

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.
- 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.

This manual describes the following products.

Name of product	Abbreviation
FANUC Robot R–2000*i*	R–2000*i*

Table of Contents

1. INTRODUCTION

This chapter explains the manual plan and the safety precautions that must be observed in working with the FANUC Robot.

☐ Contents of this chapter

1.1 Manual Plan

FANUC Robot series (R–J3*i* Mate CONTROLLER) LR HANDLING TOOL Operator's Manual.

This manual describes how to operate the FANUC Robot LR Mate 100*i* MODEL B an all–purpose compact robot. It is controlled by the FANUC R–J3*i* Mate controller (called the robot controller hereinafter) containing the LR Handling tool software.

This manual describes the following items for versatile mini robot:

- Setting the system for manipulating workpieces
- Operating the robot
- Creating and changing a program
- Executing a program
- Status indications
- Alarm codes and system variables

Using this manual

Each chapter of the manual describes a single operation of the robot. The user can select and read chapters describing required operations. The user can understand all the information presented in this the manual in five or six hours if he or she reads it from cover to cover once.

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

Identification

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

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

Specifications of products

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

Item to be checked	Section
Memory status	7.11 Memory Use Status Display
Software option list	A.1 List of Menus
Menu displayed when an option is selected	A.1 List of Menus
Program instruction that can be used when an option is selected	A.3 List of Program Instructions

Related manuals

The following manuals are available:

R-J3*i* Mate controller	OPERATOR'S MANUAL LR HANDLING TOOL B–81524EN	Intended readers: Operators responsible for designing, introducing, operating, and adjusting the robot system at the work site. Topics: Functions, operations and the procedure for operating the robot. Programming procedure, interface and alarm. Use: Guide to teaching, introducing, and adjusting the robot at the work site, and application designing.
	MAINTENANCE MANUAL B–81525EN	Topics: Installing and activating the system, connecting the mechanical unit to the peripheral device and maintenance the robot.
Mechanical unit	Maintenance manual B–81595EN	Topics: Installing and activating the robot, connecting the mechanical unit to the controller, maintaining the robot. Use: Guide to installation, activation, connection, and maintenance.

Notation

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

WARNING

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

CAUTION

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

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

Carefully read and save this manual.

1.2 Workers

A robot cannot do anything alone. The robot can operate only after it is equipped with a hand or other device and connected with peripheral equipment to form a system.
Give considerations for the safety of not only the robot but also the entire system. When using the robot, provide a safety fence and other safety measures. FANUC defines the system personnel as indicated below. Check which worker should be trained in a specialist robot course.

Operator

The jobs of an operator are:

- Turning on and off the system
- Starting and stopping programs of a robot
- Recovering the system from an alarm state

The operator must not enter the area enclosed by the safety fence to do his or her work.

Programmer or teaching operator

The jobs of the programmer or teaching operator include the jobs of the operator and the following:

- Teaching of a robot, adjustment of the peripheral equipment, and other work that must be done in the area enclosed by the safety fence

The programmer or teaching operator should be trained in a specialist robot course.

Maintenance engineer

The jobs of the maintenance engineer include the jobs of the programmer and the following:

- Repair and maintenance of the robot

The maintenance engineer should be trained in a specialist robot course.

1.3 General Safety Precautions

This section lists general safety precautions. Before starting to use the robot, read the precautions. The subsequent sections of the manual indicate other precautions. Take each of the precautions.

General rules

WARNING When the robot is used, the following precautions should be taken. Otherwise, the robot and peripheral equipment can be adversely affected, or workers can be severely injured.

– Avoid using the robot in a flammable environment.
– Avoid using the robot in an explosive environment.
– Avoid using the robot in an environment full of radiation.
– Avoid using the robot under water or at high humidities.
– Avoid using the robot to carry a person or animal.
– Avoid using the robot as a stepladder. (Never climb up on or hang from the robot.)

WARNING Robot personnel must wear the following safety articles:

– Clothing suitable for each job
– Safety shoes
– Helmet

NOTE Programmers and maintenance staff should be trained in a suitable course at FANUC.

Notes on installation

WARNING The robot should be transported and installed by accurately following the procedures recommended by FANUC. Wrong transportation or installation may cause the robot to fall, resulting in severe injury to workers.

CAUTION In the first operation of the robot after installation, the operation should be restricted to low speeds. Then, the speed should be gradually increased to check the operation of the robot.

Notes on operation

WARNING Before the robot is started, it should be checked that no one is in the area of the safety fence. At the same time, a check must be made to ensure that there is no risk of hazardous situations. If detected, such a situation should be eliminated before the operation.

CAUTION Operators should be ungloved while manipulating the operator’s panel or teach pendant. Operation with gloved fingers could cause an operation error.

NOTE Programs, system variables, and other information can be saved on memory card or floppy disks. Be sure to save the data periodically in case the data is lost in an accident. (See the file input/output section for saving the data.)

Notes on programming

WARNING Programming should be done outside the area of the safety fence as far as possible. If programming needs to be done in the area of the safety fence, the programmer should take the following precautions:

- Before entering the area of the safety fence, ensure that there is no risk of dangerous situations in the area.
- Be prepared to press the emergency stop button whenever necessary.
- Robot motions should be made at low speeds.
- Before starting programming, check the entire system status to ensure that no remote instruction to the peripheral equipment or motion would be dangerous to the user.

CAUTION After programming is completed, a text execution should be given according to a specified procedure. (See the section of program execution on this manual). During the text execution, workers must stay out of the safety fence.

NOTE Programmers should be trained in a suitable course at FANUC.

Notes on maintenance

WARNING During maintenance, the robot and system should be in the power–off state. If the robot or system is in the power–on state, a maintenance operation could cause a shock hazard. If necessary, a lock should be provided to prevent any other person from turning on the robot or system. If maintenance needs to be executed in the power–on state, the emergency stop button should be pressed.

WARNING When replacing a part, the maintenance worker should read the maintenance manual and learn the replacement procedure beforehand. If a wrong procedure is followed, an accident may occur, causing damage to the robot and injury to the worker.

WARNING When entering the area enclosed by the safety fence, the maintenance worker should check the entire system to make sure that no dangerous situations are present. If the worker needs to enter the area of the fence while a dangerous situation exists, the worker should always take extreme care and check the current system status.

WARNING A part should be replaced with a part recommended by FANUC. If other parts are used, malfunction or damage would occur. Especially, a fuse that is not recommended by FANUC should not be used. Such a fuse may cause a fire.

WARNING When a motor or brake is removed, the robot arm should be supported with a crane or other equipment beforehand so that the arm would not fall during the removal.

WARNING If a robot motion is necessary during maintenance, the following precautions should be taken:

- Reserve an escape route. During the maintenance, always check the motions of the whole system so that the escape route will not be blocked by the robot or peripheral equipment.
- Always pay attention to risk of dangerous situations and get prepared to press the emergency stop button whenever necessary.

WARNING When a motor, decelerator, or other heavy load is handled, a crane or other equipment should be used to protect maintenance workers from excessive load. Otherwise, the maintenance workers would be severely injured.

CAUTION Whenever grease is spilled on the floor, it should be removed as quickly as possible to prevent dangerous falls.

CAUTION The robot should not be stepped on or climbed up during maintenance. If it is attempted, the robot would be adversely affected. In addition, a misstep can cause injury to the worker.

CAUTION The following parts are heated. If a maintenance worker needs to touch such a part in the heated state, the worker should wear heat–resistant gloves or use other protective tools.

- Servo motor
- Inside the control unit

CAUTION When a part is replaced, all bolts and other related components should put back into their original places. A careful check must be given to ensure that no components are missing or left unmounted.

CAUTION Before the maintenance of the pneumatic system is started, the supply pressure should be shut off and the pressure in the piping should be reduced to zero.

CAUTION After a part is replaced, a text execution should be given for the robot according to a predetermined method. (See the program execution of this manual.) During the text execution, the maintenance staff should work outside the safety fence.

CAUTION After the maintenance is completed, spilled oil or water and metal chips should be removed from the floor around the robot and within the safety fence.

CAUTION When a part is replaced, care must be taken to prevent dust from entering the robot.

NOTE Each maintenance worker or inspection worker should be trained in a suitable course at FANUC.

NOTE Maintenance should be done under suitable light. Care must be taken that the light would not cause any danger.

NOTE The robot should be periodically inspected. (Refer to the maintenance manual.) A failure to do the periodical inspection can adversely affect the performance or service life of the robot and also may cause an accident.

1.4 Safety Precautions

Safety precautions

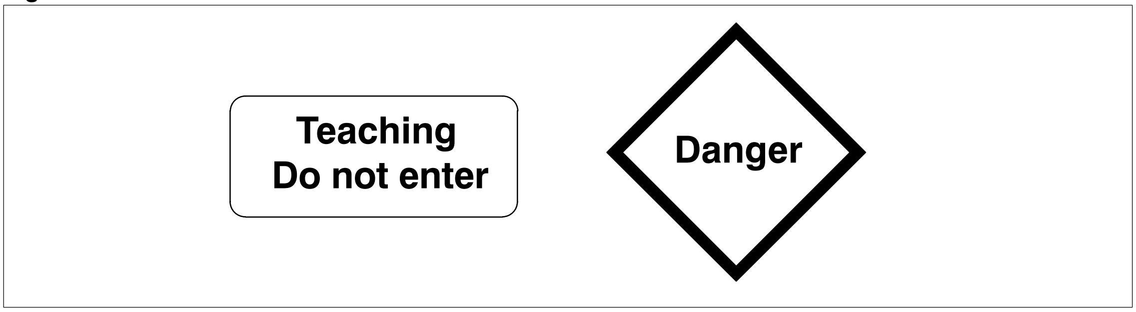
Unlike ordinary automatic machines, robots have arms and wrists which can be moved in all operation space. A robot is quite flexible, but on the other hand, it is quite dangerous. The robot is usually connected with peripheral equipment to comprise an automated system. Users must take safety precautions for the entire system.

The safety precautions are described below.

Safety precautions related to installation and layout

- Use warning lamps and other provisions to indicate that the robot is operating.

Figure 1–1. Alarm Indications

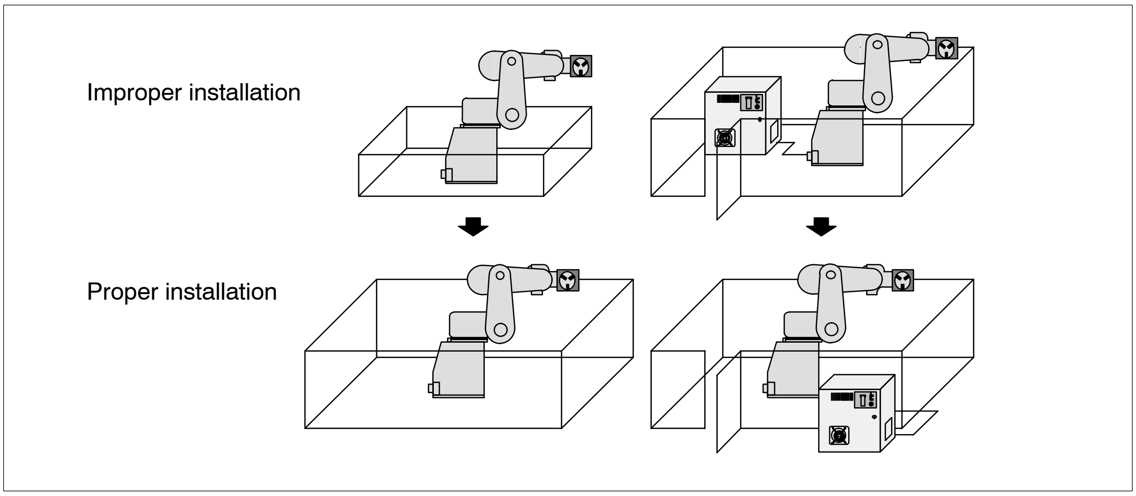


- Put a protective fence with safety door around the system so that only the operator can enter the operating area by the door. Design the system so that it will stop when the door is opened.

NOTE Connect the *FENCE input signal to the safety door. Refer to the maintenance manual for explanations about how to connect.

NOTE When the *SFSPD (safety speed) input signal is turned off, the control unit halts the robot.

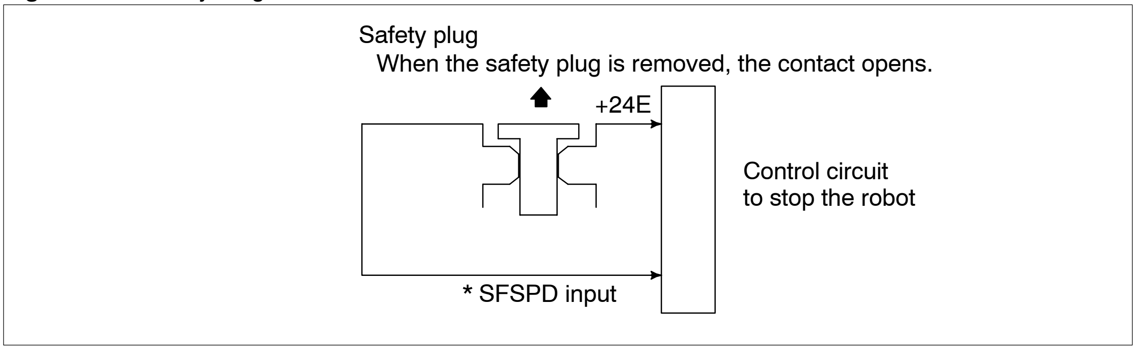
- Put a protective fence so that the motion range of the robot is surrounded completely. Moreover, put the controller outside of the protective fence.



- Install an emergency stop button where it will be readily accessible to the operator.

NOTE Upon receiving an emergency stop signal, the controller immediately stops the robot.

Figure 1–2. Safety Plug



Safety precautions related to system design

- Install a safety joint between robot wrists. If an abnormal external force is applied to the robot, the safety joint breaks and the robot stops.

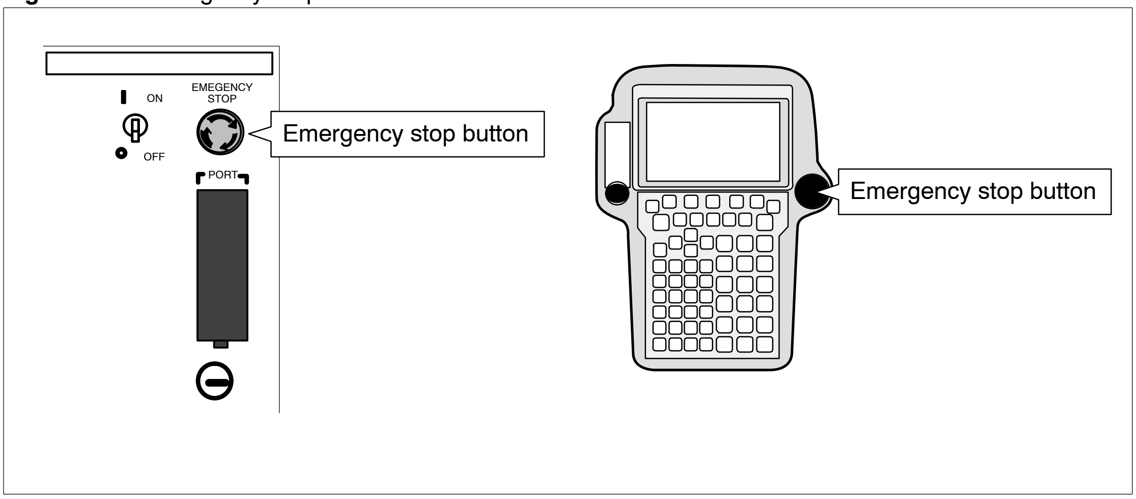
NOTE When the hand break (*HBK) input signal goes off, the controller immediately stops the robot.

- Hand breakage detection can be disabled when the *HBK input signal is off. This can be set on the system setting screen. See the section of the system config menu.
- Ground all peripheral units properly.
- When a desired operating area is smaller than the maximum operating area of the robot, the desired area can be specified by parameters.
- The robot receives interlock signals sent from remote equipment. Upon receiving a signal indicating the operating status of the remote equipment, the robot can stop or halt.

Safety precautions related to inspection and maintenance

- Before starting the inspection or maintenance, turn off the controller. Place a guard to prevent someone else from switching the power on.
- Before disconnecting the pneumatic system, release the supply pressure.
- Before starting an inspection in which the electrical system of the robot need not be operated, press the emergency stop button.
- When carrying out an inspection in which the robot needs to be operated, carefully observe the motion of the robot. Immediately press the emergency stop button whenever required.

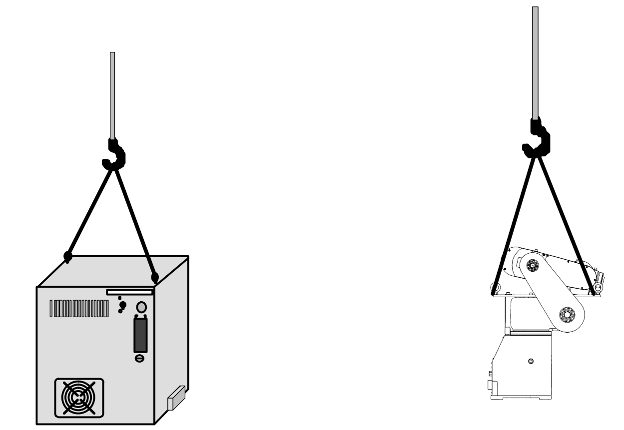
Figure 1–3. Emergency Stop Button



Safety precautions related to transportation

- When carrying the robot or another unit on a carrier such as a crane or fork lift, securely fasten the robot to the carrier.
- Carefully inspect the crane, fork lift, other carrying equipment, and carrying handle on the product.

Figure 1–4. Carrying the Robot



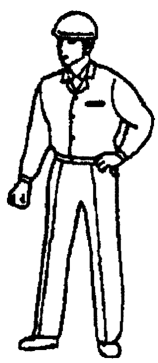
Safety precautions related to operation

- All robot system operators are requested to attend FANUC training courses to learn the safety precautions and functions of the robot.
- Before beginning to program the robot, make sure that there are no abnormal or dangerous conditions around the robot and peripheral equipment.
- Before working within the operating area of the robot, even when the robot is not running, switch the power off or press the emergency stop button. Place a guard to prevent someone else from entering the operating area of the robot or activating the robot from the operator's panel.

Table 1–1. Safety precautions item

Operator	Workshop	Transportation and installation
Avoid dangerous behavior. Wear working clothes, safety shoes, and a safety helmet.	Keep the workshop neat, tidy, and clean. Install a protective fence and warning indications. Provide ventilation. Never bring flammable material to the workshop.	Keep the transportation lane free from obstacles. When carrying the robot or another unit on a carrier such as a fork lift or crane, securely fasten it to the carrier. Keep a sufficient operating area. Make connection s properly.
Operation	**Maintenance and inspection**	**Hand**
Attend training classes. Master the operating procedures. Exclude unauthorized personnel.	Use only FANUC products for repair. Before starting maintenance or inspection, turn the power off. Close the controller door.	Inspect and take care of cables. Check the pneumatic pressure. Inspect the hand mechanism.

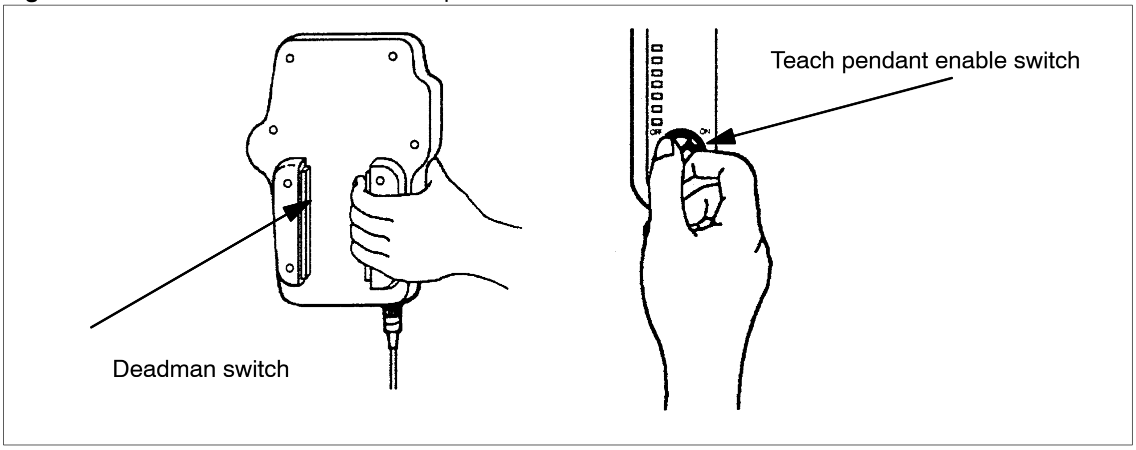
Figure 1–5. Safety Clothes and Safety Helmet



- Before approaching the robot to program it, hold the teach pendant in your hand, press the deadman switch, and set the teach pendant enable switch on.

NOTE If the deadman switch is released while the teach pendant enable switch is on, the robot immediately stops.

Figure 1–6. Deadman switch and Teach pendant enable switch



- Before moving the robot by jog feed, carefully observe the operation of the jog keys and the robot.
- Before moving the robot by jog feed, sufficiently lower the feedrate override of the robot.

2. OVERVIEW

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

☐ Contents of this chapter

A FANUC robot system consists of the tool software for manipulating workpieces, the mechanical unit of the robot itself (FANUC Robot LR Mate 100*i* MODEL B), and the Robot control unit (FANUC SYSTEM R–J3*i* Mate).

The robot system offers outstanding performance when manipulating automobile parts and other products.

Tool software for manipulating workpieces

The tool software for manipulating workpieces is an application software package installed on the Robot control unit. Any work can be performed by specifying menus and instructions from the teach pendant. The tool software for manipulating workpieces contains instructions for controlling the robot, hands, remote control units, and other peripheral devices.

The I/O between an additional axis or control unit and another peripheral device can be controlled using the tool software for manipulating workpieces. Other peripheral devices include cell controllers, floppy disk drives, printers, or vision sensors.

Robot

Robot has a hand or another end effector interface for control to do work. The LR Mate 100*i* MODEL B is ideal for manipulating workpieces.

Controller

The Robot control unit supplies power to drive the mechanical unit.

The controller supplies the power for operating the mechanical unit.

The controller of the robot LR Mate 100*i* MODEL B is available in panel–mount type, which can be mounted to a machine tool, and the stand–alone type (FANUC SYSTEM R–J3*i* Mate), which has a control section housed in a cabinet.(These are hereinafter referred to as the R–J3*i* Mate controller.) Both types have identical functions for controlling the robot.

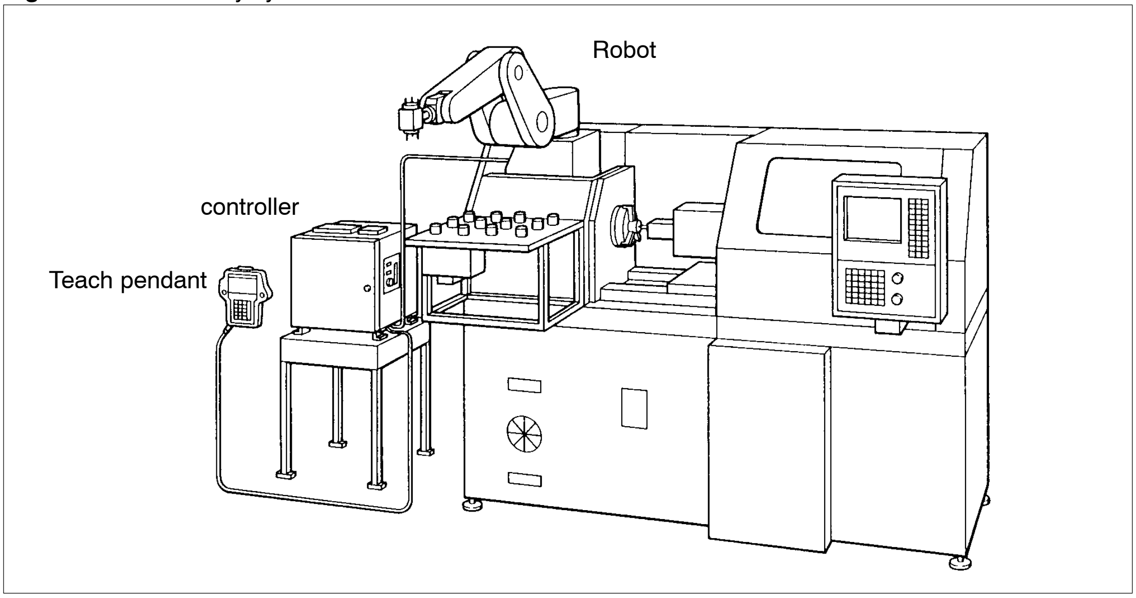
The tool software for manipulating workpieces is installed on the Robot control unit to control the teach pendant, operator's panel, and external peripheral devices.

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

- The remote control units are used to control the Robot control unit.
- The hands, floppy disk drive, printer, and other devices are operated using I/O and serial communication units.

Fig. 2–1 shows a typical robot system for manipulating workpieces. The system consists of a robot, the Robot control unit, and peripheral devices.

Figure 2–1. Assembly system for car doors



2.1 LR Handling Tool Software

The LR Handling tool software has been specially designed to perform material handling operations. The LR Handling tool software is contained in the Robot and enables the following:

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

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

2.1.1 System setting

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

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

2.1.2 Jog feed of the robot

Jog feed of the robot is the operation of moving the robot as desired by manually entering commands on the teach pendant. When a motion instruction of a program is executed, the robot is moved to the target position by jog feed, then the position is recorded. (For the jog feed of the robot, see 5.)

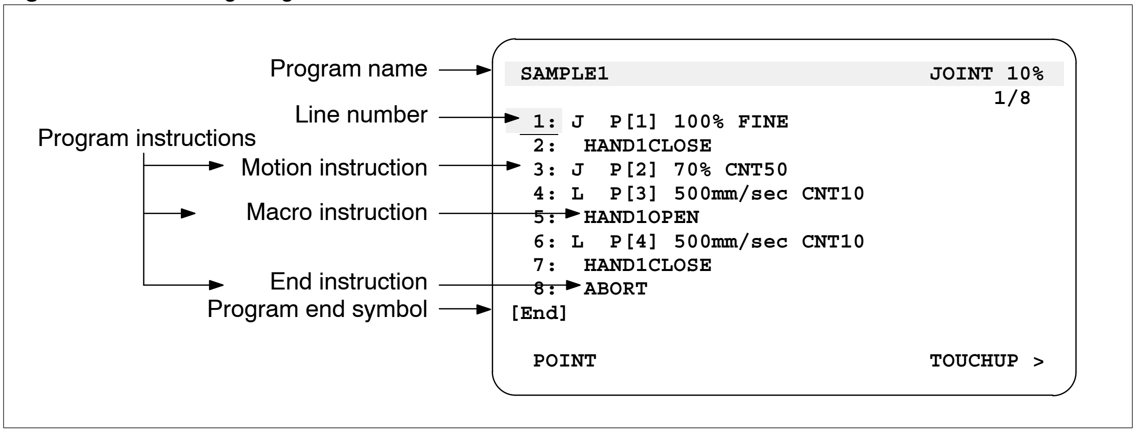
2.1.3 Program

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

The Spot teach pendant is used to create or correct a program. (For creation of a program, see Chapter 5.) The program contains the following instructions. Figure 2–3 shows a basic program for manipulating workpieces.

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

Figure 2–2. Handling Program



2.1.4 Test operation (test execution)

After the system is set and a program is created, perform the test operation in the test execution mode to check the program for normal operation. (For the test operation, see Sections 6.2 and 6.3.)

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

2.1.5 Automatic operation (operation execution)

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

- Specified programs are started one after another. (For automatic operation, see Sections 3.7 and 6.6.)
- During automatic operation, position data can be corrected (online position correction Section 6.7).
- The processing is halted, then aborted or resumed. (For halting a program, see Section 6.1.)

2.2 Robot

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

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

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

Figure 2–3. Main axes and wrist axes

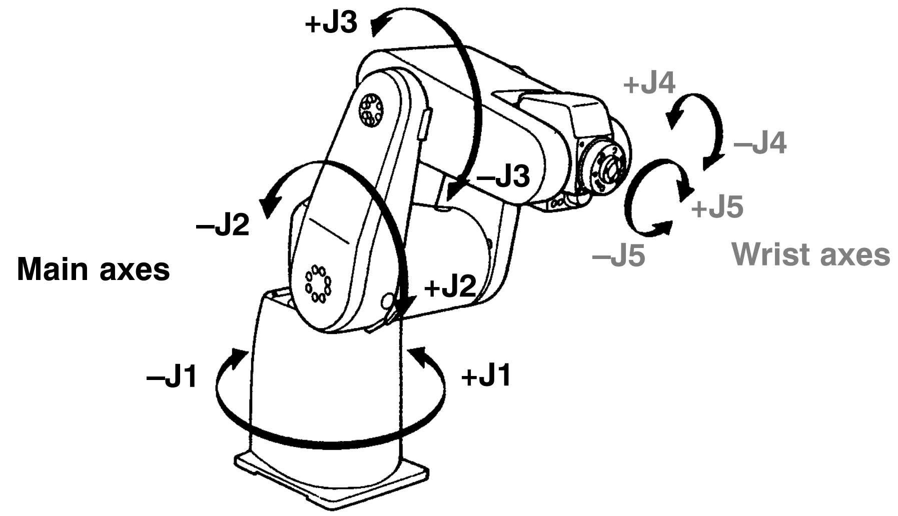


Figure 2–4. Hand with Fingers

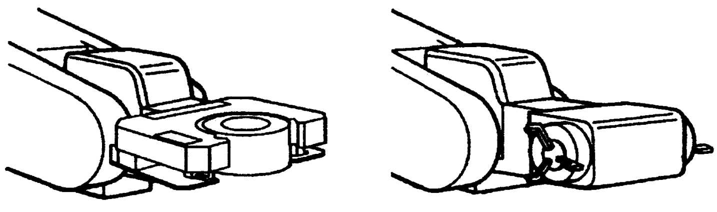
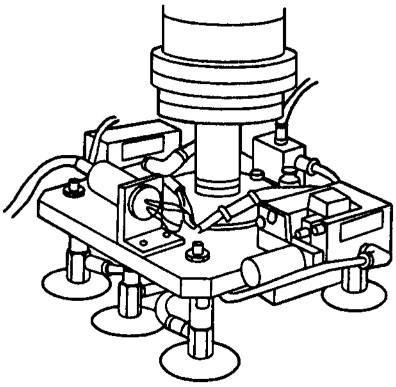


Figure 2–5. Hand with Suction Cups and No Fingers



2.3 Controller

R–J3*i* Mate controller includes a power unit, user interface circuit, motion controlling circuit, memory circuit, and input/output circuit.

The controller of robot LR Mate 100*i* MODEL B is available in two types: panel mount type, which can be mounted to a machine tool, and the stand–alone type (FANUC SYSTEM R–J3*i* Mate). (These are referred to as the R–J3*i* Mate controller.) These types have identical functions for controlling the robot.

The stand–alone type R–J3*i* Mate, however, has additional switches and connectors.

- The R–J3*i* Mate has a circuit protector switch for turning the power on and off and an emergency stop button.
- The R–J3*i* Mate has a communication port.

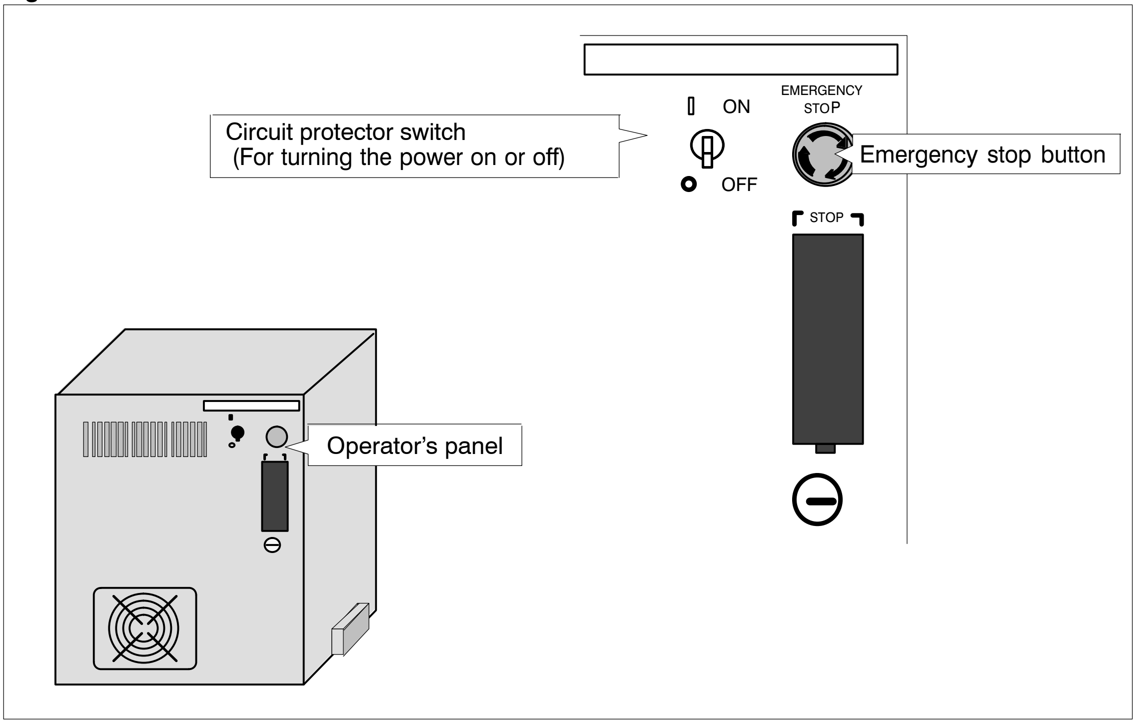
The user should use a teach pendant and operator's box to operate the control unit.

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

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

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

Figure 2–6. R–J3*i* Mate controller



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

Disconnecting the teach pendant

Disconnect the teach pendant cable from the front panel of the R–J3*i* Mate controller, as follows:

1. Set the connect/disconnect switch to DISCONNECT.
2. Grasp the connector and carefully pull it towards you.

Connecting the teach pendant

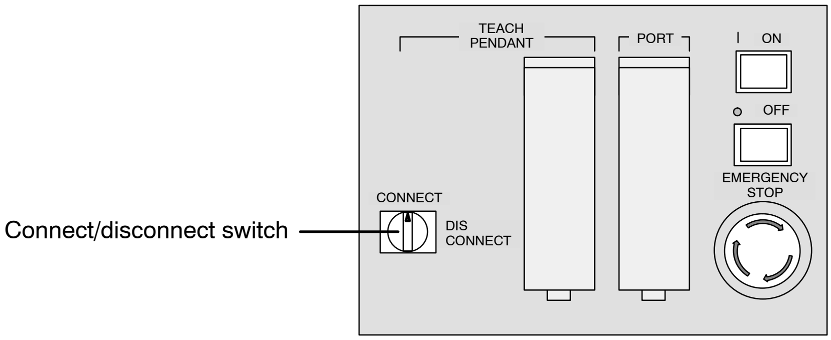
Connect the teach pendant cable to the front panel of the R–J3*i* Mate controller, as follows:

1. Align the cable and panel connectors, then carefully push the cable connector fully home.
2. Set the connect/disconnect switch to CONNECT.

> **CAUTION**
> Failure to observe the above connection/disconnection procedures may result in damage to the teach pendant or controller.

NOTE The teach pendant must always be connected when the system is first started.

Figure 2–7. Connect/disconnect switch



2.3.1 Teach pendant

The teach pendant provides an interface between the LR Handling tool software and the operator. The teach pendant is connected to the PC board in the controller by a cable.

The following operations can be performed using the teach pendant:

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

The teach pendant includes the following:

- Liquid crystal display of 40 characters by 16 lines
- 11 LEDs
- 61 keys (Two keys are for LR Handling tool.)

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

The following switches are also provided:

Teach pendant enable switch	This switch enables or disables the teach pendant. When the teach pendant is disabled, a jog feed, program generation, or test execution cannot be carried out.
Deadman switch	DEADMAN SWITCH is used as an enabling device. When the teach pendant is enabled, this switch allows robot motion only while the deadman switch is gripped. If you release this switch, the robot stops immediately.
Emergency stop button	When pressed, the emergency stop button immediately stops the robot.

Figure 2–8. Switches on the Teach Pendant

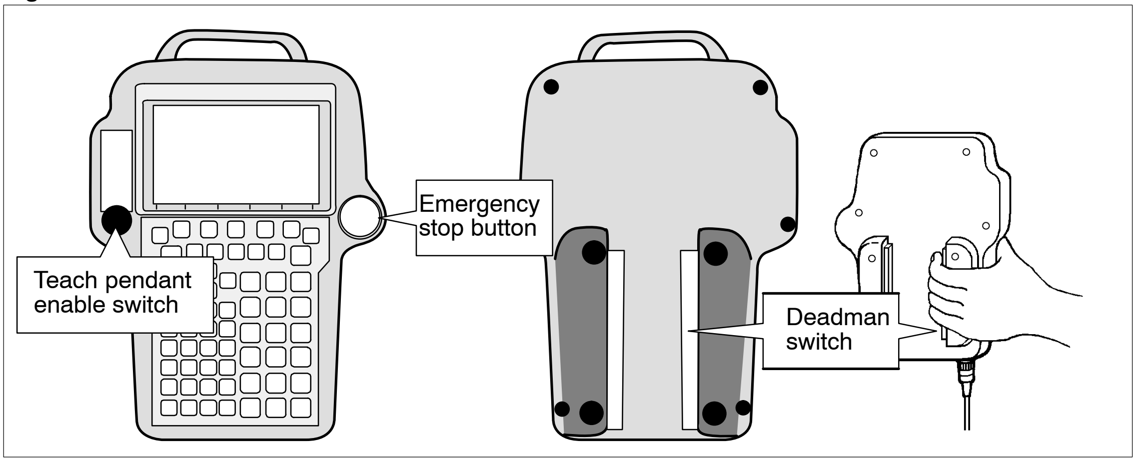
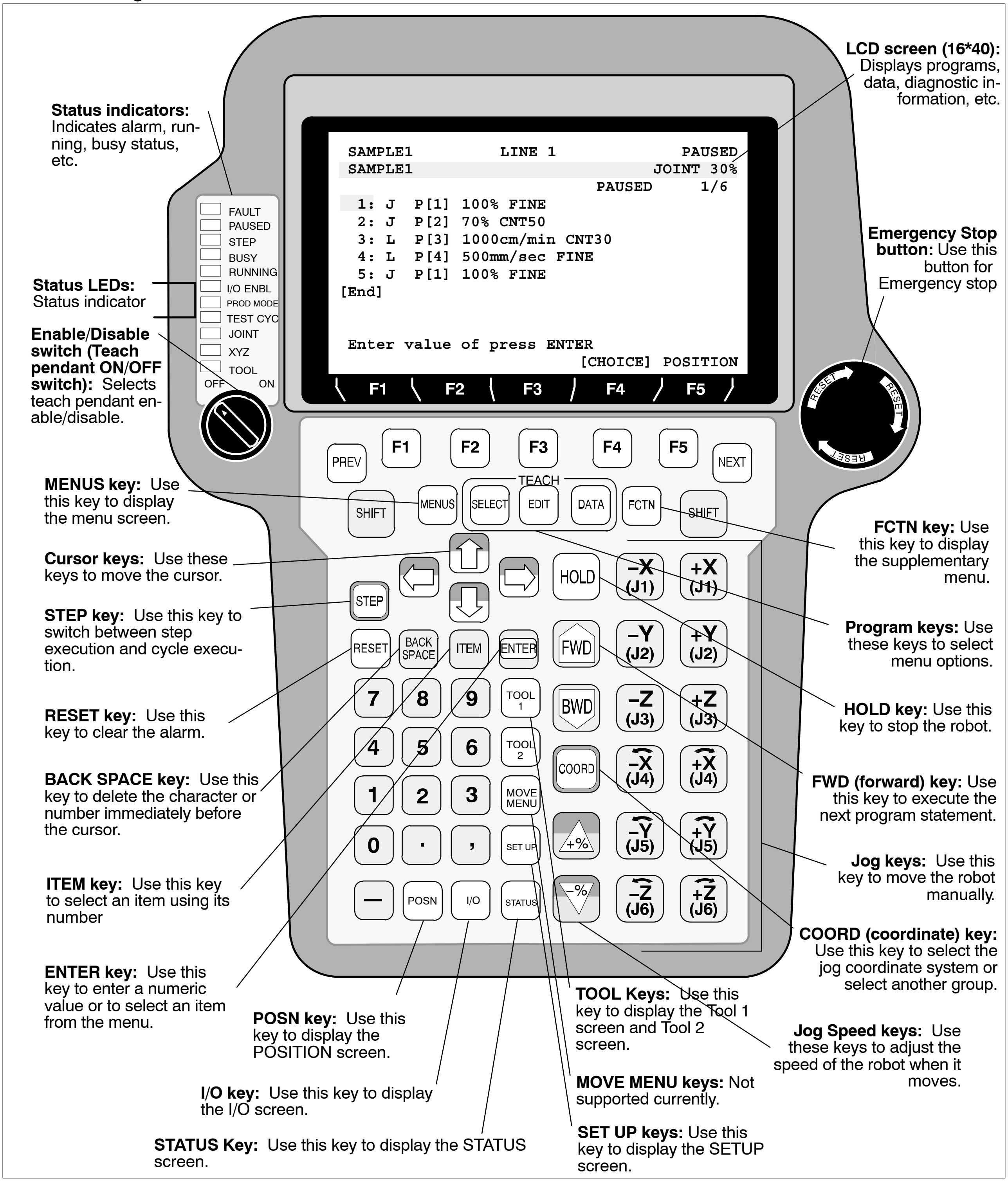


Figure 2–9 shows the teach pendant.

Figure 2–9. Teach Pendant



Keys on the teach pendant

The teach pendant has the following keys:

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

Table 2–1. Keys related to menus

Key	Function
F1 F2 F3 F4 F5	The function (F) key to select a function menu at the last line of the screen.
NEXT	The next page key to switch the function key menu on the next page.
MENUS FCTN	The MENUS key to display the screen menu. The FCTN key to display the function menu.
SELECT EDIT DATA	The SELECT key to display the program selection screen. The EDIT key to display the program edit screen. The DATA key to display the program data screen.
MOVE MENU SETUP	The MOVE MENU key moves the robot to the reference position. Create a program which moves the robot to the reference position and assign this program to a macro instruction so that this can be started by this MOVE MENU key. The SET UP key displays the setup screen.
STATUS I/O POSN	The STATUS key displays the current position screen. The I/O key displays the I/O screen. The POSN key displays the current position screen.

Table 2–2. Keys related to jog feed

Key	Function
SHIFT	The SHIFT key is used to execute a jog feed of the robot, teach the position data, and start a program. The right and left Shift keys have the same function.
–Z (J3) –Y (J2) –X (J1) +Z (J3) +Y (J2) +X (J1) –Z (J6) –Y (J5) –X (J4) +Z (J6) +Y (J5) +X (J4)	The jog keys are effective while a Shift key is held down. They are used for jog feed.

Table 2–2. (Cont'd) Keys related to jog feed

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

Table 2–3. Keys related to execution

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

Table 2–4. Keys related to editing

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

Table 2–5. Keys related to handling

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

LEDs on the teach pendant

Figure 2–10. LEDs on the teach pendant

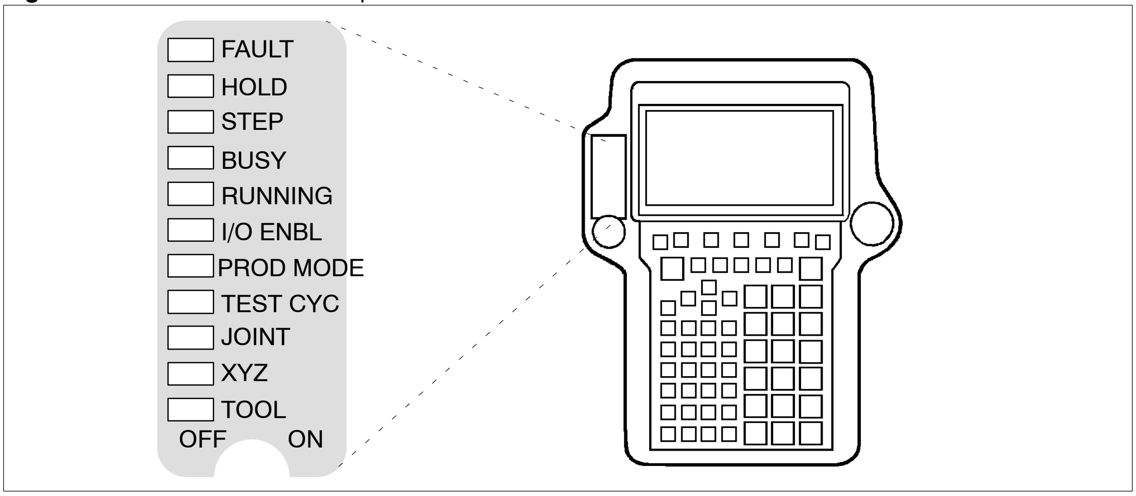


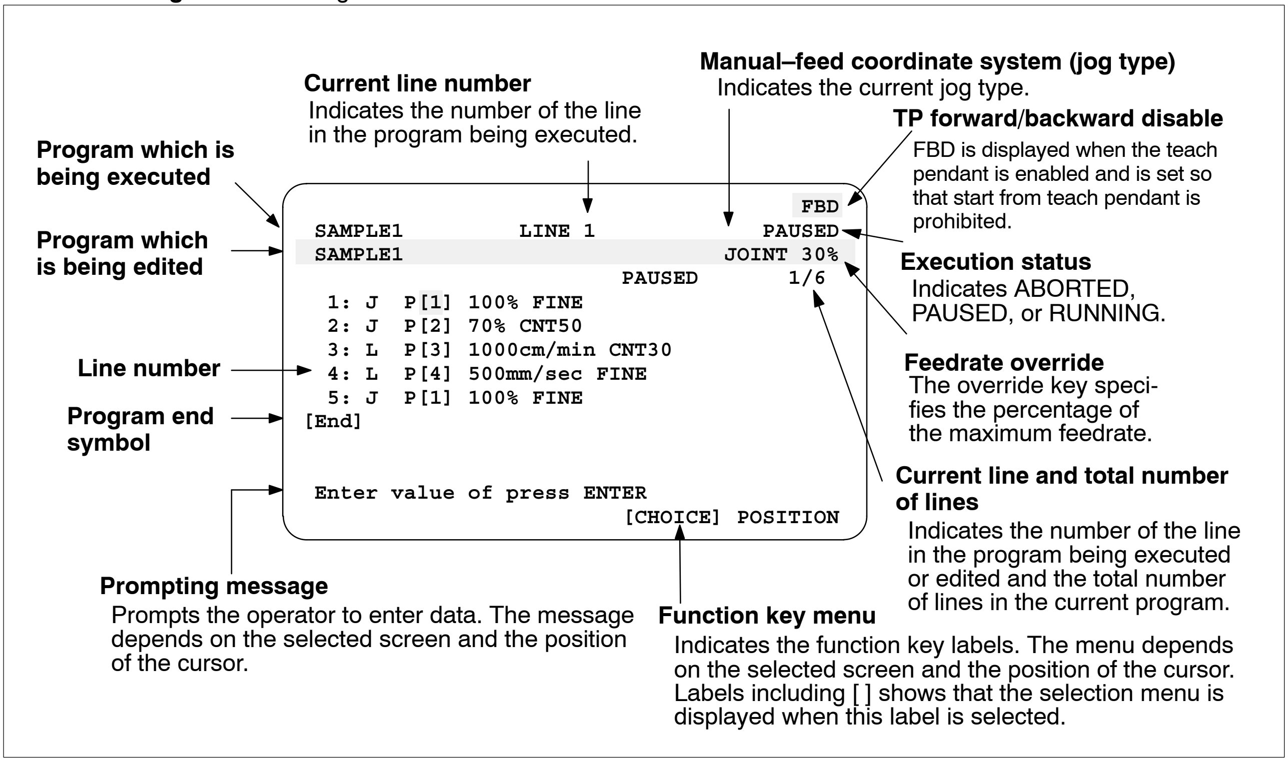
Table 2–6. LEDs on the teach pendant

LED	Function
FAULT	The FAULT LED indicates that an alarm has occurred.
HOLD	The HOLD LED indicates that the HOLD button is being pressed or HOLD signal is being input
STEP	The STEP LED indicates that it is under step operation mode.
BUSY	The BUSY LED is lit while the robot is working. It is also lit when a program is executed or when the printer or floppy disk drive unit is operating.
RUNNING	The RUNNING LED indicates that the program is being executed.
JOINT	The JOINT LED is lit when joint jog is selected as the manual–feed coordinate system (jog type).
XYZ	The XYZ LED is lit when Cartesian jog (JGFRM World frame or USER) is selected as the manual–feed coordinate system (jog type).
TOOL	The TOOL LED is lit when tool jog (TOOL) is selected as the manual–feed coordinate system (jog type).

2.3.1.1 Display screen of the teach pendant

The liquid crystal display screen (liquid crystal display) displays the Handling tool software screen shown in Figure 2–11. To operate the robot, select a screen corresponding to a desired function. The screen is selected by the screen menus shown in Figure 2–12.

Figure 2–11. Program Edit Screen



Screen menu and function menu

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

Figure 2–12, Figure 2–13, and Figure 2–14 show the screen menu, auxiliary menu, and quick menu respectively.

Screen menu

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

Figure 2–12. Screen menu

MENUS

```
1 UTILITIES
2 TEST CYCLE
3 MANUAL FCTNS
4 ALARM
5 I/O
6 SETUP
7 FILE
8
9 USER
0 --NEXT--
```

Page 1

```
1 SELECT
2 EDIT
3 DATA
4 STATUS
5 POSITION
6 SYSTEM
7
8
9
0 --NEXT--
```

Page 2

Table 2–7. Screen menu

LED	Function
UTILITIES	The utility screen is used to display the hints'.
TEST CYCLE	The test cycle screen is used to specify the data for test operation.
MANUAL FCTNS	The manual operation screen is used to execute macro instructions.
ALARM	The alarm history screen shows the history and details of alarms.
I/O	The I/O screen is used to display and set manual output, simulated input/output, and assign of signals.
SETUP	The setting screen is used to set the system.
FILE	The file screen is used to read or store files.
USER	The user screen shows user messages.
SELECT	The program selection screen is used to list or create the programs.
EDIT	The program edit screen is used to correct and execute a program.
DATA	The program data screen shows the values in registers, position registers, and pallet register.
STATUS	The status screen shows the system status.
POSITION	The current position screen shows the current position of the robot.
SYSTEM	The system screen is used to set system variables and mastering.

Function menu

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

Figure 2–13. Function menu

FCTN

```
1 ABORT (ALL)
2 Disable FWD/BWD
3 CHANGE GROUP
4 TOGGLE SUB GROUP
5 TOGGLE WRIST JOG
6
7 RELEASE WAIT
8
9
0 -- NEXT --
```

Page 1

```
1 QUICK/FULL MENUS
2 SAVE
3 PRINT SCREEN
4 PRINT
5
6
7
8
9
0 -- NEXT --
```

Page 2

Table 2–8. Function menu

LED	Function
ABORT (ALL)	ABORT forces a program which is being executed or temporarily halted to terminate.
Disable FWD/BWD	Disable FWD/BWD enables or disables starting a program with a teach pendant
CHANGE GROUP	Changes the operation group for jog feed. Displayed only when multiple groups are set.
TOGGLE SUB GROUP	TOGGLE SUB GROUP toggles jog between robot standard axes and extended axes.
TOGGLE WRIST JOG	TOGGLE WRIST JOG toggles jog between the attitude control feed and the wrist joint feed which does not maintain the wrist attitude in linear feed.
RELEASE WAIT	Skips the wait instruction currently being executed. When the wait state is released, execution of the program stops temporarily at the line subsequent to the wait instruction.
QUICK/FULL MENUS	QUICK/FULL MENUS toggles the menu between a usual screen menu and a quick menu.
SAVE	SAVE saves the data related to the current screen on a floppy disk or memory card.
PRINT SCREEN	PRINT SCREEN prints the data displayed on the current screen.
PRINT	PRINT prints the data on the current screen exactly.

Quick menu

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

- ALARM / alarm occurrence and alarm history screen
- UTILITIES / hint screen
- Setup screen
- DATA / register screen
- STATUS/jig screen
- I/O/digital/group/Robot I/O screen
- POSITION screen

Figure 2–14. Quick menu

```
1 ALARM
2 UTILITIES
3 Setup
4 DATA
5 STATUS
6 I/O
7 POSITION
8
9
0 -- NEXT --
```

NOTE The program selection screen can be displayed by the SELECT key. But the only available function is selecting a program.

NOTE The program edit screen can be displayed by the EDIT key. But the only available functions are changing position and speed values.

2.3.2 Operator panel

The operator panel has buttons, switches, and connectors.

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

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

The operator panel also has an RS–232C communication port.

Table 2–9 lists the switches on the operator's panel.

Table 2–9. Switches on the Operator Panel

Switch	Function
Power–on/off button Circuit–protector	Turns on and off the power to the Robot control unit.
Emergency stop button	Press this button to stop the robot immediately. Turn the emergency stop button clockwise to release it.
Start button (option)	Starts the currently selected program. Lit while the program is being started.
Three mode switch (option)	Enables the user to select operation mode suitable to the robot operation conditions or the status of its use.

2.3.3 Remote controller

Remote control units are external devices connected to the Robot control unit to configure a system.

These are control units for controlling the operation of the system created by the user using peripheral devices and I/O provided by the robot control unit.

2.3.4 CRT/KB

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

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

2.3.5 Communication

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

- One standard RS–232–C port

2.3.6 Input/output

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

The input/output signals include the following:

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

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

I/O unit model A, I/O unit model B, and Process I/O PC board can be connected to the controller.

2.3.7 Peripheral I/O

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

Peripheral I/O signals perform the following:

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

2.3.8 Motion of the robot

A single motion instruction specifies a motion of the robot, or a movement of the tool center point (TCP) from the current position to the target position. The Robot uses a motion control system that comprehensively controls the tool path, acceleration/deceleration, positioning, feedrate, and other factors.

The R–J3*i* Mate control unit can control up to eight axes, divided into up to three operation groups (multiple motion function). The control unit can control up to eight axes for a group. The operation groups are independent of one another, but can be synchronized to operate the robot simultaneously.

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

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

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

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

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

2.3.9 Emergency Stop devices

This robot has following emergency stop devices.

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

When an emergency stop button is pushed, the robot stops immediately in any cases. The external emergency stop outputs or inputs the emergency stop signal for peripheral devices (e.g. safety fence, gate). The signal terminal is on the controller and operator's box inside.

2.3.10 Extended axis

A maximum of three axes of one group can be added to the standard axes (usually five axes) of the robot. The Robot can control up to 8 axes. The extended axis has the following two types:

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

3. SETTING UP THE HANDLING SYSTEM

The LR Handling tool application can be used after required data is specified. This chapter describes the data that can be specified.

3.1 I/O

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

General–purpose I/O

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

This group includes the following signals:

- Digital I/O: SDI[i]/SDO[i]
- Group I/O: GI[i]/GO[i]
- Analog I/O: AI[i]/AO[i]

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

Specialized I/O

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

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

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

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

Configuring I/O

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

Rack

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

– 0 = Process I/O PC board

– 1 to 16 = I/O Unit–MODEL A / B

SLOT

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

- When the process I/O PC board is used, the first connected board is SLOT 1, the second is SLOT 2 and others are numbered sequentially as this.
- When the I/O Unit–MODEL A or B is used, SLOT is the number identifying the connected module.

Configuration of the I/O Link

The I/O Link consists of the following hardware.

- Printed circuit board for controlling the robot
- FANUC I/O Unit–MODEL A (I/O Unit–MODEL A)
- FANUC I/O Unit–MODEL B (I/O Unit–MODEL B)
- Process I/O printed board

I/O link data should be set at CNC side when R–J3*i* Mate is connected to CNC. Refer to the operator's manual of CNC for details.

In Figure 3–1, R–J3*i* Mate and I/O Unit–MODEL A are connected to CNC as slave of I/O link.

In Figure 3–2, the process I/O PC board and the I/O Unit–MODEL A are connected to the robot control PC board as the slave of I/O link. At this time, the robot controller board is a master of I/O link against these I/O modules.

Since JD4 and JD1A is the same connector, these can not be used at the same time. To specify one of these, set a system variable, $IOMASTER, as the following.

– TRUE : The robot control PC board is master. (JD4)
– FALSE : The robot control PC board is slave. (JD1A)

Moreover, the default setting is FALSE.

Figure 3–1. Input/Output Module Configuration (When R–J3*i* Mate is slave)

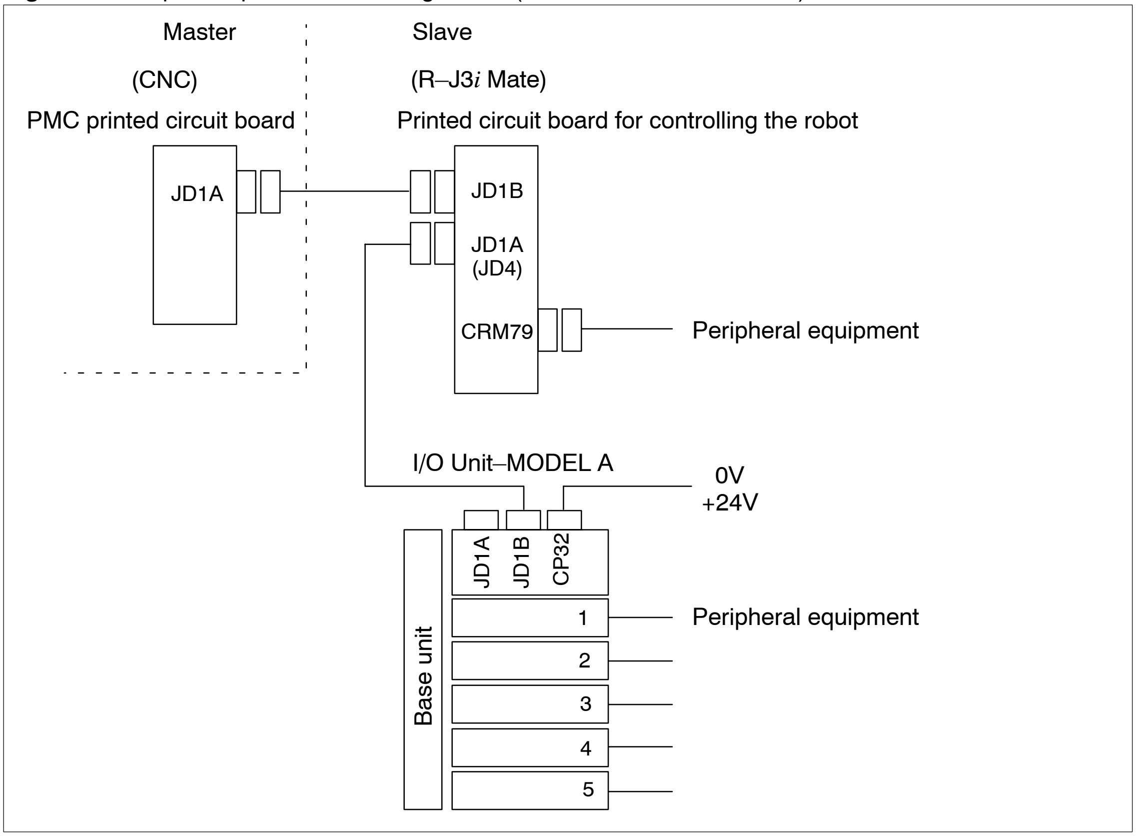
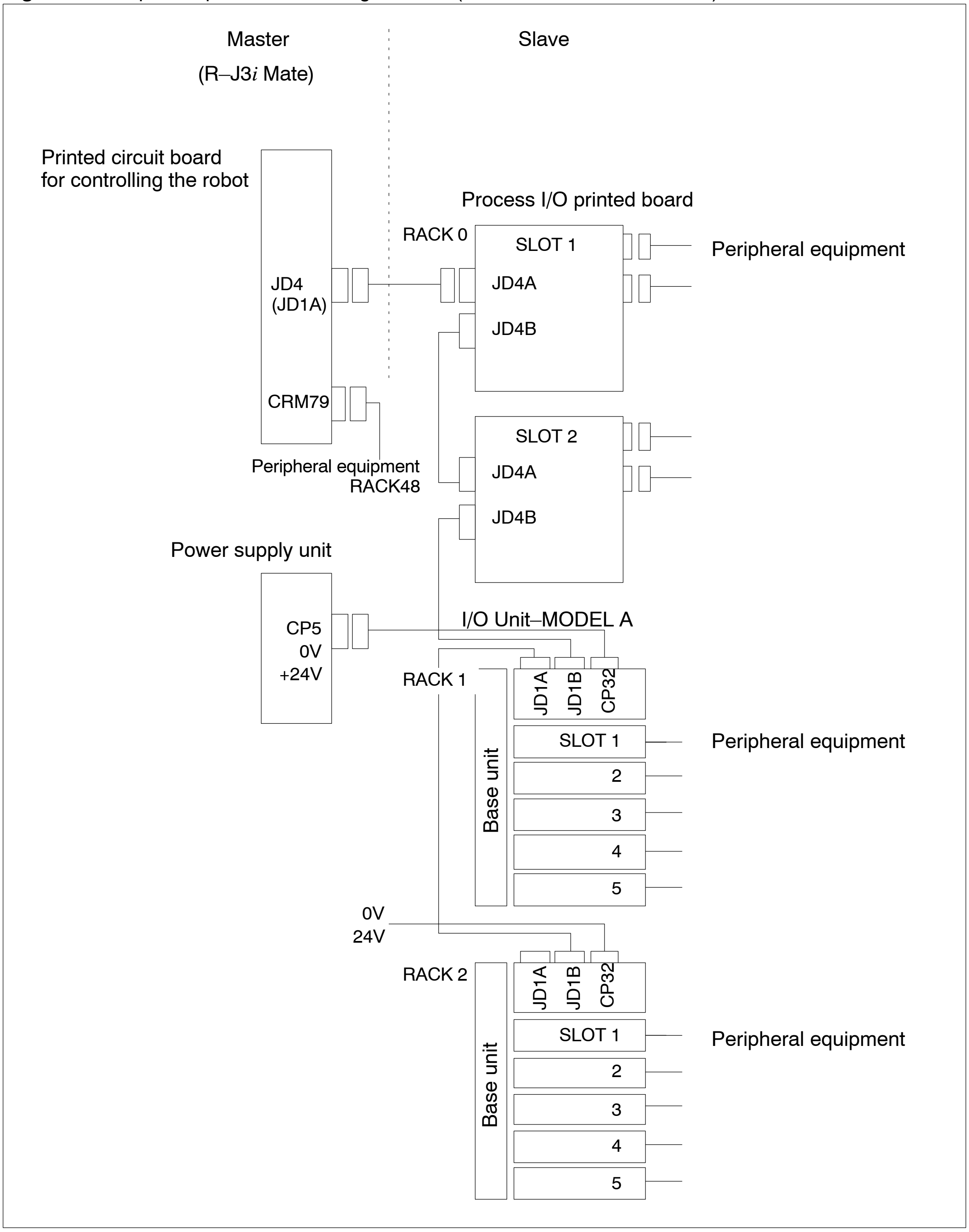


Figure 3–2. Input/Output Module Configuration 2 (When R–J3*i* Mate is master)



Process I/O PC board

As for Input/Output signal lines on the process I/O PC board, when the peripheral I/O is allocated to the process I/O PC board, 18 input and 20 output signals are allocated in the peripheral I/O. (See Section 3.3 ”Peripheral I/O”) I/O signal lines except the peripheral I/O are allocated in digital I/O and group I/O (See Section 3.1.1, ”Digital I/O” and Section 3.1.2 ”Group I/O”).

NOTE The first four signal lines on the process I/O printed circuit board are fixed to 24 V common.

Figure 3–3. Process I/O PC board

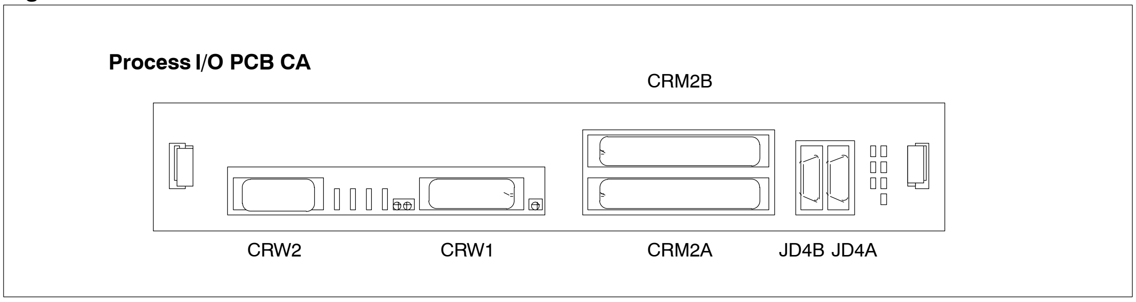
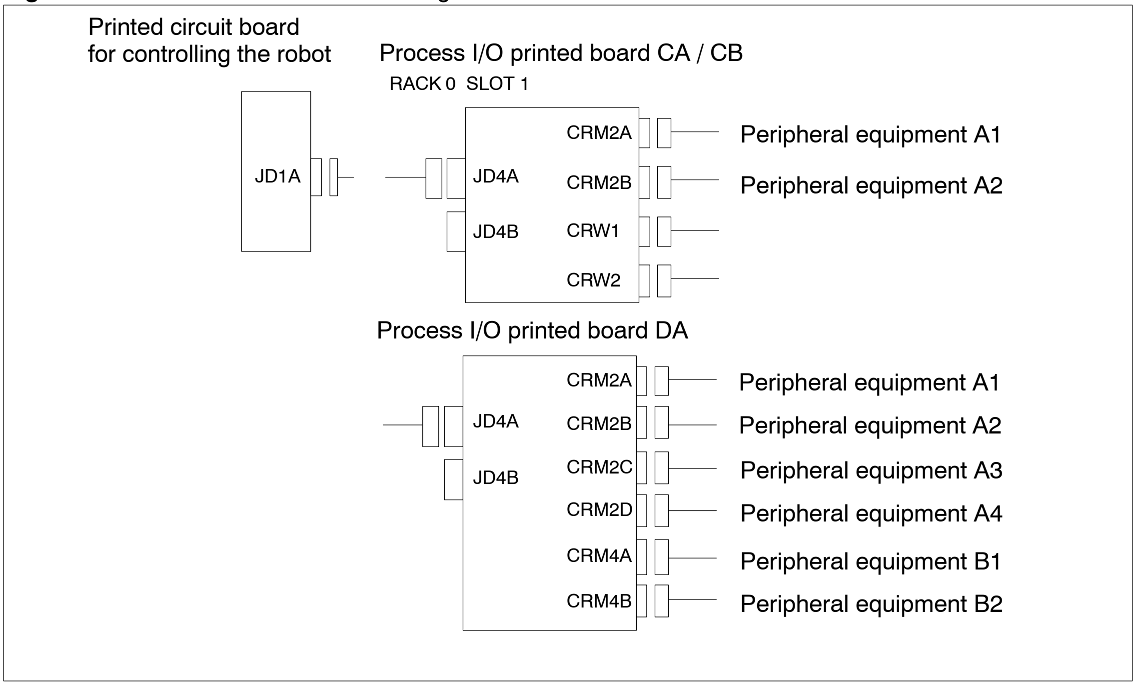


Figure 3–4. Process I/O PC board Configuration



For details of process I/O PC board, refer to MAINTENANCE MANUAL.

Figure 3–5. Process I/O PC board interface

Peripheral equipment A1

CRM2A

01	in 1	19	out 13	33	out 1
02	in 2	20	out 14	34	out 2
03	in 3	21	out 15	35	out 3
04	in 4	22	out 16	36	out 4
05	in 5	23		37	
06	in 6	24	out 17	38	out 5
07	in 7	25	out 18	39	out 6
08	in 8	26	out 19	40	out 7
09	in 9	27	out 20	41	out 8
10	in 10	28		42	
11	in 11	29	in 17	43	out 9
12	in 12	30	in 18	44	out 10
13	in 13	31	in 19	45	out 11
14	in 14	32	in 20	46	out 12
15	in 15			47	
16	in 16			48	
17				49	
18				50	

Peripheral equipment A2

CRM2B

01	in 21	19	out 33	33	out 21
02	in 22	20	out 34	34	out 22
03	in 23	21	out 35	35	out 23
04	in 24	22	out 36	36	out 24
05	in 25	23		37	
06	in 26	24	out 37	38	out 25
07	in 27	25	out 38	39	out 26
08	in 28	26	out 39	40	out 27
09	in 29	27	out 40	41	out 28
10	in 30	28		42	
11	in 31	29	in 37	43	out 29
12	in 32	30	in 38	44	out 30
13	in 33	31	in 39	45	out 31
14	in 34	32	in 40	46	out 32
15	in 35			47	
16	in 36			48	
17				49	
18				50	

Peripheral equipment A3

CRM2C

01	in 41	19	out 53	33	out 41
02	in 42	20	out 54	34	out 42
03	in 43	21	out 55	35	out 43
04	in 44	22	out 56	36	out 44
05	in 45	23		37	
06	in 46	24	out 57	38	out 45
07	in 47	25	out 58	39	out 46
08	in 48	26	out 59	40	out 47
09	in 49	27	out 60	41	out 48
10	in 50	28		42	
11	in 51	29	in 57	43	out 49
12	in 52	30	in 58	44	out 50
13	in 53	31	in 59	45	out 51
14	in 54	32	in 60	46	out 52
15	in 55			47	
16	in 56			48	
17				49	
18				50	

Peripheral equipment A4

CRM2D

01	in 61	19	out 73	33	out 61
02	in 62	20	out 74	34	out 62
03	in 63	21	out 75	35	out 63
04	in 64	22	out 76	36	out 64
05	in 65	23		37	
06	in 66	24	out 77	38	out 65
07	in 67	25	out 78	39	out 66
08	in 68	26	out 79	40	out 67
09	in 69	27	out 80	41	out 68
10	in 70	28		42	
11	in 71	29	in 77	43	out 69
12	in 72	30	in 78	44	out 70
13	in 73	31	in 79	45	out 71
14	in 74	32	in 80	46	out 72
15	in 75			47	
16	in 76			48	
17				49	
18				50	

Peripheral equipment B1

CRM4A

01	in 81	08	out 85	14	out 81
02	in 82	09	out 86	15	out 82
03	in 83	10	out 87	16	out 83
04	in 84	11	out 88	17	out 84
05	in 85	12		18	
06	in 86	13	in 88	19	
07	in 87			20	

Peripheral equipment B2

CRM4B

01	in 89	08	out 93	14	out 89
02	in 90	09	out 94	15	out 90
03	in 91	10	out 95	16	out 91
04	in 92	11	out 96	17	out 92
05	in 93	12		18	
06	in 94	13	in 96	19	
07	in 95			20	

Welding interface

CRW1

01	aout 1	13	ain 1	23	WDO 1
02	aout 1–C	14	ain 1–C	24	WDO 2
03	aout 2	15	ain 2	25	WDO 3
04	aout 2–C	16	ain 2–C	26	WDO 4
05	WDI 1	17		27	WDO 5
06	WDI 2	18		28	WDO 6
07	WDI 3	19	0V	29	WDO 7
08	WDI 4	20	0V	30	WDO 8
09	WDI 5	21	0V	31	WDI +
10	WDI 6	22	0V	32	WDI –
11	WDI 7			33	+24V
12	WDI 8			34	+24V

Analog input interface

CRW2

01		08	ain 6	14	ain 3
02		09	ain 6–C	15	ain 3–C
03		10		16	ain 4
04		11		17	ain 4–C
05		12		18	ain 5
06		13		19	ain 5–C
07				20	

in** and out** are physical numbers.
ain *–C is the common signal line for ain *.

I/O Unit–MODEL A

I/O Unit–MODEL A (Modular I/O) is the I/O module which includes the plural modules. Plural modules can be connected within the limits of 512 signal lines in all modules. The I/O unit–MODEL A can be used only in master mode. Before using it, contact FANUC.

Figure 3–6. I/O Unit–MODEL A

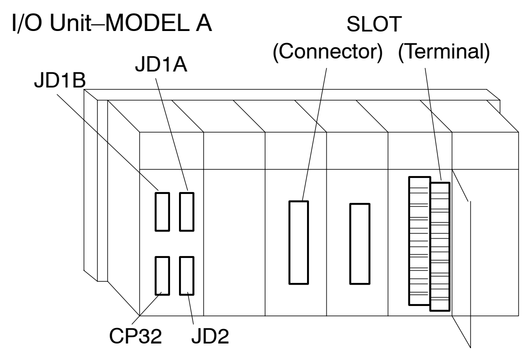
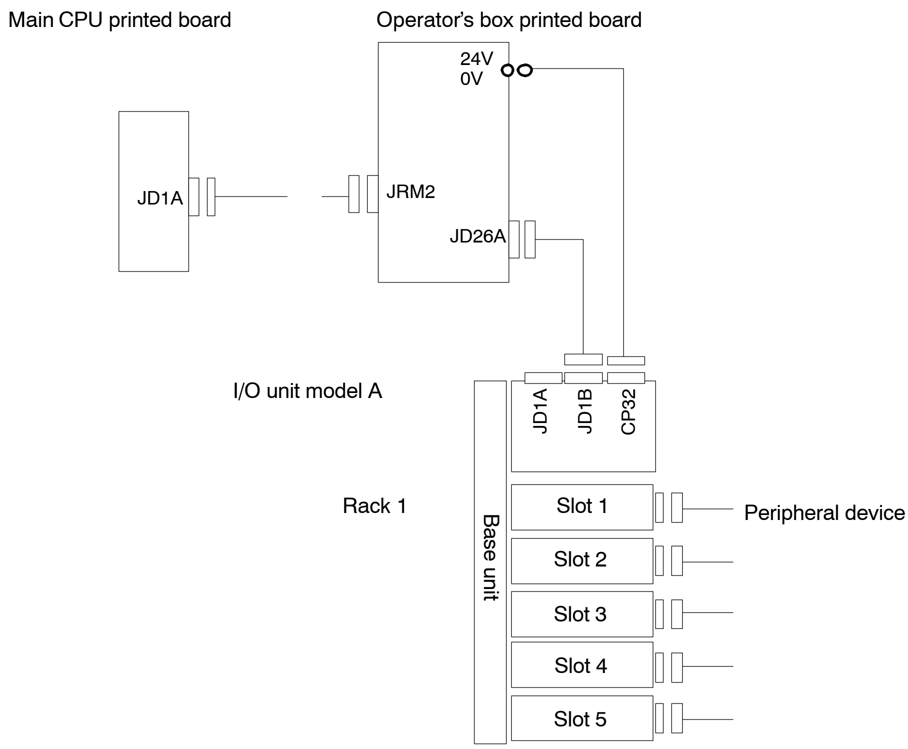


Figure 3–7. I/O Unit–MODEL A Configuration



When using only the I/O unit, assign 18 inputs and 20 outputs of the peripheral device I/O to appropriate signal lines (→ Section 3.3, "Peripheral Devices").

When the I/O unit and process I/O printed circuit board are used simultaneously, the inputs and outputs of the peripheral device I/O are automatically assigned to signal lines on the process I/O printed circuit board.

For details of FANUC I/O Unit–MODEL A, refer to FANUC I/O Unit–MODEL A manual (B–61813EN)

Figure 3–8. I/O Unit MODEL A interface

AID 32 A/B

01	in 31	19	in 29	33	in 32
02	in 28	20	in 26	34	in 30
03	in 25	21	CMC	35	in 27
04	+ 24V	22	in 23	36	CMC
05	0 V	23	in 21	37	in 24
06	in 20	24	in 18	38	in 22
07	in 17	25		39	in 19
08	+ 24V	26		40	CMC
09	0 V	27	in 13	41	CMC
10	in 15	28	in 10	42	in 16
11	in 12	29	CMA	43	in 14
12	in 9	30	in 7	44	in 11
13	+ 24V	31	in 5	45	CMA
14	0 V	32	in 2	46	in 8
15	in 4			47	in 6
16	in 1			48	in 3
17	+ 24V			49	CMA
18	0 V			50	CMA

AID 32 E/F

01	in 31	19	in 29	33	in 32
02	in 28	20	in 26	34	in 30
03	in 25	21	CMD	35	in 27
04	+ 24V	22	in 23	36	CMD
05	0 V	23	in 21	37	in 24
06	in 20	24	in 18	38	in 22
07	in 17	25		39	in 19
08	+ 24V	26		40	CMC
09	0 V	27	in 13	41	CMC
10	in 15	28	in 10	42	in 16
11	in 12	29	CMB	43	in 14
12	in 9	30	in 7	44	in 11
13	+ 24V	31	in 5	45	CMB
14	0 V	32	in 2	46	in 8
15	in 4			47	in 6
16	in 1			48	in 3
17	+ 24V			49	CMA
18	0 V			50	CMA

AID 16 C/D

01	CM	02	in 1
03	in 2	04	in 3
05	in 4	06	in 5
07	in 6	08	in 7
09	in 8	10	in 9
11	in 10	12	in 11
13	in 12	14	in 13
15	in 14	16	in 15
17	in 16	18	
19		20	CM

AIA 16 G

01	in 1	02	in 2
03	in 3	04	in 4
05	in 5	06	in 6
07	in 7	08	in 8
09	in 9	10	in 10
11	in 11	12	in 12
13	in 13	14	in 14
15	in 15	16	in 16
17		18	
19	CM	20	

AOD 32 A/C/D

01	out 31	19	out 29	33	out 32
02	out 28	20	out 26	34	out 30
03	out 25	21	CMD	35	out 27
04	+ 24V	22	out 23	36	CMD
05	0 V	23	out 21	37	out 24
06	out 20	24	out 18	38	out 22
07	out 17	25		39	out 19
08	+ 24V	26		40	CMC
09	0 V	27	out 13	41	CMC
10	out 15	28	out 10	42	out 16
11	out 12	29	CMB	43	out 14
12	out 9	30	out 7	44	out 11
13	+ 24V	31	out 5	45	CMB
14	0 V	32	out 2	46	out 8
15	out 4			47	out 6
16	out 1			48	out 3
17	+ 24V			49	CMA
18	0 V			50	CMA

CM indicates the common signal line.

AOD 16 C/D

01	CMA	02	out 1
03	out 2	04	out 3
05	out 4	06	out 5
07	out 6	08	out 7
09	out 8	10	CMA
11	CMB	12	out 9
13	out 10	14	out 11
15	out 12	16	out 13
17	out 14	18	out 15
19	out 16	20	CMB

AOD 08 C/D

01		02	
03		04	in 1
05		06	in 2
07		08	in 3
09		10	in 4
11		12	in 5
13		14	in 6
15		16	in 7
17		18	in 8
19		20	

AOA 12 F

01	out 1	02	out 2
03	out 3	04	out 4
05	out 5	06	out 6
07		08	
09	CMA	10	
11	out 7	12	out 8
13	out 9	14	out 10
15	out 11	16	out 12
17		18	
19	CMB	20	

AOA 08 E

01		02	out 1
03		04	out 2
05		06	out 3
07		08	out 4
09		10	CMA
11		12	out 5
13		14	out 6
15		16	out 7
17		18	out 8
19		20	CMB

AOA 05 E

01	CMA	02	out 1
03	out 2	04	out 1–C
05	out 4	06	out 2
07	out 6	08	out 2–C
09	out 8	10	out 3
11	CMB	12	out 3–C
13	out 10	14	out 4
15	out 12	16	out 4–C
17	out 14	18	out 5
19	out 16	20	out 5–C

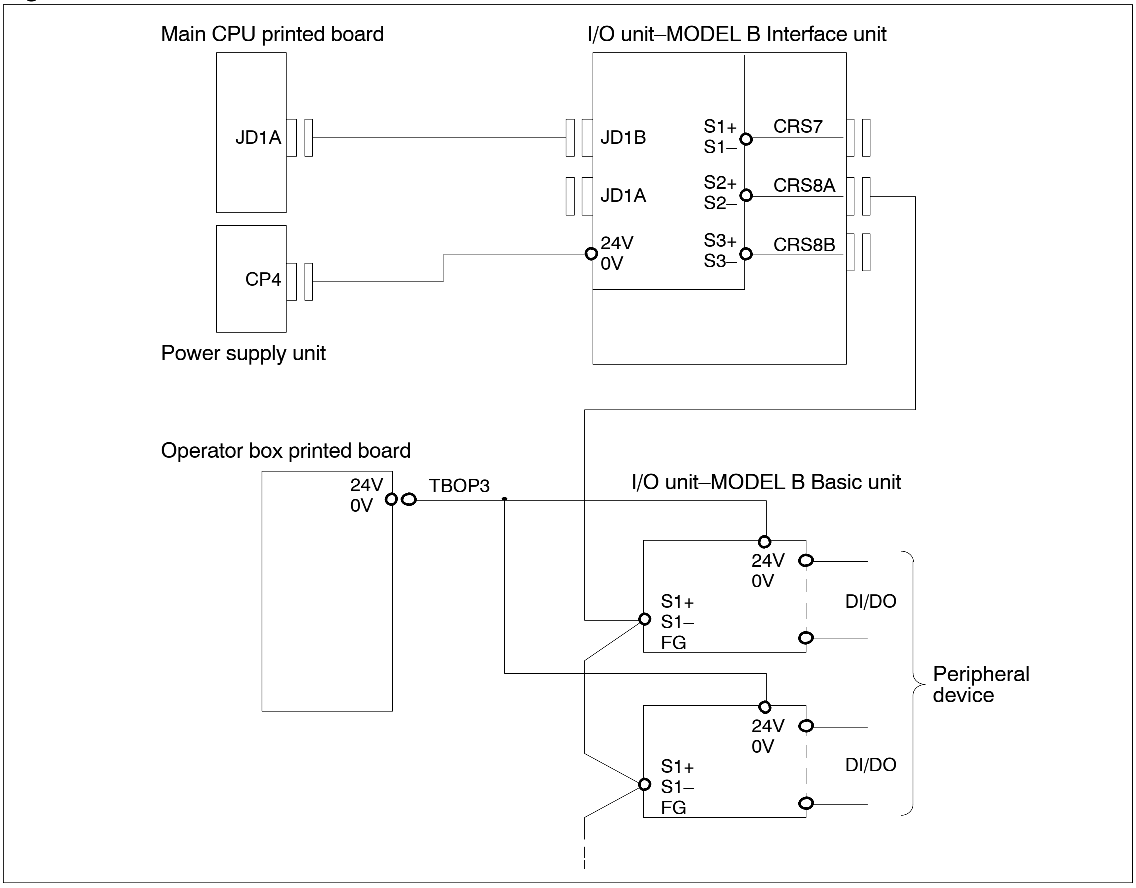
in**, out** indicates the physical number

I/O unit–MODEL B

The I/O unit–MODEL B consists of an interface unit and more than one DI/DO unit. The DI/DO units are used to input/output signals. The interface unit is used to assemble I/O information in the DI/DO units and transfers it to or from the robot controller.

Combining an appropriate number of DI/DO units of different types makes it possible to provide a necessary number of input/output points. Twisted pair cables are used to connect the DI/DO units with the interface unit, thus allowing the DI/DO units to be installed at a distance from the interface unit.

Figure 3–9. I/O Unit–MODEL B



Refer to the FANUC I/O Unit Model B Connection Manual (B–62163EN) for details of the I/O unit–MODEL B.

When the I/O unit–MODEL B is used, the setting is needed on I/O link screen.

3.1.1 Digital I/O

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

Configuration of Input/Output

In digital I/O, the configuration of the signal lines can be redefined. The following items are set. Refer to Figure 3–4 for the configuration of the rack and slot.

> **CAUTION**
> When a process I/O printed circuit board is connected, the standard assignment is made at the factory. When no process I/O printed circuit board is connected and I/O unit model A/B is connected, all digital input/output signals are assigned to the digital I/O at the factory. No digital input/output signals are assigned to the peripheral device I/O. Divide the digital input/output signals between the digital I/O and peripheral device I/O and reassign the signals to them.

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

RACK

The rack indicates the master of I/O link, the CRM79 interface, and the kind of hardware which composes I/O module to robot control PC board.

- 32 = JD1B interface (at slave mode)
  (CNC as the master of I/O link to the robot control PC board)
- 0 = JD4 interface (Process I/O PC board) (at master mode)
- 1 or more = JD4 interface (I/O Unit–MODEL A) and, MODEL B (at master mode)
- 48 = CRM79 interface

SLOT

The slot indicates the number of I/O module parts which composes RACK. However, it is fixed to 1 for the CRM79 interface and JD1B interface (CNC).

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

START PT

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

NOTE A physical number specifies the pin of Input/Output lines on the I/O module. Logical number is allocated to this physical number. And eight signal lines which are represented in logical number and are included in the same class are allocated at the same time.

NOTE Physical numbers starting with in 19 and out 21 can be assigned to the digital I/O because 18 input physical numbers (in 1 to 18) and 20 output physical numbers (out 1 to 20) on the process I/O printed circuit board are assigned to the peripheral device I/O.

NOTE Any physical number can be specified as the start point. Not allocated signal is automatically allocated to other logical number.

Polarity

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

– **NORMAL** = The current is turned on when the signal is set on.

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

Complementary

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

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

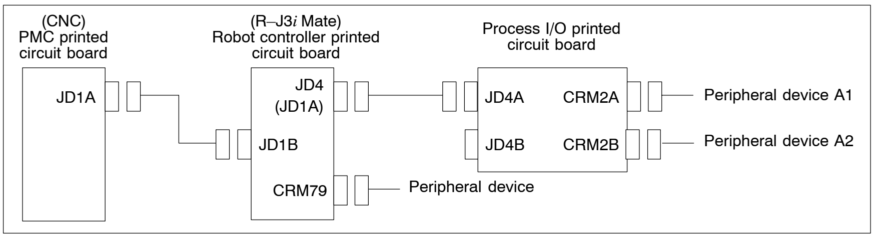
Output

The value of a digital output signal can be specified by executing a program or performing manual operation. (See Section 4.6, “I/O Instruction,” and Section 6.4, “Manual I/O Control.”)

Simulated input/output

When simulated input/output is selected, a program can be tested without sending or receiving signals to or from the external equipment. (See Section 6.3.1,“Specifying test execution”)

Figure 3–10. Digital I/O and peripheral I/O interfaces (with UOP allocated to CRM2A)



Simple

I/O Link interface (JD1B)
Input signal address (72 point)

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

Output signal address (68 point)

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

Peripheral device interface
Connector number CRM79

01	SDI 101			33	SDO 101
02	SDI 102	19	SDICOM1	34	SDO 102
03	SDI 103	20	SDICOM2	35	SDO 103
04	SDI 104	21	SDO 120	36	SDO 104
05	SDI 105	22	SDI 117	37	SDO 105
06	SDI 106	23	SDI 118	38	SDO 106
07	SDI 107	24	SDI 119	39	SDO 107
08	SDI 108	25	SDI 120	40	SDO 108
09	SDI 109	26	SDO 117	41	SDO 109
10	SDI 110	27	SDO 118	42	SDO 110
11	SDI 111	28	SDO 119	43	SDO 111
12	SDI 112	29	0V	44	SDO 112
13	SDI 113	30	0V	45	SDO 113
14	SDI 114	31	24E	46	SDO 114
15	SDI 115	32	24E	47	SDO 115
16	SDI 116			48	SDO 116
17	0V			49	24E
18	0V			50	24E

Peripheral device interface
CRM81

	A	B
01	SDI 81	SDI 82
02	SDI 83	SDI 84
03	SDI 85	SCI 86
04	SDI 87	SDI 88
05	SDICOM3	
06		
07	SDO81	SDO82
08	SDO83	SDO84
09	reserve	0V
10	24E	0V

Peripheral device A1
CRM2A

01	*HOLD			33	CMDENBL
02	RESET(*1)	19	SDI 129	34	FAULT
03	START(*2)	20	SDI 130	35	BATALM
04	ENBL	21	SDI 131	36	BUSY
05	PNS1	22	SDI 132	37	
06	PNS2	23		38	SDO 121
07	PNS3	24	SDI 133	39	SDO 122
08	PNS4	25	SDI 134	40	SDO 123
09	SDI 121	26	SDI 135	41	SDO 124
10	SDI 122	27	SDI 136	42	
11	SDI 123	28		43	SDO 125
12	SDI 124	29	SDI 129	44	SDO 126
13	SDI 125	30	SDI 130	45	SDO 127
14	SDI 126	31	SDI 131	46	SDO 128
15	SDI 127	32	SDI 132	47	
16	SDI 128			48	
17				49	
18				50	

*1 RESET functions also as CSTOPI at the same time.
*2 This pin functions as PNSTROBE on the rising edge of the signal, and as START on the falling edge of the signal.

All

I/O Link interface (JD1B)
Input signal address (72 point)

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

Output signal address (68 point)

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

Peripheral device interface
Connector number CRM79

01	SDI 101			33	SDO 101
02	SDI 102	19	SDICOM1	34	SDO 102
03	SDI 103	20	SDICOM2	35	SDO 103
04	SDI 104	21	SDO 120	36	SDO 104
05	SDI 105	22	SDI 117	37	SDO 105
06	SDI 106	23	SDI 118	38	SDO 106
07	SDI 107	24	SDI 119	39	SDO 107
08	SDI 108	25	SDI 120	40	SDO 108
09	SDI 109	26	SDO 117	41	SDO 109
10	SDI 110	27	SDO 118	42	SDO 110
11	SDI 111	28	SDO 119	43	SDO 111
12	SDI 112	29	0V	44	SDO 112
13	SDI 113	30	0V	45	SDO 113
14	SDI 114	31	24E	46	SDO 114
15	SDI 115	32	24E	47	SDO 115
16	SDI 116			48	SDO 116
17	0V			49	24E
18	0V			50	24E

Peripheral device interface
CRM81

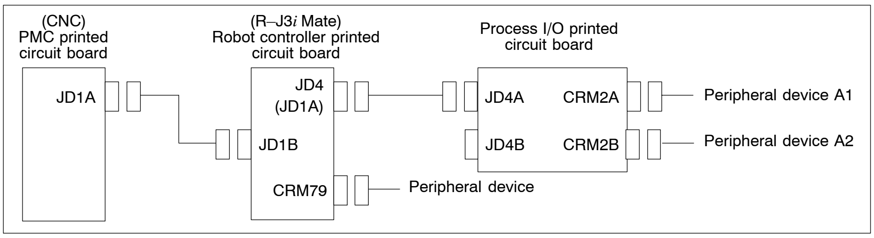
	A	B
01	SDI 81	SDI 82
02	SDI 83	SDI 84
03	SDI 85	SCI 86
04	SDI 87	SDI 88
05	SDICOM3	
06		
07	SDO81	SDO82
08	SDO83	SDO84
09	reserve	0V
10	24E	0V

Peripheral device A1
CRM2A

01	*IMSTP			33	CMDENBL
02	*HOLD	19	ACK3/SNO3	34	SYSRDY
03	*SFSPD	20	ACK4/SNO4	35	PROGRUN
04	CSTOPI	21	ACK5/SNO5	36	PAUSED
05	RESET	22	ACK6/SNO6	37	
06	START	23		38	HELD
07	HOME	24	ACK7/SNO7	39	FAULT
08	ENBL	25	ACK8/SNO8	40	ATPERCH
09	RSR1/PNS1	26	SNACK	41	TPENBL
10	RSR2/PNS2	27	reserve	42	
11	RSR3/PNS3	28		43	BATALM
12	RSR4/PNS4	29	PNSTROBE	44	BUSY
13	RSR5/PNS5	30	PROD_START	45	ACK1/SNO1
14	RSR6/PNS6	31	SDI 121	46	ACK2/SNO2
15	RSR7/PNS7	32	SDI 122	47	
16	RSR8/PNS8			48	
17				49	
18				50	

Note) Set the I/O link as the master.

Figure 3–11. Digital I/O and peripheral I/O interfaces (with UOP allocated to JD1B)



Simple

I/O Link interface (JD1B)
Input signal address (72 point)

	bit 0	bit 1	bit 2	bit 3	bit 4	bit 5	bit 6	bit 7
0	*HOLD	RESET(*1)	START(*2)	ENBL	PNS1	PNS2	PNS3	PNS4
1	SDI 1	SDI 2	SDI 3	SDI 4	SDI 5	SDI 6	SDI 7	SDI 8
2	SDI 9	SDI 10	SDI 11	SDI 12	SDI 13	SDI 14	SDI 15	SDI 16
3	SDI 17	SDI 18	SDI 19	SDI 20	SDI 21	SDI 22	SDI 23	SDI 24
4	SDI 25	SDI 26	SDI 27	SDI 28	SDI 29	SDI 30	SDI 31	SDI 32
5	SDI 33	SDI 34	SDI 35	SDI 36	SDI 37	SDI 38	SDI 39	SDI 40
6	SDI 41	SDI 42	SDI 43	SDI 44	SDI 45	SDI 46	SDI 47	SDI 48
7	SDI 49	SDI 50	SDI 51	SDI 52	SDI 53	SDI 54	SDI 55	SDI 56
8	SDI 57	SDI 58	SDI 59	SDI 60	SDI 61	SDI 62	SDI 63	SDI 64

Output signal address (68 point)

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

Connector number

Peripheral device interface
CRM79

01	SDI 101			33	SDO 101
02	SDI 102	19	SDICOM1	34	SDO 102
03	SDI 103	20	SDICOM2	35	SDO 103
04	SDI 104	21	SDO 120	36	SDO 104
05	SDI 105	22	SDI 117	37	SDO 105
06	SDI 106	23	SDI 118	38	SDO 106
07	SDI 107	24	SDI 119	39	SDO 107
08	SDI 108	25	SDI 120	40	SDO 108
09	SDI 109	26	SDO 117	41	SDO 109
10	SDI 110	27	SDO 118	42	SDO 110
11	SDI 111	28	SDO 119	43	SDO 111
12	SDI 112	29	0V	44	SDO 112
13	SDI 113	30	0V	45	SDO 113
14	SDI 114	31	24E	46	SDO 114
15	SDI 115	32	24E	47	SDO 115
16	SDI 116			48	SDO 116
17	0V			49	24E
18	0V			50	24E

Peripheral device A1 (Enabled only when the I/O link is set as the master)
CRM2A

01	SDI 121			33	SDO 121
02	SDI 122	19	SDO 133	34	SDO 122
03	SDI 123	20	SDO 134	35	SDO 123
04	SDI 124	21	SDO 135	36	SDO 124
05	SDI 125	22	SDO 136	37	
06	SDI 126	23		38	SDO 125
07	SDI 127	24	SDO 127	39	SDO 126
08	SDI 128	25	SDO 128	40	SDO 127
09	SDI 129	26	SDO 129	41	SDO 128
10	SDI 130	27	SDO 120	42	
11	SDI 131	28		43	SDO 129
12	SDI 132	29	SDI 137	44	SDO 130
13	SDI 133	30	SDI 138	45	SDO 131
14	SDI 134	31	SDI 139	46	SDO 132
15	SDI 135	32	SDI 140	47	
16	SDI 136			48	
17				49	
18				50	

Peripheral device interface
CRM81

	A	B
01	SDI 81	SDI 82
02	SDI 83	SDI 84
03	SDI 85	SCI 86
04	SDI 87	SDI 88
05	SDICOM3	
06		
07	SDO81	SDO82
08	SDO83	SDO84
09	reserve	0V
10	24E	0V

*1 RESET functions also as CSTOPI at the same time.
*2 This pin functions as PNSTROBE on the rising edge of the signal, and as START on the falling edge of the signal.

All

I/O Link interface (JD1B)
Input signal address (72 point)

	bit 0	bit 1	bit 2	bit 3	bit 4	bit 5	bit 6	bit 7
0	*IMSTP	*HOLD	*SFSPD	CSTOPI	RESET	START	HOME	ENBL
1	RSR1/PNS1	RSR2/PNS2	RSR3/PNS3	RSR4/PNS4	RSR5/PNS5	RSR6/PNS6	RSR7/PNS7	RSR8/PNS8
2	PNSTROBE	PROD_START	SDI 1	SDI 2	SDI 3	SDI 4	SDI 5	SDI 6
3	SDI 7	SDI 8	SDI 9	SDI 10	SDI 11	SDI 12	SDI 13	SDI 14
4	SDI 15	SDI 16	SDI 17	SDI 18	SDI 19	SDI 20	SDI 21	SDI 22
5	SDI 23	SDI 24	SDI 25	SDI 26	SDI 27	SDI 28	SDI 29	SDI 30
6	SDI 31	SDI 32	SDI 33	SDI 34	SDI 35	SDI 36	SDI 37	SDI 38
7	SDI 39	SDI 40	SDI 41	SDI 42	SDI 43	SDI 44	SDI 45	SDI 46
8	SDI 47	SDI 48	SDI 49	SDI 50	SDI 51	SDI 52	SDI 53	SDI 54

Output signal address (68 point)

	bit 0	bit 1	bit 2	bit 3	bit 4	bit 5	bit 6	bit 7
0	CMDENBL	SYSRDY	PROGRUN	PAUSED	HELD	FAULT	ATPERCH	TPENBL
1	BATALM	BUSY	ACK1/SNO1	ACK2/SNO2	ACK3/SNO3	ACK4/SNO4	ACK5/SNO5	ACK6/SNO6
2	ACK7/SNO7	ACK8/SNO8	SNACK	reserve	SDO 1	SDO 2	SDO 3	SDO 4
3	SDO 5	SDO 6	SDO 7	SDO 8	SDO 9	SDO 10	SDO 11	SDO 12
4	SDO 13	SDO 14	SDO 15	SDO 16	SDO 17	SDO 18	SDO 19	SDO 20
5	SDO 21	SDO 22	SDO 23	SDO 24	SDO 25	SDO 26	SDO 27	SDO 28
6	SDO 29	SDO 30	SDO 31	SDO 32	SDO 33	SDO 34	SDO 35	SDO 36
7	SDO 37	SDO 38	SDO 39	SDO 40	SDO 41	SDO 42	SDO 43	SDO 44
8	SDO 45	SDO 46	SDO 47	SDO 48				

Connector number

Peripheral device interface
CRM79

01	SDI 101			33	SDO 101
02	SDI 102	19	SDICOM1	34	SDO 102
03	SDI 103	20	SDICOM2	35	SDO 103
04	SDI 104	21	SDO 120	36	SDO 104
05	SDI 105	22	SDI 117	37	SDO 105
06	SDI 106	23	SDI 118	38	SDO 106
07	SDI 107	24	SDI 119	39	SDO 107
08	SDI 108	25	SDI 120	40	SDO 108
09	SDI 109	26	SDO 117	41	SDO 109
10	SDI 110	27	SDO 118	42	SDO 110
11	SDI 111	28	SDO 119	43	SDO 111
12	SDI 112	29	0V	44	SDO 112
13	SDI 113	30	0V	45	SDO 113
14	SDI 114	31	24E	46	SDO 114
15	SDI 115	32	24E	47	SDO 115
16	SDI 116			48	SDO 116
17	0V			49	24E
18	0V			50	24E

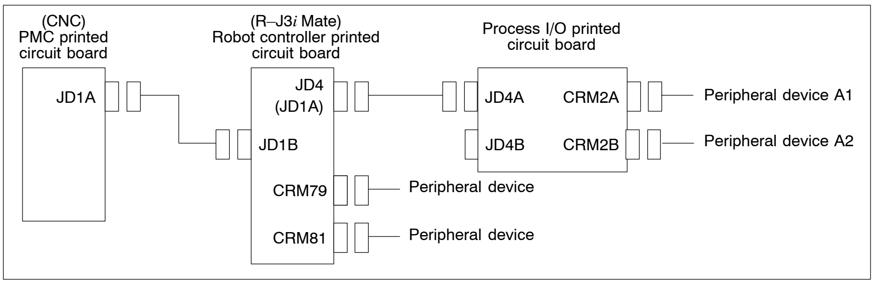
Peripheral device A1 (Enabled only when the I/O link is set as the master)
CRM2A

01	SDI 121			33	SDO 121
02	SDI 122	19	SDO 133	34	SDO 122
03	SDI 123	20	SDO 134	35	SDO 123
04	SDI 124	21	SDO 135	36	SDO 124
05	SDI 125	22	SDO 136	37	
06	SDI 126	23		38	SDO 125
07	SDI 127	24	SDO 127	39	SDO 126
08	SDI 128	25	SDO 128	40	SDO 127
09	SDI 129	26	SDO 129	41	SDO 128
10	SDI 130	27	SDO 120	42	
11	SDI 131	28		43	SDO 129
12	SDI 132	29	SDI 137	44	SDO 130
13	SDI 133	30	SDI 138	45	SDO 131
14	SDI 134	31	SDI 139	46	SDO 132
15	SDI 135	32	SDI 140	47	
16	SDI 136			48	
17				49	
18				50	

Peripheral device interface
CRM81

	A	B
01	SDI 81	SDI 82
02	SDI 83	SDI 84
03	SDI 85	SCI 86
04	SDI 87	SDI 88
05	SDICOM3	
06		
07	SDO81	SDO82
08	SDO83	SDO84
09	reserve	0V
10	24E	0V

Figure 3–12. Digital I/O and peripheral I/O interfaces (with UOP allocated to CRM79)



Simple

I/O Link interface (JD1B)
Input signal address (72 point)

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

Output signal address (68 point)

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

Peripheral device interface
Connector number CRM79

01	SDI 101			33	SDO 101
02	SDI 102	19	SDICOM1	34	SDO 102
03	SDI 103	20	SDICOM2	35	SDO 103
04	SDI 104	21	SDO 120	36	SDO 104
05	SDI 105	22	SDI 117	37	SDO 105
06	SDI 106	23	SDI 118	38	SDO 106
07	SDI 107	24	SDI 119	39	SDO 107
08	SDI 108	25	SDI 120	40	SDO 108
09	*HOLD	26	SDO 117	41	SDO 109
10	RESET(*1)	27	SDO 118	42	SDO 110
11	START(*2)	28	SDO 119	43	SDO 111
12	ENBL	29	0V	44	SDO 112
13	PNS1	30	0V	45	CMDENBL
14	PNS2	31	24E	46	FAULT
15	PNS3	32	24E	47	BATALM
16	PNS4			48	BUSY
17	0V			49	24E
18	0V			50	24E

Peripheral device A1 (Enabled only when the I/O link is set as the master)
CRM2A

01	SDI 121			33	SDO 121
02	SDI 122	19	SDO 133	34	SDO 122
03	SDI 123	20	SDO 134	35	SDO 123
04	SDI 124	21	SDO 135	36	SDO 124
05	SDI 125	22	SDO 136	37	
06	SDI 126	23		38	SDO 125
07	SDI 127	24	SDO 127	39	SDO 126
08	SDI 128	25	SDO 128	40	SDO 127
09	SDI 129	26	SDO 129	41	SDO 128
10	SDI 130	27	SDO 120	42	
11	SDI 131	28		43	SDO 129
12	SDI 132	29	SDI 137	44	SDO 130
13	SDI 133	30	SDI 138	45	SDO 131
14	SDI 134	31	SDI 139	46	SDO 132
15	SDI 135	32	SDI 140	47	
16	SDI 136			48	
17				49	
18				50	

Peripheral device interface (option)
CRM81

	A	B
01	SDI 81	SDI 82
02	SDI 83	SDI 84
03	SDI 85	SCI 86
04	SDI 87	SDI 88
05	SDICOM3	
06		
07	SDO81	SDO82
08	SDO83	SDO84
09	reserve	0V
10	24E	0V

*1 RESET functions also as CSTOPI at the same time.
*2 This pin functions as PNSTROBE on the rising edge of the signal, and as START on the falling edge of the signal.

All

I/O Link interface (JD1B)
Input signal address (72 point)

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

Output signal address (68 point)

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

Peripheral device interface
Connector number CRM79

01	*IMSTP			33	CMDENBL
02	*HOLD	19	SDICOM1	34	SYSRDY
03	*SFSPD	20	SDICOM2	35	PROGRUN
04	CSTOPI	21	reserve	36	PAUSED
05	RESET	22	PNSTROBE	37	HELD
06	START	23	PROD_START	38	FAULT
07	HOME	24	SDI 119	39	ATPERCH
08	ENBL	25	SDI 120	40	TPENBL
09	RSR1/PNS1	26	ACK7/SNO7	41	BATALM
10	RSR2/PNS2	27	ACK8/SNO8	42	BUSY
11	RSR3/PNS3	28	SNACK	43	ACK1/SNO1
12	RSR4/PNS4	29	0V	44	ACK2/SNO2
13	RSR5/PNS5	30	0V	45	ACK3/SNO3
14	RSR6/PNS6	31	24E	46	ACK4/SNO4
15	RSR7/PNS7	32	24E	47	ACK5/SNO5
16	RSR8/PNS8			48	ACK6/SNO6
17	0V			49	24E
18	0V			50	24E

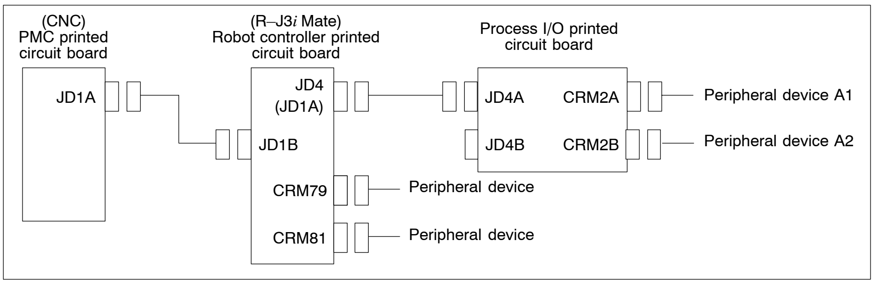
Peripheral device A1 (Enabled only when the I/O link is set as the master)
CRM2A

01	SDI 121			33	SDO 121
02	SDI 122	19	SDO 133	34	SDO 122
03	SDI 123	20	SDO 134	35	SDO 123
04	SDI 124	21	SDO 135	36	SDO 124
05	SDI 125	22	SDO 136	37	
06	SDI 126	23		38	SDO 125
07	SDI 127	24	SDO 127	39	SDO 126
08	SDI 128	25	SDO 128	40	SDO 127
09	SDI 129	26	SDO 129	41	SDO 128
10	SDI 130	27	SDO 120	42	
11	SDI 131	28		43	SDO 129
12	SDI 132	29	SDI 137	44	SDO 130
13	SDI 133	30	SDI 138	45	SDO 131
14	SDI 134	31	SDI 139	46	SDO 132
15	SDI 135	32	SDI 140	47	
16	SDI 136			48	
17				49	
18				50	

Peripheral device interface (option)
CRM81

	A	B
01	SDI 81	SDI 82
02	SDI 83	SDI 84
03	SDI 85	SCI 86
04	SDI 87	SDI 88
05	SDICOM3	
06		
07	SDO81	SDO82
08	SDO83	SDO84
09	reserve	0V
10	24E	0V

Figure 3–13. Digital I/O and peripheral I/O interfaces (with UOP allocated to CRM81)



I/O Link interface (JD1B)
Input signal address (72 point)

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

Output signal address (68 point)

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

Peripheral device interface
Connector number CRM79

01	SDI 101			33	SDO 101
02	SDI 102	19	SDICOM1	34	SDO 102
03	SDI 103	20	SDICOM2	35	SDO 103
04	SDI 104	21	SDO 120	36	SDO 104
05	SDI 105	22	SDI 117	37	SDO 105
06	SDI 106	23	SDI 118	38	SDO 106
07	SDI 107	24	SDI 119	39	SDO 107
08	SDI 108	25	SDI 120	40	SDO 108
09	SDI 109	26	SDO 117	41	SDO 109
10	SDI 110	27	SDO 118	42	SDO 110
11	SDI 111	28	SDO 119	43	SDO 111
12	SDI 112	29	0V	44	SDO 112
13	SDI 113	30	0V	45	SDO 113
14	SDI 114	31	24E	46	SDO 114
15	SDI 115	32	24E	47	SDO 115
16	SDI 116			48	SDO 116
17	0V			49	24E
18	0V			50	24E

Peripheral device A1 (Enabled only when the I/O link is set as the master)
CRM2A

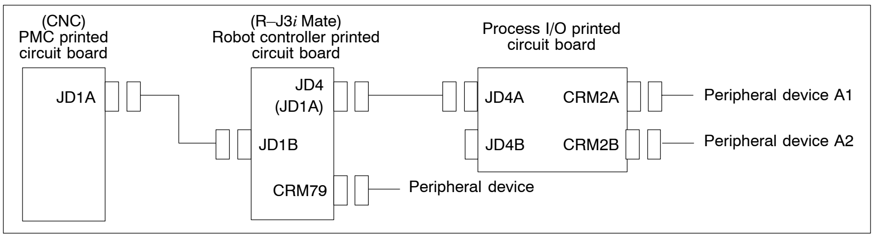
01	SDI 121			33	SDO 121
02	SDI 122	19	SDO 133	34	SDO 122
03	SDI 123	20	SDO 134	35	SDO 123
04	SDI 124	21	SDO 135	36	SDO 124
05	SDI 125	22	SDO 136	37	
06	SDI 126	23		38	SDO 125
07	SDI 127	24	SDO 127	39	SDO 126
08	SDI 128	25	SDO 128	40	SDO 127
09	SDI 129	26	SDO 129	41	SDO 128
10	SDI 130	27	SDO 120	42	
11	SDI 131	28		43	SDO 129
12	SDI 132	29	SDI 137	44	SDO 130
13	SDI 133	30	SDI 138	45	SDO 131
14	SDI 134	31	SDI 139	46	SDO 132
15	SDI 135	32	SDI 140	47	
16	SDI 136			48	
17				49	
18				50	

Peripheral device interface (option)
CRM81

	A	B
01	*HOLD	RESET(*1)
02	START(*2)	ENBL
03	PNS1	PNS2
04	PNS3	PNS4
05	SDICOM3	
06		
07	CMDENBL	FAULT
08	BATALM	BUSY
09	reserve	0V
10	24E	0V

*1 RESET functions also as CSTOPI at the same time.
*2 This pin functions as PNSTROBE on the rising edge of the signal, and as START on the falling edge of the signal.

Figure 3–14. Digital I/O and peripheral I/O interfaces (with no UOP allocated)



I/O Link interface (JD1B)
Input signal address (72 point)

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

Output signal address (68 point)

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

Connector number

Peripheral device interface
CRM79

01	SDI 101			33	SDO 101
02	SDI 102	19	SDICOM1	34	SDO 102
03	SDI 103	20	SDICOM2	35	SDO 103
04	SDI 104	21	SDO 120	36	SDO 104
05	SDI 105	22	SDI 117	37	SDO 105
06	SDI 106	23	SDI 118	38	SDO 106
07	SDI 107	24	SDI 119	39	SDO 107
08	SDI 108	25	SDI 120	40	SDO 108
09	SDI 109	26	SDO 117	41	SDO 109
10	SDI 110	27	SDO 118	42	SDO 110
11	SDI 111	28	SDO 119	43	SDO 111
12	SDI 112	29	0V	44	SDO 112
13	SDI 113	30	0V	45	SDO 113
14	SDI 114	31	24E	46	SDO 114
15	SDI 115	32	24E	47	SDO 115
16	SDI 116			48	SDO 116
17	0V			49	24E
18	0V			50	24E

Peripheral device (Enabled only when the I/O link is set as the master)
CRM2A

01	SDI 121			33	SDO 121
02	SDI 122	19	SDO 133	34	SDO 122
03	SDI 123	20	SDO 134	35	SDO 123
04	SDI 124	21	SDO 135	36	SDO 124
05	SDI 125	22	SDO 136	37	
06	SDI 126	23		38	SDO 125
07	SDI 127	24	SDO 127	39	SDO 126
08	SDI 128	25	SDO 128	40	SDO 127
09	SDI 129	26	SDO 129	41	SDO 128
10	SDI 130	27	SDO 120	42	
11	SDI 131	28		43	SDO 129
12	SDI 132	29	SDI 137	44	SDO 130
13	SDI 133	30	SDI 138	45	SDO 131
14	SDI 134	31	SDI 139	46	SDO 132
15	SDI 135	32	SDI 140	47	
16	SDI 136			48	
17				49	
18				50	

Peripheral device interface
CRM81

	A	B
01	SDI 81	SDI 82
02	SDI 83	SDI 84
03	SDI 85	SCI 86
04	SDI 87	SDI 88
05	SDICOM3	
06		
07	SDO81	SDO82
08	SDO83	SDO84
09	reserve	0V
10	24E	0V

Procedure 3–1 Configuring Digital I/O

CAUTION
When a process I/O printed circuit board is connected, the standard assignment is made at the factory. When no process I/O printed circuit board is connected and I/O unit model A/B is connected, all digital input/output signals are assigned to the digital I/O at the factory and no digital input/output signals are assigned to the peripheral device I/O. Divide the digital input/output signals between the digital I/O and peripheral device I/O and reassign the signals to them.

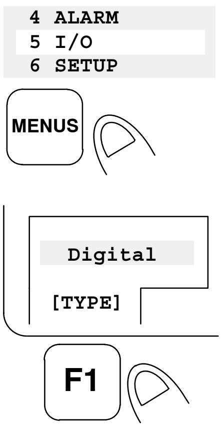
Step

1 Press the MENUS key. The screen menu is displayed.

2 Select 5 [I/O].

3 Press F1 [TYPE]. The screen change menu is displayed.

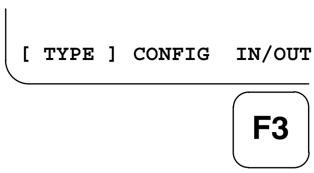
4 Select "Digital."



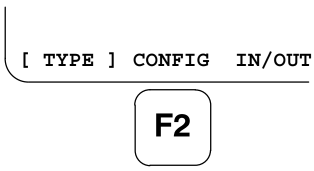
Digital I/O Selection Screen

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

[TYPE]  CONFIG  IN/OUT   ON     OFF
```

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



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



Digital I/O Configuration Screen

```
 I/O Digital Out              JOINT  10 %
    #      RANGE     RACK  SLOT  START  STAT.
    1  DO[  1- 20]    0     1      21 ACTIV
    2  DO[ 21-512]    0     0       0 UNASG

 [ TYPE ] MONITOR IN/OUT  DELETE   HELP
```

7 Manipulating the I/O assignment screen

a) Place the cursor on "Range," and specify the range of signals to be assigned.

b) Line division is performed automatically according to the specified range.

c) Enter appropriate values for “Rack,” “Slot,” and “Start point.”

d) When the entered values are valid, abbreviation “PEND” is displayed in “Status.”
If any entered value is invalid, abbreviation “INVAL” is displayed in “Status.”

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

The abbreviations that will appear in “Status” mean the following:
ACTIV: This assignment is now in use.
PEND: Assignment is normal. Turning the power off and on again causes the ACTIV status to be entered.
INVAL: A specified value is invalid.
UNASG: No assignment has been made.

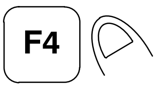
NOTE If process I/O printed circuit boards are connected, 18 input signals and 20 output signals on the first board are connected to the peripheral I/O by standard setting.

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

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

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

```
DETAIL   HELP >
```



Digital I/O detail screen

```
I/O Digital Out                JOINT  10 %
Port Detail                          1/3

    Digital Output: [    1]

  1 Comment : [                        ]

  2     Polarity : NORMAL

  3 Complementary : FALSE   [   1 -   2]



[ TYPE ]  PRV-PT  NXT-PT
```

To return to the selection screen, press PREV key.

10 To add a comment:

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

```
           JOINT  30 %

 1  [              ]
```

b Select the method of naming the comment.

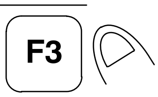
c Press the appropriate function keys to add the comment.

d When you are finished, press the ENTER key.

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

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

```
[ TYPE ]  PREV  NEXT
```



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

```
 I/O Digital Out              JOINT  30 %
                                      3/32
 #      RANGE     RACK  SLOT  START PT
   1 SDO[   1-   8] 0     1     21
   2 SDO[   9-  16] 0     1     29
   3 SDO[  17-  24] 0     1     37
   4 SDO[  25-  32] 0     2      1

 [ TYPE ] MONITOR IN/OUT  DETAIL   HELP >

 [ TYPE ] VERIFY                        >
```

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

WARNING

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

CAUTION

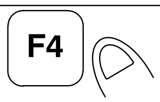
In the first power–up after I/O re–allocation, power recovery would not be executed even if it is enabled.

CAUTION

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

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

```
 I/O Digital In

  DI[1]    S    OFF

IN/OUT    ON     OFF
```




```
 I/O Digital In                  JOINT 30%

  DI[1]   S   ON   [DIGITAL1           ]

 [TYPE]  CONFIG  IN/OUT   ON      OFF
```

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

WARNING

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

3.1.2 Group I/O

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

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

Assignment of I/O signal

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

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

RACK

The rack indicates the master of I/O link to the robot control PC board, the CRM79 interface, and the kind of hardware which composes I/O modules.

– 32 = JD1B interface (at slave mode)
  CNC as the master of I/O link to the robot control PC board
– 0 = JD4 interface (process I/O PC board) (at master mode)
– 1 or more = JD4 interface (I/O Unit–MODEL A / B) (at master mode)
– 48 = CRM79 interface

SLOT

The slot indicates the number of I/O module parts which composes the rack. However, it is fixed to 1 for the CRM79 interface and JD1B interface (CNC).

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

START PT

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

NOTE A physical number specifies the Input/Output pin on the I/O module. Logical number is allocated to this physical number.

NOTE Because the physical numbers for eighteen inputs (”in 1” to ”in 18”) and twenty outputs (”out 1” to ”out 20”) on the first process I/O printed circuit board on the I/O link are allocated to the peripheral I/O signals, the physical numbers for the group I/O signals are ”in 19” and above and ”out 21” and above.

NOTE When two or more I/O boards are connected, the signal lines on the different boards can not be allocated to one group.

NUM PTS

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

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

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

CAUTION

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

Execution of output

The value of the group output can be set by executing the program or manual I/O control.(See Section 4.6, ”I/O instruction”, and Section 6.4,”Manual I/O Control”)

Execution of simulated I/O

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

Procedure 3–2 Configuring group I/O

Step

1 Press the MENUS key. The screen menu is displayed.

2 Select 5 [I/O].

3 Press F1 [TYPE]. The screen change menu is displayed.

4 Select Group. Group I/O list screen is displayed.

```
 4 ALARM
 5 I/O
 6 SETUP
```

MENUS

```
   Group

 [TYPE]
```

F1

Group I/O list screen

```
  I/O Group Out                  JOINT  30 %
        #   SIM VALUE
    GO[   1]   *      0 [              ]
    GO[   2]   *      0 [              ]
    GO[   3]   *      0 [              ]
    GO[   4]   *      0 [              ]
    GO[   5]   *      0 [              ]
    GO[   6]   *      0 [              ]
    GO[   7]   *      0 [              ]
    GO[   8]   *      0 [              ]
    GO[   9]   *      0 [              ]
    GO[  10]   *      0 [              ]
  [ TYPE ] CONFIG  IN/OUT SIMULATE UNSIM
```

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

```
 [ TYPE ] CONFIG  IN/OUT
```

F3

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

```
 [ TYPE ] CONFIG  IN/OUT
```

F2

Group I/O configuration screen

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

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

```
 [ TYPE ] MONITOR IN/OUT
```

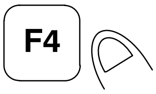
F2

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

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

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

```
IN/OUT  DETAIL   HELP >
```



Group I/O detail screen

```
 I/O Group Out                  JOINT  10 %
 Port Detail                          1/1

     Group Output: [    1]


   1 Comment : [                        ]





 [ TYPE ]  PRV-PT  NXT-PT
```

To return to the selection screen, press PREV key.

9 To add a comment:

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

```
           JOINT  30 %

 1  [              ]
```

b Select the method of naming the comment.

c Press the appropriate function keys to add the comment.

d When you are finished, press the ENTER key.

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

11 When you are finished, press the PREV key to return to the selection screen.

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

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

> **CAUTION**
>
> In the first power–up after I/O re–allocation, power recovery would not be executed even if it is enabled.

> **CAUTION**
>
> After all I/O signals are set, the setting information should be saved in external storage (floppy disk, for example) in case the information needs to be re–loaded. Otherwise, the current setting information would be lost when it is changed.

3.1.3 Analog I/O

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

Configuration of input/output

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

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

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

RACK

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

– 0 = process I/O printed circuit board
– 1 to 16 = I/O unit–MODEL A / B
  The base unit of the I/O unit–MODEL A and the interface unit of the I/O unit–MODEL B are defined as racks 1, 2, ..., according to the sequence of connection.

SLOT

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

CHANNEL

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

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

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

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

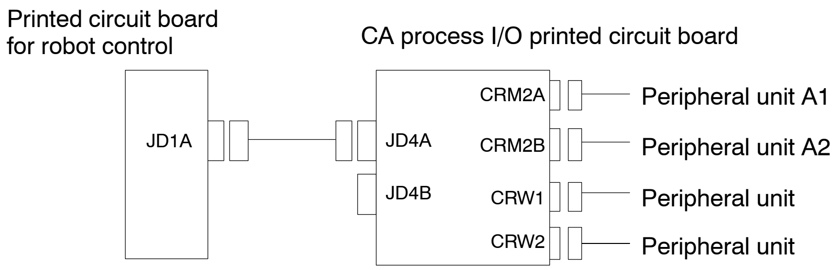
Execution of output

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

Execution of simulated I/O

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

Figure 3–15. Analog I/O Interface



Welding interface

CRW1

01	aout 1	13	ain 1	23	WDO 1
02	aout 1–C	14	ain 1–C	24	WDO 2
03	aout 2	15	ain 2	25	WDO 3
04	aout 2–C	16	ain 2–C	26	WDO 4
05	WDI 1	17		27	WDO 5
06	WDI 2	18		28	WDO 6
07	WDI 3	19	0V	29	WDO 7
08	WDI 4	20	0V	30	WDO 8
09	WDI 5	21	0V	31	WDI +
10	WDI 6	22	0V	32	WDI –
11	WDI 7			33	+24V
12	WDI 8			34	+24V

Analog input interface

CRW2

01		08	ain 6	14	ain 3
02		09	ain 6–C	15	ain 3–C
03		10		16	ain 4
04		11		17	ain 4–C
05		12		18	ain 5
06		13		19	ain 5–C
07				20	

ain *–C is the common signal line for ain *.

Procedure 3–3 Setting analog I/O

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

Step

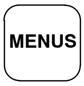
1 Press the MENUS key. The screen menu is displayed.

2 Select 5, [I/O].

3 Press F1, [TYPE]. The screen change menu is displayed.

4 Select Analog. The analog I/O list screen is displayed.

```
4 ALARM
5 I/O
6 SETUP
```




```
Analog
[TYPE]
```

Analog I/O list screen

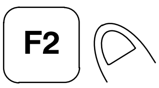
```
 I/O Analog In                  JOINT  30 %
       #   SIM VALUE                 1/25
   AI[   1]  U      0 [              ]
   AI[   2]  U      0 [              ]
   AI[   3]  *      0 [              ]
   AI[   4]  *      0 [              ]
   AI[   5]  *      0 [              ]
   AI[   6]  * 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

 [ TYPE ] CONFIG  IN/OUT SIMULATE UNSIM
```

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

```
[ TYPE ] CONFIG  IN/OUT
```

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

```
[ TYPE ] CONFIG  IN/OUT
```

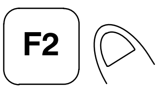


Analog I/O configuration screen

```
 I/O Analog In                  JOINT  30 %
                                     1/25
  AI #  RACK  SLOT  CHANNEL
   1     0     1       1
   2     0     1       2
   3     0     0       0
   4     0     0       0
   5     0     0       0
   6     0     0       0
   7     0     0       0
   8     0     0       0
   9     0     0       0

 [ TYPE ] MONITOR IN/OUT  DETAIL   HELP >
```

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

```
[ TYPE ] MONITOR IN/OUT
```



7 To configure the signals, move the cursor to each item and enter the value.

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

```
[ TYPE ] MONITOR IN/OUT
```

F2

```
 I/O Analog In                    JOINT  30 %
       #   SIM VALUE                   1/25
   AI[   1]  U      0 [analog sign1]
   AI[   2]  U      0 [analog sign2]
   AI[   3]  *      0 [            ]
   AI[   4]  *      0 [            ]

 [ TYPE ] CONFIG  IN/OUT SIMULATE UNSIM
```

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

Analog I/O detail screen

```
IN/OUT  DETAIL   HELP >
```

F4

```
 I/O Analog Out                   JOINT  10 %
 Port Detail                            1/1

     Analog Output: [    1]


   1 Comment : [                        ]




 [ TYPE ]  PRV-PT  NXT-PT
```

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

PREV

10 To add a comment:

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

```
           JOINT  30 %

 1  [              ]
```

ENTER

b Select the method of naming the comment.

c Press the appropriate function keys to add the comment.

d When you are finished, press the ENTER key.

11 To specify the signal attribute, move the cursor to the corresponding field, and select the function key.

12 When you are finished, press the PREV key to return to the selection screen.

13 Turn the controller off and on again so that it can use the new information.

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

> **CAUTION**
>
> In the first power–up after I/O re–allocation, power failure recovery would not be executed even if it is enabled.

CAUTION

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

3.2 Robot I/O

Robot I/O are signals digital signals

Robot to operate the following executions.

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

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

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

Hand breakage input signal, *HBK

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

NOTE Hand breakage detection can be disabled on the system setting screen. See the item of enabling and disabling hand breakage detection in Section 3.14, ”SYSTEM CONFIG MENU.”

Abnormal air pressure input signal, *PPABN input

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

NOTE The abnormal air pressure signal can be enabled or disabled on the system setting screen. See the item of enabling and disabling the abnormal air pressure signal in Section 3.14, ”SYSTEM CONFIG MENU.”

*PPABN input and RDI [6] are allocated to the same pin, and cannot be used at the same time. This pin can be used as RDI [6] only when abnormal air pressure detection is disabled.

***ROT input**

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

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

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

RDI [1 to 6] INPUT
RDO [1 to 6] OUTPUT

The end effector signals, (RDI [1 to 6] and RDO [1 to 6], are general–purpose input and output signals.

The signals in each of the following pairs are mutually exclusive as standard: RDO [1] and RDO [2]; RDO [3] and RDO [4]; and RDO [5] and RDO [6].

*PPABN input and RDI [6] are allocated to the same pin, and cannot be used at the same time. This pin can be used as RDI [6] only when abnormal air pressure detection is disabled.

Procedure 3–4 Setting Robot I/O

Step

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

```
 4 ALARM
 5 I/O
 6 SETUP
```

MENUS

```
  Robot
 [TYPE]
```

F1

Robot I/O Selection Screen

```
 I/O Robot Out                      JOINT 30%
       #        STATUS                1/24
  RDO[1]          OFF  [                   ]
  RDO[2]          OFF  [                   ]
  RDO[3]          OFF  [                   ]
  RDO[4]           ON  [                   ]
  RDO[5]           ON  [                   ]
  RDO[6]          OFF  [                   ]
  RDO[7]          OFF  [                   ]
  RDO[8]           ON  [                   ]
  RDO[9]           ON  [                   ]

[TYPE]            IN/OUT     ON       OFF   >
```

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

```
[TYPE]            IN/OUT
```

F3

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

```
[TYPE]  NUM-SRT CMT-SRT DETAIL  HELP >
```

F2

Robot I/O Detail Screen

```
 I/O Robot Out                      JOINT  10 %
 Port Detail                               1/3

     Robot Dig. Output: [    1]

   1 Comment : [                        ]

   2    Polarity : NORMAL

   3 Complementary : FALSE    [   1 -   2]



 [ TYPE ]  PRV-PT  NXT-PT
```

NOTE On the detailed robot I/O screen,
Items 1: COMMENT
Items 2: POLARITY
Items 3: COMPLEMENTARY

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

7 To add a comment:

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

8 To set the polarity and the complementary pair, move the cursor to the setting column,and select the function key menu.

9 When you are finished,press PREV to return to the list screen.

```
I/O Robot Out                        JOINT  30 %
      #   STATUS                          1/24
  RO[   1]   OFF      [              ]
  RO[   2]   OFF      [              ]
  RO[   3]   OFF      [              ]
  RO[   4]    ON      [              ]

[ TYPE ]        IN/OUT     ON      OFF
```

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

WARNING

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

CAUTION

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

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

```
I/O Robot Out

 RO[1]         OFF

IN/OUT    ON     OFF
```

F4

```
I/O Robot Out                       JOINT 30%

 RO[1]        ON   [                      ]

[TYPE]         IN/OUT     ON       OFF
```

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

WARNING

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

3.3 Peripheral I/O

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

- The JD1B interface (CNC is connected as a master of I/O link to R–J3*i* Mate.) (at slave mode)
- The JD4 interface (The process I/O PC board, the I/O Unit MODEL A and the MODEL B are connected as the slave of I/O link to R–J3*i* Mate.)
- CRM79 interface

Configuration of I/O

18 input signals and 20 output signals of peripheral I/O are assigned to the JD1B interface in default setting at shipment. In this case, 20 input signals and 20 output signals can be used as the general–purpose signal (DI/DO) in the CRM79 interface.

NOTE When the peripheral I/O is allocated to the JD1B interface, don't use the DI[1 to 18]/DO[1 to 20] which is assigned to the same physical number as peripheral I/O.

If the JD1B interface is not used, allocate the peripheral I/O to the CRM79 interface. In this case, 2 input signals can be used as the general–purpose signal (DI) in CRM79 interface.

NOTE When the peripheral I/O is allocated to the CRM79 interface, don't use the DI[101 to 118] and the DO[101 to 120] which is assigned to the same physical number as peripheral I/O.

If the system starts with master mode, the peripheral I/O is allocated to the process I/O printed circuit board automatically.

See Figure 3–10 to 3-14 for the peripheral I/O assignment.

CAUTION

When a process I/O printed circuit board is connected at the master mode, the standard assignment is made at the factory. When no process I/O printed circuit board is connected and I/O unit model A/B is connected, all digital input/output signals are assigned to the digital I/O at the factory. No digital input/output signals are assigned to the peripheral device I/O. Divide the digital input/output signals between the digital I/O and peripheral device I/O and reassign the signals to them.

Remote condition

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

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

- The teach pendant enable switch is set off.
- The remote signal (SI[2]) is on. (For how to turn the remote signal on and off, see the description of Remote/Local setup in Section 3.16, "SYSTEM CONFIG MENU.")
- The *SFSPD input of the peripheral device I/O is on.
- The ENBL input of the peripheral device I/O is on.
- A value of 0 (peripheral device) is set for system variable $RMT_MASTER.

NOTE $RMT_MASTER Specifies the kind of remote device.

0 : Peripheral device
1 : CRT/KB
2 : Host computer
3 : No remote device

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

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

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

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

NOTE Peripheral I/O signals are disabled in the initial state. To enable these signals, set TRUE at "Enable UI signals" on the system configuration screen.

> **WARNING**
> **When connecting the peripheral equipments related to the emergency stop function (for example Protective Fence) to each signal of a robot (for example external emergency stop, fence, servo, etc.), confirm whether emergency stop can work to prevent from connecting incorrectly.**

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

The immediate stop signal specifies an emergency stop by the software.

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

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

> **WARNING**
> **The *IMSTP signal is controlled by software. The use of this signal for safety–critical processing is not recommended. To link this signal with the emergency stop, use this signal together with the EMGIN1 or EMGIN2 signal on the operator's panel printed circuit board. For details of these signals, refer to the "Maintenance Manual."**

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

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

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

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

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

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

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

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

> **WARNING**
> **The *SFSPD signal controls deceleration and stop by software. To stop the robot immediately for safety purposes, use this signal together with the FENCE1 or FENCE2 signal on the operator's panel printed circuit board. For details of these signals, refer to the "Maintenance Manual."**

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

CSTOPI input UI [4] (Always enabled.)

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

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

WARNING

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

Fault reset input signal, RESET, UI [5]

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

Enable input signal, ENBL, UI [8]

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

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

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

These are robot service request signals. When one of these signals is received, the RSR program corresponding to the signal is selected and started to perform automatic operation. When another program is being executed or is stopped temporarily, the selected program is added to the queue and is started once the program being executed terminates. (→ Section 3.5.1, ”Robot service request”)

PNS1 to PNS8 UI [9–16] PNSTROBE UI [17] (Enabled in the remote state.)
[Option = external program selection]

These are program number select signals and a PN strobe signal. When the PNSTROBE input is received, the PNS1 to PNS8 inputs are read to select a program to be executed. When another program is being executed or temporarily stopped, these signals are ignored. (→ Section 3.5.2, ”Program number select”)

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

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

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

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

When another program is being executed or temporarily stopped, this signal is ignored. (Program number select Section 3.5.2)

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

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

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

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

CMDENBL input UO [1]

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

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

SYSRDY output UO [2]

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

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

PROGRUN output UO [3]

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

PAUSED output UO [4]

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

HELD output UO [5]

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

FAULT output UO [6]

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

ATPERCH output UO [7]

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

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

TPENBL output UO [8]

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

BATALM output UO [9]

BATALM indicates a low–voltage alarm for the backup battery of the control unit or robot pulse coder. Turn the power to the control unit on and replace the battery.

BUSY output UO [10]

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

ACK1 to ACK8 outputs UO [11–18]

When the RSR function is enabled, ACK1 to ACK4 are used together with the function. When an RSR input is accepted, a pulse of the corresponding signal is output as an acknowledgment. The pulse width can be specified. (→ Section 3.5.1, ”Robot service request”)

SNO1 to SNO8 outputs UO [11–18] [Option = external program selection]

When the PNS function is enabled, SNO1 to SNO8 are used together with the function. The currently selected program number (signal corresponding to the PNS1 to PNS8 inputs) is always output, in binary code, as confirmation. The selection of another program changes SNO1 to SNO8. (→ Section 3.5.2, ”Program number select”)

SNACK output UO [19] [Option = external program selection]

When the PNS function is enabled, SNACK is used together with the function. When the PNS inputs are accepted, a pulse of this signal is output as an acknowledgment. The pulse width can be specified. (→ Section 3.5.2, ”Program number selection”)

Procedure 3–5 Configurating Peripheral I/O 1

> **CAUTION**
> When a process I/O printed circuit board is connected at the master mode, the standard assignment is made at the factory. When no process I/O printed circuit board is connected and I/O unit model A/B is connected, all digital input/output signals are assigned to the digital I/O at the factory and no digital input/output signals are assigned to the peripheral device I/O. Divide the digital input/output signals between the digital I/O and peripheral device I/O and reassign the signals to them.

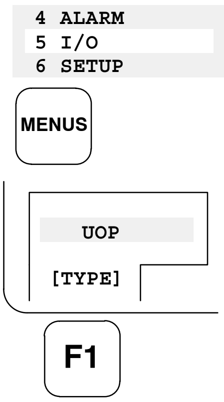
Step

1 Press the MENUS key. The screen menu is displayed.

2 Select 5, [I/O].

3 Press the F1 key, [TYPE]. The screen change menu is displayed.

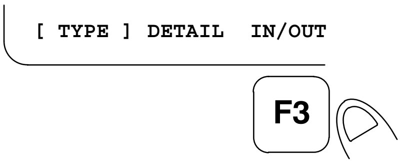
4 Select UOP.



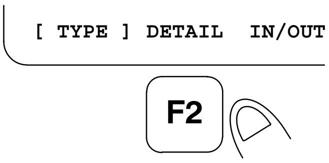
Peripheral I/O Selection Screen

```
I/O UOP Out                          JOINT 30%
    #         STATUS
  UO[1]        ON    [*HOLD                  ]
  UO[2]        OFF   [FAULT reset            ]
  UO[3]        OFF   [Start                  ]
  UO[4]        ON    [Enable                 ]
  UO[5]        OFF   [PNS1                   ]
  UO[6]        OFF   [PNS2                   ]
  UO[7]        OFF   [PNS3                   ]
  UO[8]        OFF   [PNS4                   ]
  UO[9]        *     [                       ]

[ TYPE ] DETAIL  IN/OUT    ON      OFF
```

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



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



Peripheral I/O configuration screen

```
I/O UOP In                          JOINT  10 %
                                           1/3
   #      RANGE     RACK   SLOT   START   STAT.
   1  UI[  1-  8]     0      1       1   ACTIV
   2  UI[  9- 16]     0      1       9   ACTIV
   3  UI[ 17- 18]     0      1      17   ACTIV

[ TYPE ] MONITOR IN/OUT  DELETE   HELP
```

To return to the list screen, press F2,MONITOR

7 Manipulating the I/O assignment screen

a) Place the cursor on “Range,” and specify the range of signals to be assigned.

b) Line division is performed automatically according to the specified range.

c) Enter appropriate values for “Rack,” “Slot,” and “Start point.”

d) When the entered values are valid, abbreviation “PEND” is displayed in “Status.”
If any entered value is invalid, abbreviation “INVAL” is displayed in “Status.”

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

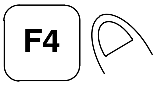
The abbreviations that will appear in “Status” mean the following:

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

NOTE In default setting, input pins 1 to 18 and output pins 1 to 20 is assigned to the peripheral I/O.

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

```
DETAIL   HELP >
```



Peripheral I/O detail screen

```
I/O UOP In                       JOINT  10 %
Port Detail                             1/1

    User Opr. Panel Input: [    1]


  1 Comment : [*IMSTP                  ]




[ TYPE ]  PRV-PT  NXT-PT
```

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

9 To add a comment:

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

```
            JOINT  30 %

1  [                ]
```

b Select the method of naming the comment.

c Press the appropriate function keys to add the comment.

d When you are finished, press the ENTER key.

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

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

11 When you are finished, press the PREV key to return to the selection screen.

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

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

> **CAUTION**
> In the first power–up after I/O re–allocation, power failure recovery would not be executed even if it is enabled.

CAUTION

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

CAUTION

When peripheral I/O signals are allocated to the CRM79 interface, do not use the following signals because they are given the same physical numbers as for the peripheral I/O signals:
SDI [101] to SDI [118]
SDO [101] to SDO [120]

CAUTION

When peripheral I/O signals are allocated to the JD1B interface, do not use the following signals because they are given the same physical numbers as for the peripheral I/O signals:
SDI [1] to SDI [18]
SDO [1] to SDO [20]

3.4 Operator's Panel I/O

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

For the operator's panel I/O, the signal numbers cannot be mapped (redefined). Sixteen input and sixteen output signals are defined as standard. For the definition of the signals of the operator's panel I/O, see Fig. 3–12.

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

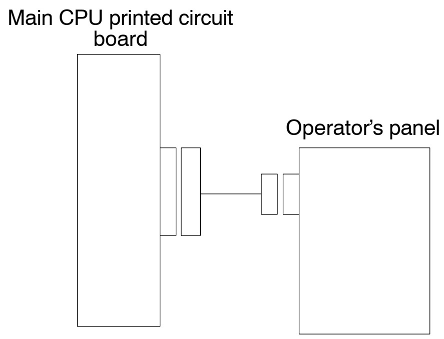
The operator's panel is enabled when the following operator's panel enable conditions are satisfied:

- The enable switch on the teach pendant is set to off.
- The remote signal (SI[2]) is off. (For how to turn the remote signal on and off, see the description of #139–3–1 in Section 3.16, "SYSTEM CONFIG MENU.")
- The *SFSPD input of the peripheral device I/O is on.

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

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

Figure 3–16. Operator's Panel I/O



Logical number	Operator's panel input
SI 0	
SI 1	FAULT_RESET
SI 2	REMOTE
SI 3	*HOLD
SI 4	USER#1
SI 5	USER#2
SI 6	START
SI 7	

Logical number	Operator's panel output
SO 0	REMOTE LED
SO 1	CYCLE START
SO 2	HOLD
SO 3	FAULT LED
SO 4	BATTERY ALARM
SO 5	USER#1
SO 6	USER#2
SO 7	TPENBL

Table 3–1. Operator's Panel Input Signals

Input signal	Description
***HOLD** SI [3] Always enabled. Not provided for the operator's panel.	The temporary stop (hold) signal specifies temporary stop of the program. The *HOLD signal is on in the normal status. When this signal is turned off: ● The robot operation being executed is decelerated, then stopped. ● The program being executed is temporarily stopped.
FAULT_RESET SI [2] Always enabled. Not provided for the operator's panel.	The alarm release (fault reset) signal releases the alarm state. When the servo power is off, this signal turns on the servo power. In this case, the alarm state is not released until the servo power is turned on.

Table 3–1. (Cont'd) Operator's Panel Input Signals

Input signal	Description
REMOTE SI [2] Always enabled. Not provided for the operator's panel.	The remote signal (remote) switches between remote mode and local mode of the system. In remote mode (SI[2] = on), when the remote conditions are satisfied, a program can be started using the peripheral device I/O. In local mode (SI[2] = off), when the operator's panel enable conditions are satisfied, a program can be started from the operator's panel. To turn the remote signal (SI[2]) on and off, set Remote/Local setup on the system config menu. For details, see Section 3.16, "SYSTEM CONFIG MENU."
START SI [6] Enabled in the operator's panel enable state. option	The start signal starts the currently selected program using the teach pendant from the line to which the cursor is positioned or restarts a temporarily stopped program. This signal functions at its falling edge when turned off after being turned on.

Table 3–2. Operator's Panel Output Signals

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

Procedure 3–6 Displaying the operator's panel I/O

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

Step

1 Press MENUS to display the screen menu.

2 Select "5 I/O."

3 Press F1 [TYPE] to display the screen switching menu.

4 Select "SOP."

```
4 ALARM
5 I/O
6 SETUP
```

MENUS

```
SOP
TYPE
```

F1

Operator's panel I/O list screen

```
I/O Sop Out                            JOINT 30%
     #          STATUS
  SO[0]           ON   [Remote LED         ]
  SO[1]          OFF   [Cycle start        ]
  SO[2]          OFF   [Hold               ]
  SO[3]           ON   [Fault LED          ]
  SO[4]           ON   [Butt alarm         ]
  SO[5]          OFF   [User LED#1         ]
  SO[6]          OFF   [User LED#2         ]
  SO[7]           ON   [TP enabled         ]
  SO[8]          OFF   [                   ]
  SO[9]          OFF   [                   ]
```

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

```
[ TYPE ]          IN/OUT
```

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

3.5 I/O Link Screen

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

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

- I/O link list screen
- Model B unit list screen
- Signal count setting screen

3.5.1 I/O Link list screen

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

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

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

```
 I/O Link Device                  JOINT 100%

   Device name    Comment    Rack  Slot
  1 PrcI/O CA  [            ]  0     1
  2 Model B    [            ]  1     0
  3 Model A    [            ]  2     0
  4 Model A    [            ]  3     0

[TYPE]         DETAIL      CLR_ASG
```

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

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

Word on TP	Device
PrcI/O AA	Process I/O Board AA
PrcI/O AB	Process I/O Board AB
PrcI/O BA	Process I/O Board BA
PrcI/O BB	Process I/O Board BB
PrcI/O CA	Process I/O Board CA
PrcI/O CB	Process I/O Board CB
PrcI/O DA	Process I/O Board DA
PrcI/O EA	Process I/O Board EA
PrcI/O EB	Process I/O Board EB
PrcI/O FA	Process I/O Board FA
PrcI/O GA	Process I/O Board GA
R–J2 Mate	R–J2 Mate. Slave Mode
Weld I/F	Weld Interface Board
Others	Other I/O devices except above devices

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

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

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

F5 (CLR_ASG) is described later.

3.5.2 Model B unit list screen

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

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

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

[TYPE]         LIST  [CHOICE] CLR_ASG
```

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

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

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

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

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

[TYPE]         LIST  [CHOICE] CLR_ASG
```

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

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

[TYPE]         LIST  [CHOICE] CLR_ASG
```

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

- BMD88A1
- BID16A1
- BOD16A1
- BOA12A1

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

- BMD88P1
- BID16P1
- BOD16P1
- BIA16A1
- BMD88Q1

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

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

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

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

3.5.3 Signal count setting screen

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

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

```
 I/O Link Device                 JOINT 100%

  90-30 PLC     Rack 1  Slot 1

       Pore name                 Points
   1 Digital Input:                0
   2 Digital Output:               0


[TYPE]         LIST           CLR_ASG
```

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

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

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

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

Explanation of F5 (CLR_ASG)

When the number of signals is set for a model–B unit or I/O unit on the I/O link screen, the I/O assignment may differ from the standard assignment according to the setting procedure. The following operation can set all I/O assignment to the standard settings. When setting the number of signals for a model–B unit or I/O unit for the first time, perform the following operation.

* When the unit is used with non–standard settings, this operation deletes the assignment information.

Press F5 (CLR_ASG). The following message appears.

```
Clear all assignments?
                                    YES       NO

                                    F4        F5
```

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

3.6 I/O Connection Function

The I/O connection function enables the RDI/SDI status to be output to SDO/RDO to report the signal input status to external devices.

The standard input/output ranges are shown below:

- RDI[mmm] → SDO[nnn].
  (1<=mmm<=8, 0<=nnn<=256)
- SDI[iii] → RDO[jjj].
  (0<=iii<=256, 1<=jjj<=8)
- SDI[kkk] → SDO[lll].
  (0<=kkk<=256, 0<=lll<=256)

Explanation of the function/settings

Assign signals and enable or disable each assignment on Interconnect.

The following three types of screens are available:

- DI DO connection setting screen (RDI → SDO)
- DI DO connection setting screen (SDI → RDO)
- DI DO connection setting screen (SDI → SDO)

DI DO connection setting screen (RDI → SDO)

DI DO connection setting screen (SDI → RDO)

Assign SDI signal numbers to RDO1 to RDO8. Whether to enable or disable each assignment can also be set.

DI DO connection setting screen (SDI → RDO)

Assign SDI signal numbers to RDO1 to RDO8. Whether to enable or disable each assignment can also be set.

DI DO connection setting screen (SDI → SDO)

Assign an SDO signal number to each SDI number.

Whether to enable or disable each assignment can also be set.

Example) When ”ENABLE DI[2] → RO[3] → RDO[3]” is set, the status of SDI[2] is output to RDO[3].

NOTE When SDI[i] → SDO[j] is set and this assignment is enabled, the status of SDI[i] is output to SDO[j] at regular intervals. Therefore, if the contents of SDO[j] are changed using the TP or a program, the change is not reflected.

NOTE Whether to enable or disable each assignment can be changed only on the setting screen, described above.

NOTE When different multiple input signals are assigned to the same output signal, the status of each input signal is output. For example, assume that the following settings are made:

ENABLE DI[1] → RDO[1] → SDO[1]
ENABLE DI[2] → RDO[1] → SDO[1]

In this case, when the status of RDI[1] is ON and the status of RDI[2] is OFF, the SDO[1] output will be unpredictable. (SDO[1] alternately indicates ON and OFF in practice.)

Procedure 3–7 Setting the I/O connection function

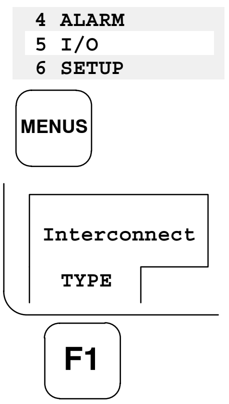
Step

1 Press MENUS to display the screen menu.

2 Select ”5 I/O.”

3 Press F1, [TYPE] to display the screen switching menu.

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

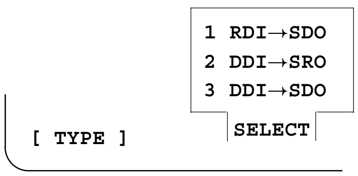


DI DO connection setting screen (RDI SDO)

```
INTERCONNECT                     JOINT 100%
                                        1/8
No.    Enb/Disabl    INPUT     OUTPUT
1      ENABLE        RI  [1] → DO  [0]
2      DISABLE       RI  [2] → DO  [0]
3      DISABLE       RI  [3] → DO  [0]
4      DISABLE       RI  [4] → DO  [0]
5      DISABLE       RI  [5] → DO  [0]
6      DISABLE       RI  [6] → DO  [0]
7      DISABLE       RI  [7] → DO  [0]
8      DISABLE       RI  [8] → DO  [0]

[TYPE]           [SELECT]  ENABLE  DISABLE
```

5 Press SELECT.

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



DI RO connection setting screen (SDI RDO)

```
INTERCONNECT                     JOINT 100%
                                        1/8
No.    Enb/Disabl    INPUT     OUTPUT
1      ENABLE        DI  [0] → RO  [1]
2      DISABLE       DI  [0] → RO  [2]
3      DISABLE       DI  [0] → RO  [3]
4      DISABLE       DI  [0] → RO  [4]
5      DISABLE       DI  [0] → RO  [5]
6      DISABLE       DI  [0] → RO  [6]
7      DISABLE       DI  [0] → RO  [7]
8      DISABLE       DI  [0] → RO  [8]

[TYPE]           [SELECT]  ENABLE  DISABLE
```

3.7 Setting Automatic Operation

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

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

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

- The teach pendant enable switch is off.
- The remote signal (SI[2]) is on. (For how to turn the remote signal (SI[2]) on and off, see the description of Remote/Local setup in Section 3.16, "SYSTEM CONFIG MENU.")
- The *SFSPD signal of the peripheral I/O is set on.
- The ENBL signal of the peripheral I/O is set on.
- System variable $RMT_MASTER is set to 0 (peripheral equipment).

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

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

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

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

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

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

3.7.1 Robot service request (RSR)

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

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

NOTE In the initial status, the peripheral device input signal (UI) is disabled. To enable the signal, select TRUE for Enable UI signals on the system setting screen.

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

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

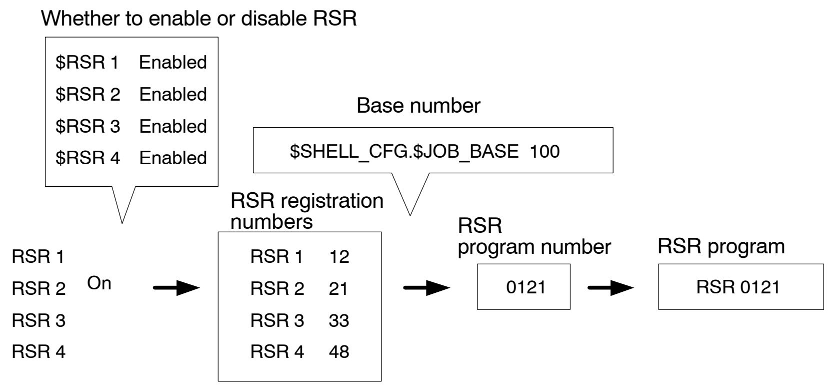
The base number is set in $SHELL_CFG.$JOB_BASE and can be changed using Base number on the RSR setting screen or a program parameter instruction.

3 A pulse of the RSR acknowledgment output (ACK1 to ACK8) corresponding to the RSR1 to RSR8 input is output. When the ACK1 to ACK8 signal is output, the control unit accepts another RSR input.

4 When a program is in the terminated state, the selected program is started. When another program is being executed or is temporarily stopped, the request (job) is entered into the queue and the selected program is started when the program being executed terminates.
  Jobs (RSR programs) are executed in the order in which they are entered into the queue.

5 Waiting programs are canceled (cleared) by the cycle stop signal (CSTOPI input) or upon forced program termination.

Figure 3–17. Robot Service Request



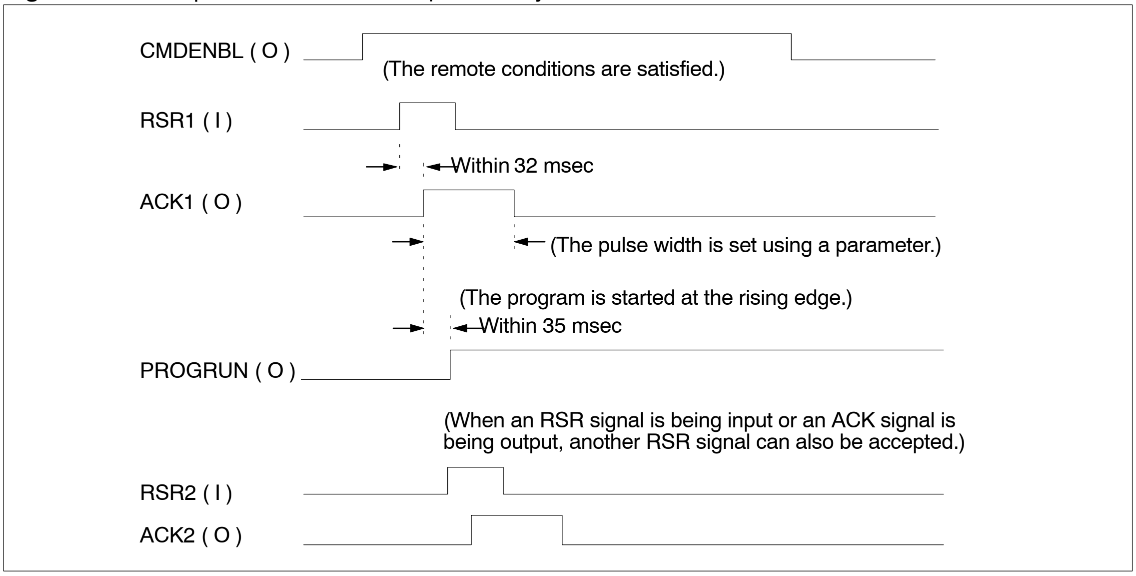
1 Inputs the RSR2 signal.
2 Checks whether RSR2 is enabled or disabled.
3 Starts the RSR program having the selected RSR program number.

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

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

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

Figure 3–18. Sequence of Automatic Operation by RSR



Set RSR for SETUP RSR/PNS on the RSR setting screen.

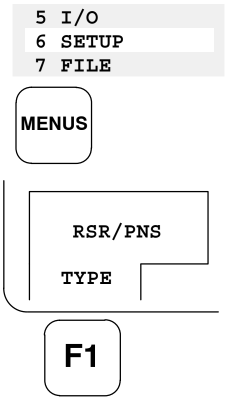
Table 3–3. RSR Setting Items

Item	Description
Program select mode	Select either the RSR or PNS automatic operation function. Both functions cannot be used simultaneously. After changing this setting, to enable the change, turn the power off, then on again.
RSR1 to 8 program number	Specifies whether to enable or disable RSR1 to RSR8 and the RSR registration numbers. When an RSR signal is disabled and the specified signal is input, the program is not started. Setting whether to enable or disable each RSR is stored in system variable $RSR1 to $RSR8.
Base number	Added to the RSR registration number to obtain the RSR program number.
Acknowledge function	Sets whether to output RSR acknowledgment signals (ACK1 to ACK8).
Acknowledge pulse width	Sets the pulse output period (unit: msec) when the output of each RSR acknowledgment signal (ACK1 to ACK8) is enabled.

Procedure 3–8 Setting RSR

Step

1 Press MENUS to display the screen menu.
2 Select SETUP.
3 Press F1, [TYPE] to display the screen switching menu.
4 Select ”RSR/PNS.” The RSR/PNS setting screen appears.
5 Position the carsor to ”Program select mode”. Press F4 [CHOICE] and select RSR, then press F3 DETAIL.



RSR

```
 RSR/PNS                                JOINT 30%
                                            1/11
   RSR                                   [ RSR]
 1 RSR1 program number    [ENABLE]  [   12]
 2 RSR2 program number    [ENABLE]  [   21]
 3 RSR3 program number    [ENABLE]  [   33]
 4 RSR4 program number    [ENABLE]  [   49]
 5 RSR5 program number    [ENABLE]  [   50]
 6 RSR6 program number    [ENABLE]  [   60]
 7 RSR7 program number    [ENABLE]  [   70]
 8 RSR8 program number    [ENABLE]  [   80]
 9 Base number                      [  100]
10 Acknowledge function             [TRUE]
11 Acknowledge pulse width (msec)   [  200]

[TYPE]
```

6 Position the cursor to the target item and enter a value.
7 After changing PNS to RSR, to enable the change, turn the power off, then on again.

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

3.7.2 Program number selection (PNS)

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

Step

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

NOTE In the initial status, the peripheral device input signal (UI) is disabled. To enable the signal, select TRUE for Enable UI signals on the system setting screen.

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

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

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

When a zero is input by the PNS1 to PNS8 inputs, the system enters the status in which no program is selected on the teach pendant.

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

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

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

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

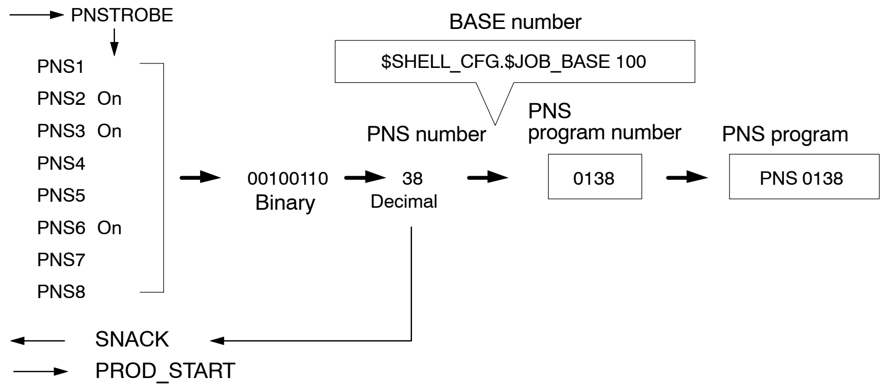
5 The control unit receives the PROD_START input and starts the program.

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

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

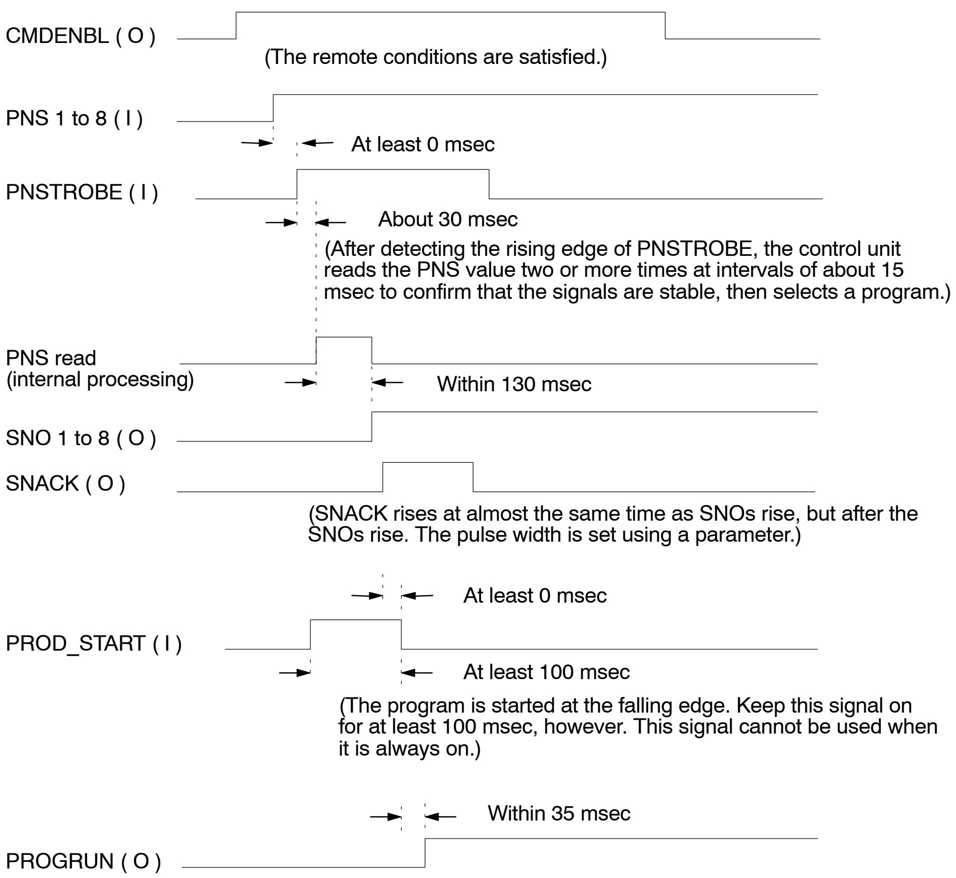
The CMDENBL output is provided to indicate whether the above conditions are satisfied.

Figure 3–19. Program Number Selection



1. The PNSTROBE signal is input.
2. Signals PNS1 to PNS8 are read and the value is converted into a decimal number.
3. The PNS program having the specified PNS program number is selected.
4. When the PROD–START signal goes low, the selected PNS program is started.

Figure 3–20. Sequence of Automatic Operation by PNS



Setting the PNS function

Set the PNS function on the PNS setting screen [6 (SETUP). RSR/PNS]. Refer to Table 3–4.

Table 3–4. Setting the PNS function

ITEMS	DESCRIPTIONS
Program select mode	Select either the RSR or PNS automatic operation function. These functions cannot be used simultaneously. After changing this setting, to enable the change, turn the power off, then on again.
Base number	The reference number is added to the PNS number to obtain a PNS program number.
Acknowledge pulse width (msec)	Sets the pulse output period (unit: msec) of the PNS acknowledgment signal (SNACK).

Procedure 3–9 Setting the PNS function

Step

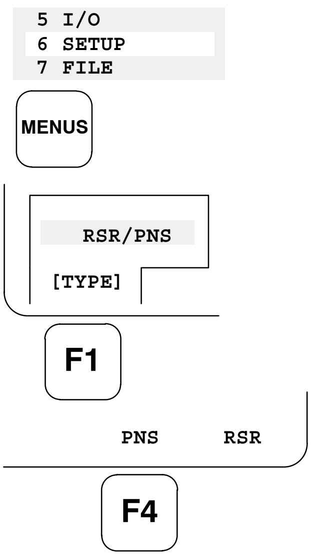
1 Press the MENUS key. The screen menu is displayed.

2 Select “6 (SETUP).”

3 Press the F1 key, TYPE. The screen change menu is displayed.

4 Select RSR/PNS. RSR/PNS Setting screen is displayed.

5 Position the carsor to ”Program select mode”. Press F4 [CHOICE] and select RSR, then press F3 DETAIL.



PNS Setting Screen

```
RSR/PNS                                JOINT 30%
                                             1/2
   PNS
 1 Base number                            [ 100]
 2 Acknowledge pulse width (msec)         [ 200]



[TYPE]
```

6 Place the cursor on a desired field and enter a value.

7 After changing RSR to PNS, to enable the change, turn the power off, then on again.

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

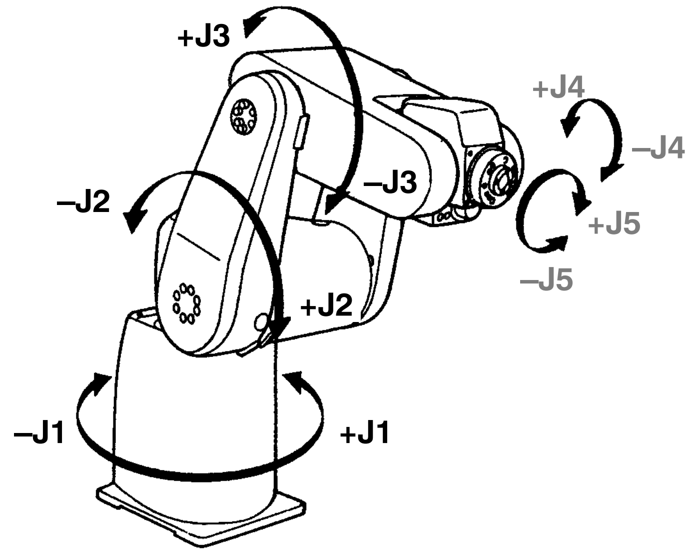
3.8 Setting Coordinate Systems

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

Joint coordinate system

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

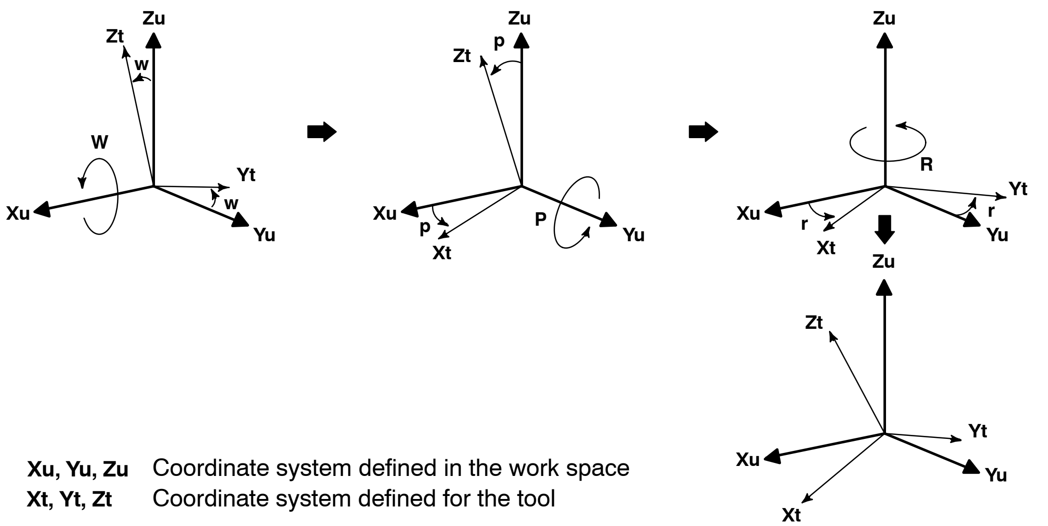
Figure 3–21. Joint Coordinate System



Cartesian coordinate system

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

Figure 3–22. Meaning of (w, p, r)



Xu, Yu, Zu Coordinate system defined in the work space
Xt, Yt, Zt Coordinate system defined for the tool

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

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

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

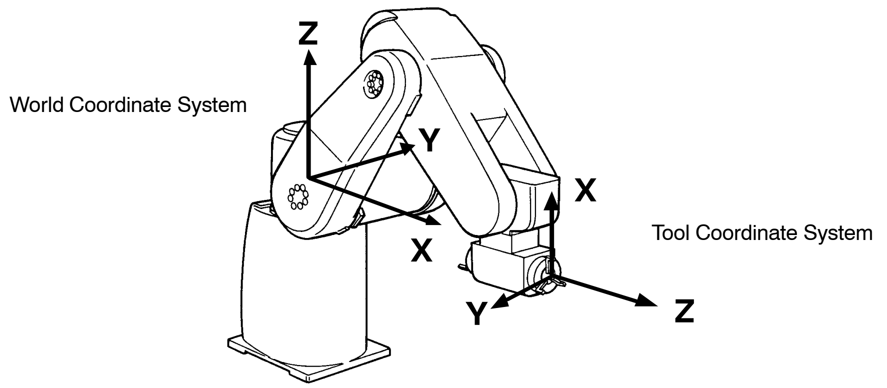
Tool coordinate system

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

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

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

Figure 3–23. World and Tool Coordinate Systems



User coordinate system

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

WARNING

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

Jog coordinate system

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

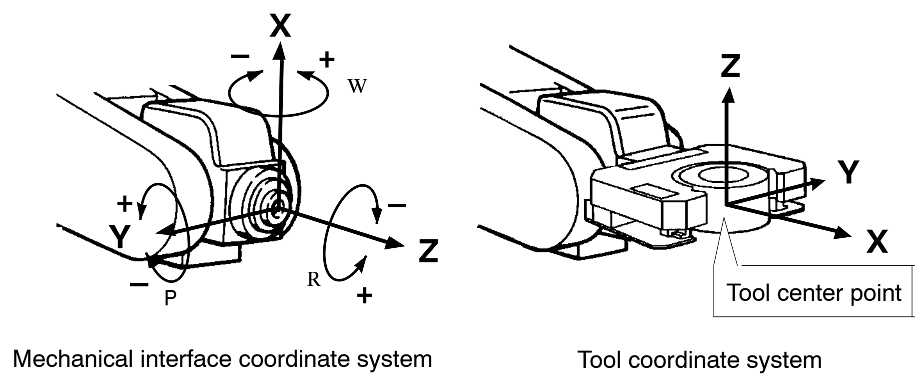
3.8.1 Setting a tool coordinate system

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

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

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

Figure 3–24. Tool Coordinate System



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

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

The tool frame can be set by three following methods.

Three Point Method (TCP auto set) [Optional function]

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

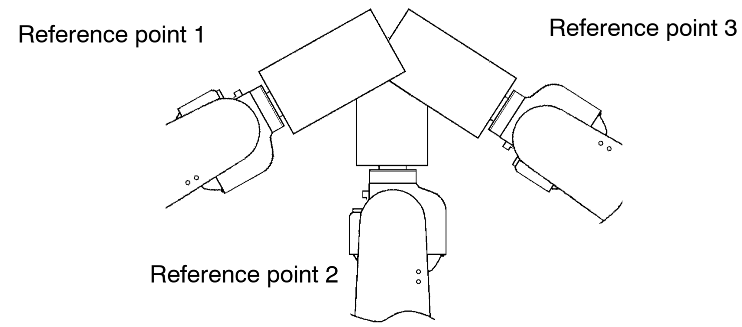
As a result, the location of TCP is automatically calculated.

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

The three approach point must be taught with the tool touching a common point from three different approach directions.

In the three point method, only the tool center point (x,y,z) can be set. The setting value of the tool orientation (w, p, r) is the standard value(0,0,0). The tool orientation should be defined by the six point method or direct list method after the location is set. (The orientation of (w, p, r) is not necessarily defined.)

Figure 3–25. TCP auto set by the three point method



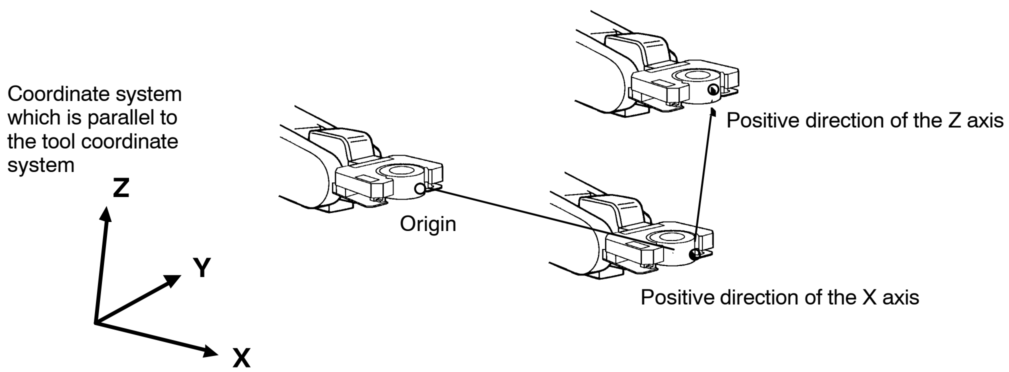
Six Point Method

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

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

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

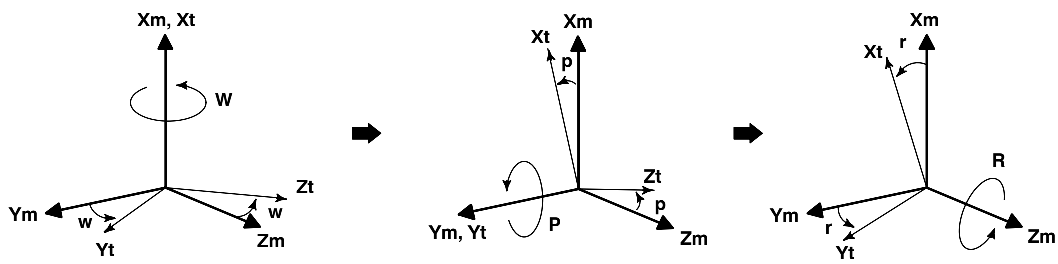
Figure 3–26. Six point method



Direct list method

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

Figure 3–27. Meaning of (w, p, r) used in direct teaching method



Xm, Ym, Zm Mechanical interface coordinate system
Xt, Yt, Zt Tool coordinate system

Procedure 3–10 TCP auto set (Three Point Method)

Step

1 Press the MENUS key. The screen menu is displayed.

2 Select “6 (SETUP).”

3 Press the F1 key, TYPE. The screen change menu is displayed.

4 Select Frames.

5 Press F3, OTHER and then select Tool Frame. Tool frame list screen is displayed.

Tool frame list screen

```
 5 I/O
 6 SETUP
 7 FILE
```

MENUS

```
   Frames

 [TYPE]
```

F1

```
 SETUP Frames                     JOINT  30 %
   Tool Frame Setup/ Direct Entry    1/9

         X       Y       Z    Comment
    1:    0.0    0.0     0.0 *************
    2:    0.0    0.0     0.0 *************
    3:    0.0    0.0     0.0 *************
    4:    0.0    0.0     0.0 *************
    5:    0.0    0.0     0.0 *************
    6:    0.0    0.0     0.0 *************
    7:    0.0    0.0     0.0 *************
    8:    0.0    0.0     0.0 *************
    9:    0.0    0.0     0.0 *************
 Active TOOL $MNUTOOLNUM[1]=1
 [ TYPE ] DETAIL [OTHER ]  CLEAR  SETIND
```

```
                  1 Tool Frame
                  2 Jog Frame
                  3 User Frame

 [ TYPE ] DETAIL  OTHER
```

F3

6 Move the cursor to the line of the tool frame number you want to set.

7 Press F2,DETAIL.The tool frame setup screen of the selected frame number is displayed.

```
 [ TYPE ] DETAIL [OTHER ]
```

F2

8 Press F2,METHOD and then select Three Point.

Tool frame setup screen (Three Point Method)

```
          1 Three Point
          2 Six Point
          3 Direct Entry

 [ TYPE ] METHOD  FRAME
```

F2

```
 SETUP Frames                     JOINT  30 %
  Tool Frame Setup/ Three Point      1/4
  Frame Number: 1
    X:     0.0   Y:     0.0   Z:     0.0
    W:     0.0   P:     0.0   R:     0.0
   Comment:                    TOOL 1
    Approach point 1:     UNINIT
    Approach point 2:     UNINIT
    Approach point 3:     UNINIT
 Active TOOL $MNUTOOLNUM[1]=1
 [ TYPE ][METHOD] FRAME
```

9 To add a comment:

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

```
              JOINT   30 %
 1  [                ]
```

ENTER

b Select the method of naming the comment.

c Press the appropriate function keys to add the comment.

d When you are finished, press ENTER key.

10 Record each approach point:

a Move the cursor to each Approach point.

b Jog the robot to the position you want to record.

c Press and hold the SHIFT key and press F5,RECORD to record the data of current position as the reference position. As for the taught reference point, RECORDED is displayed.

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

```
   Approach point 3:

 FRAME   MOVE_TO RECORD
```

SHIFT F5

```
 SETUP Frames                      JOINT   30 %

    Approach point 1:      RECORDED
    Approach point 2:      RECORDED
    Approach point 3:      UNINIT

 [ TYPE ] [METHOD]  FRAME    MOVE_TO RECORD
```

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

```
 SETUP Frames                      JOINT   30 %
  Tool Frame Setup/ Three Point     4/4
  Frame Number: 1
    X:  100.0   Y:    0.0   Z:  120.0
    W:    0.0   P:    0.0   R:    0.0
   Comment:              TOOL 1
    Approach point 1:     USED
    Approach point 2:     USED
    Approach point 3:     USED

 [ TYPE ] [METHOD]  FRAME    MOVE_TO RECORD
```

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

```
 FRAME   MOVE_TO RECORD
```

SHIFT F4

12 To see each recorded position data, move the cursor to each reference position item and press the ENTER key. The position detail screen of each position data is displayed. To return to the previous screen, press the PREV key.

13 To display the tool frame list screen, press the PREV key. You can see the settings(x,y,z,and comment) for all tool frames.

PREV

```
SETUP Frames                   JOINT  30 %
  Tool Frame Setup/ Direct Entry    1/9

          X      Y       Z    Comment
    1:  100.0   0.0  120.0 TOOL1
    2:    0.0   0.0    0.0 *************
    3:    0.0   0.0    0.0 *************
    4:    0.0   0.0    0.0 *************
    5:    0.0   0.0    0.0 *************
    6:    0.0   0.0    0.0 *************
    7:    0.0   0.0    0.0 *************
    8:    0.0   0.0    0.0 *************
    9:    0.0   0.0    0.0 *************

 [ TYPE ] DETAIL [OTHER ]  CLEAR  SETIND
```

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

```
FRAME   MOVE_TO RECORD
```

F5

> **CAUTION**
> If you do not press F5, SETIND, the tool frame will not be effective.

NOTE To select the number of a coordinate system to be used, the jog menu can also be used. See Section 5.2.3 "Moving the robot jog feed."

> **CAUTION**
> After all coordinate systems are set, the setting information should be saved in external storage (floppy disk, for example) in case the information needs to be re–loaded. Otherwise, the current setting information would be lost when it is changed.

15 To delete the data of the set frame, move the cursor to the desired frame and press F4,CLEAR.

```
[OTHER ]  CLEAR  SETIND
```

F4

Procedure 3–11 Setting Up Tool Frame Using the Six Point Method

Step

1 Display the tool frame list screen (Refer to the three point method).

```
SETUP Frames                        JOINT  30 %
 Tool Frame Setup/ Direct Entry    2/5

      X       Y       Z     Comment
  1:  100.0   0.0  120.0         TOOL1
  2:    0.0   0.0    0.0 *************
  3:    0.0   0.0    0.0 *************
  4:    0.0   0.0    0.0 *************
  5:    0.0   0.0    0.0 *************
  6:    0.0   0.0    0.0 *************
  7:    0.0   0.0    0.0 *************
  8:    0.0   0.0    0.0 *************
  9:    0.0   0.0    0.0 *************
Active TOOL $MNUTOOLNUM[1]=1
[ TYPE ] DETAIL [OTHER ]  CLEAR  SETIND
```

2 Move the cursor to the tool frame number line you want to set.

3 Press F2,DETAIL. The tool frame setup screen of the selected frame number is displayed.

4 Press F2,METHOD.

5 Select Six Point. The tool frame setup / six point screen is displayed.

Tool frame setup screen (Six Point Method)

```
[ TYPE ] DETAIL [OTHER ]

F2
```

```
1 Three Point
2 Six Point
3 Direct Entry

[ TYPE ]  METHOD  FRAME

F2
```

```
SETUP Frames                        JOINT  30 %
 Tool Frame Setup/ Six Point       1/7
 Frame Number: 2
   X:    0.0   Y:    0.0   Z:    0.0
   W:    0.0   P:    0.0   R:    0.0
  Comment:********************
   Approach point 1:      UNINIT
   Approach point 2:      UNINIT
   Approach point 3:      UNINIT
   Orient Origin Point:   UNINIT
   X Direction Point:     UNINIT
   Z Direction Point:     UNINIT
Active TOOL $MNUTOOLNUM[1]=1
[ TYPE ][METHOD]  FRAME
```

6 Add a comment and teach the reference point. For details, refer to TCP auto set (Three Point Method).

a Press and hold the SHIFT key and press F5,RECORD to record the data of current position as the reference position. As for the taught reference point, RECORDED is displayed.

```
FRAME   MOVE_TO RECORD

SHIFT          F5
```

```
SETUP Frames                        JOINT  30 %
   Approach point 1:      RECORDED
   Approach point 2:      RECORDED
   Approach point 3:      RECORDED
   Orient Origin Point:   RECORDED
   X Direction Point:     UNINIT
   Z Direction Point:     UNINIT
Active TOOL $MNUTOOLNUM[1]=1
[ TYPE ][METHOD]  FRAME   MOVE_TO RECORD
```

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

```
SETUP Frames                 JOINT  30 %
 Tool Frame Setup/ Six Point       1/7
 Frame Number: 2
   X:  200.0   Y:     0.0   Z:  255.5
   W:  -90.0   P:     0.0   R:  180.0
  Comment:                 TOOL2
   Approach point 1:      USED
   Approach point 2:      USED
   Approach point 3:      USED
   Orient Origin Point:   USED
   X Direction Point:     USED
   Z Direction Point:     USED
Active TOOL $MNUTOOLNUM[1]=1
[ TYPE ][METHOD] FRAME
```

7 Press the PREV key. The tool frame list screen is displayed. You can see all the tool frame settings.

PREV

```
SETUP Frames                 JOINT  30 %
 Tool Frame Setup/ Direct Entry    2/9

       X      Y      Z    Comment
  1:  100.0  30.0  120.0        [TOOL1]
  2:  200.0   0.0  255.0        [TOOL2]
  3:    0.0   0.0    0.0        [      ]
  4:    0.0   0.0    0.0        [      ]
  5:    0.0   0.0    0.0        [      ]
  6:    0.0   0.0    0.0        [      ]
  7:    0.0   0.0    0.0        [      ]
  8:    0.0   0.0    0.0        [      ]
  9:    0.0   0.0    0.0        [      ]
Active TOOL $MNUTOOLNUM[1]=1
[ TYPE ] DETAIL [OTHER ]  CLEAR  SETIND
```

8 To make the set tool frame effective, press F5 (SETIND), then enter the frame number.

```
[OTHER ]  CLEAR  SETIND
```

F5

CAUTION
If you do not press F5, SETIND, the tool frame will not be effective.

NOTE To select the number of a coordinate system to be used, the jog menu can also be used. See Section 5.2.3 "Moving the robot jog feed."

CAUTION
After all coordinate systems are set, the setting information should be saved in external storage (floppy disk, for example) in case the information needs to be re–loaded. Otherwise, the current setting information would be lost when it is changed.

To delete the data of the set frame, move the cursor to the desired frame and press F4,CLEAR.

```
[OTHER ]  CLEAR  SETIND
```

F4

Procedure 3–12 Setting Up Tool Frame Using the Direct List Method

Step

1 Display the tool frame list screen (Refer to the three point method).

```
 SETUP Frames                       JOINT  30 %
  Tool Frame Setup/ Direct Entry    3/9

        X      Y      Z    Comment
   1:  100.0  30.0  120.0         TOOL1
   2:  200.0   0.0  255.0         TOOL2
   3:    0.0   0.0    0.0 *************
   4:    0.0   0.0    0.0 *************
   5:    0.0   0.0    0.0 *************
   6:    0.0   0.0    0.0 *************
   7:    0.0   0.0    0.0 *************
   8:    0.0   0.0    0.0 *************
   9:    0.0   0.0    0.0 *************
 Active TOOL $MNUTOOLNUM[1]=1
 [ TYPE ] DETAIL [OTHER ]  CLEAR  SETIND
```

2 Move the cursor to the tool frame number line you want to set.

3 Press F2,DETAIL or press the ENTER key. The tool frame setup screen of the selected frame number is displayed.

4 Press F2,METHOD.

5 Select Direct Entry. Tool Frame Setup / Direct Entry screen is displayed.

Tool frame setup screen (Direct List Method)

```
 [ TYPE ] DETAIL [OTHER ]
```

F2

```
  1 Three Point
  2 Six Point
  3 Direct Entry

 [ TYPE ]  METHOD  FRAME
```

F2

```
 SETUP Frames                       JOINT  30 %
  User Frame Setup/ Direct Entry    1/7
  Frame Number: 3
    1  Comment:   ********************
    2       X:                      0.0
    3       Y:                      0.0
    4       Z:                      0.0
    5       W:                      0.0
    6       P:                      0.0
    7       R:                      0.0
            Configuration:  N D B, , 0
 Active TOOL $MNUTOOLNUM[1]=1
 [ TYPE ][METHOD] FRAME
```

6 Add a comment. Refer to TCP auto set (Three Point Method) for details.

```
              JOINT  30 %

  1  [                 ]
```

ENTER

7 Enter the coordinate values of the tool frame.
  a Move the cursor to each component.
  b Enter a new numerical value by using numerical keys.
  c Press the ENTER key. A new numerical value is set.

```
        JOINT  30 %

          0.0
```

3 5 0

```
SETUP Frames                      JOINT  30 %
 User Frame Setup/ Direct Entry    4/7
 Frame Number: 3
  1  Comment:                      TOOL3
  2      X:                          0.000
  3      Y:                          0.000
  4      Z:                        350.000
  5      W:                        180.000
  6      P:                          0.000
  7      R:                          0.000
  8      Configuration:  N D B, , 0
Active TOOL $MNUTOOLNUM[1]=1
[ TYPE ][METHOD] FRAME
```

8 To display the tool frame list screen, press the PREV key. You can see the settings of all the tool frame.

PREV

```
SETUP Frames                      JOINT  30 %
 Tool Frame Setup/ Direct Entry    3/5

        X       Y       Z     Comment
  1:  100.0   30.0   120.0          TOOL1
  2:  200.0    0.0   255.0          TOOL2
  3:    0.0    0.0   350.0          TOOL3
  4:    0.0    0.0     0.0 *************
  5:    0.0    0.0     0.0 *************
  6:    0.0    0.0     0.0 *************
  7:    0.0    0.0     0.0 *************
  8:    0.0    0.0     0.0 *************
  9:    0.0    0.0     0.0 *************
Active TOOL $MNUTOOLNUM[1]=1
[ TYPE ] DETAIL [OTHER ]  CLEAR  SETIND
```

9 To make the set tool frame effective, press F5 (SETIND), then enter the frame number.

```
[OTHER ]  CLEAR  SETIND
```

F5

> **CAUTION**
> If you do not press F5, SETIND, the tool frame will not be effective.

NOTE To select the number of a coordinate system to be used, the jog menu can also be used. See Section 5.2.3 ”Moving the robot jog feed.”

> **CAUTION**
> After all coordinate systems are set, the setting information should be saved in external storage (floppy disk, for example) in case the information needs to be re–loaded. Otherwise, the current setting information would be lost when it is changed.

To delete the data of the set frame, move the cursor to the desired frame and press F4,CLEAR.

```
[OTHER ]  CLEAR  SETIND
```

F4

3.8.2 Setting a user coordinate system

A user coordinate system is a Cartesian coordinate system defined for each work space by the user. If the coordinate system is not defined, the world coordinate system substitutes for it.

Define the user coordinate system by (x, y, z) indicating the position of the zero point and (w, p, r) indicating the angular displacement around the X–, Y–, and Z–axes on the world coordinate system.

The user coordinate system is used to specify a position register and execute the corresponding position register instruction and position comensation instruction. For the specification of the position register, see Section 7.4, “Position Register.” For the execution of the position register instruction, see Section 4.3.2, “Position Data.” For the execution of the position compensation instruction, see Section 4.3.5, “Additional Motion Instruction.”

> **CAUTION**
>
> If teaching is made by joint coordinates, changing the user coordinate system does not affect the position variables and position registers. When the robot is taught in the Cartesian format and the user coordinate system input option is not used, the position variables are not affected by the user coordinate systems. Note that both position variables and registers are affected by the user coordinate systems in other cases.

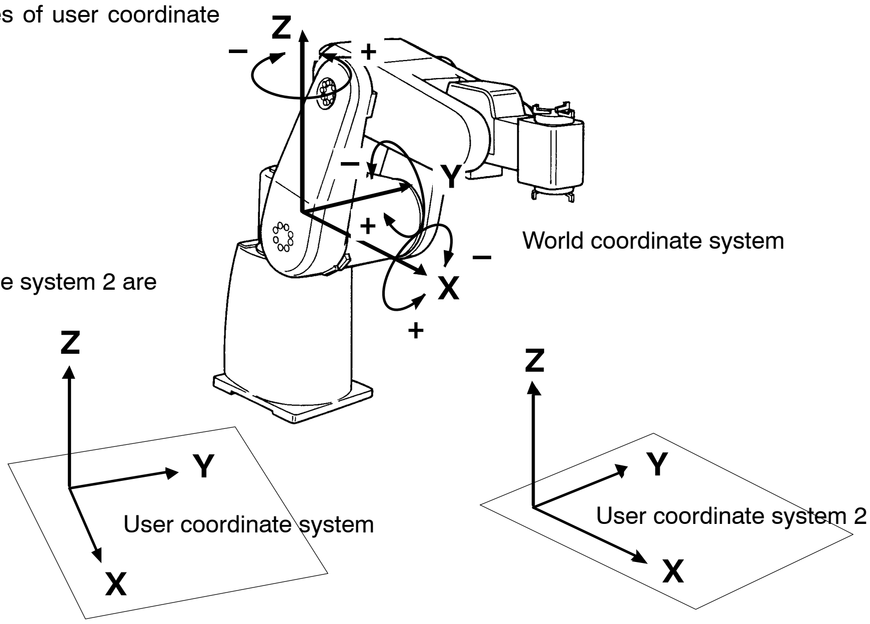
Figure 3–28. World and User Coordinate Systems

In the right figure, the coordinates of user coordinate system 1 are

x : 70.0 mm
y : –2500.0 mm
z : 200.0 mm
w : 0.0 deg
p : 0.0 deg
r : –30.0 deg

The coordinates of user coordinate system 2 are

x : 500.0 mm
y : 0.0 mm
z : –200.0 mm
w : 0.0 deg
p : 0.0 deg
r : 0.0 deg



The following system variables are changed by defining the user frame with the frame setup screen. Nine user coordinate systems can be defined. The desired one can be selected

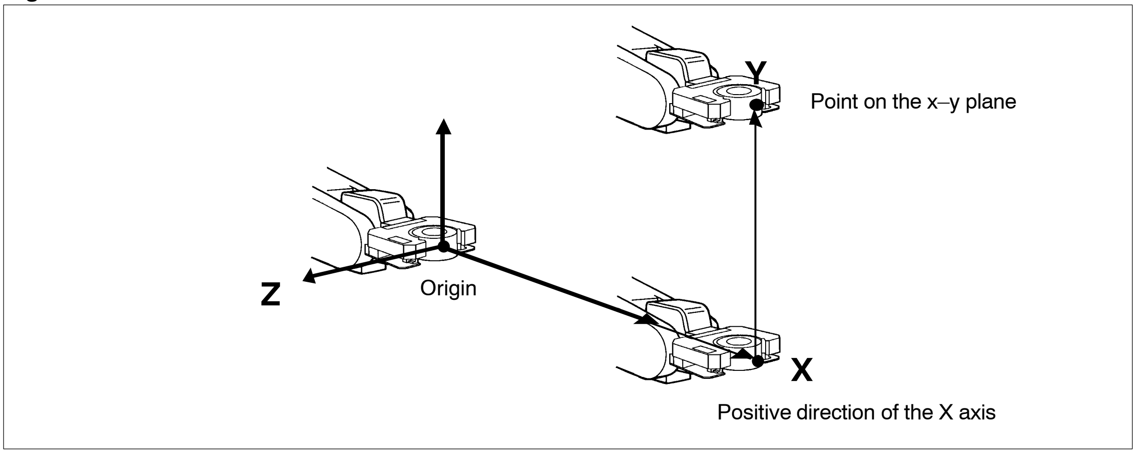
- $MNUFRAME [1, i] (Frame number i = 1 to 9) is set the value.
- $MNUFRAMENUM [1] is set the user frame number you want to use.

The user frame can be defined by the following three methods.

Three Point Method

Teach the following three points:the origin of the x–axis,the point which specifies the positive direction of the x–axis,and the point on the x–y plane.

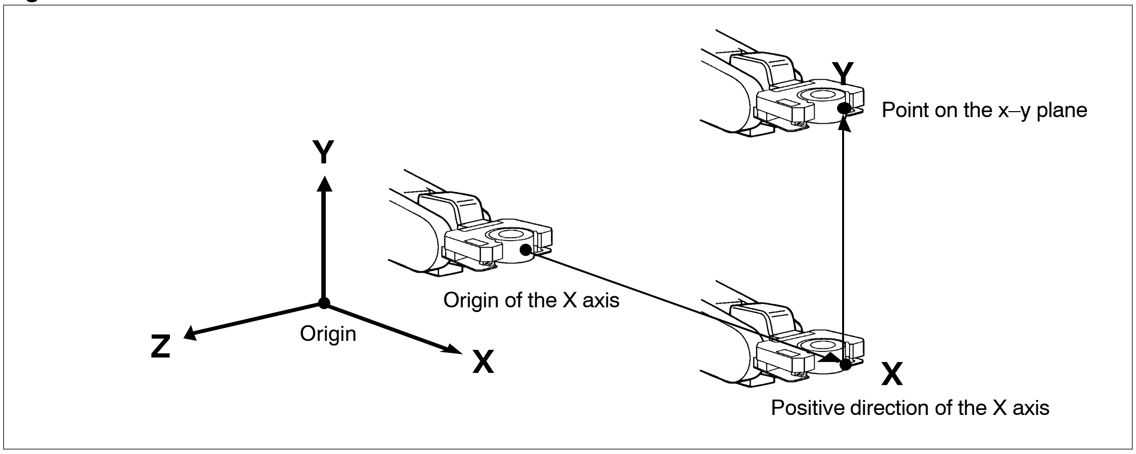
Figure 3–29. Three Point Method



Four Point Method

Teach the following four points:the origin of the x–axis parallel to the frame,the point which specifies the positive direction of the x–axis, a point on the x–y plane, and the origin of the frame.

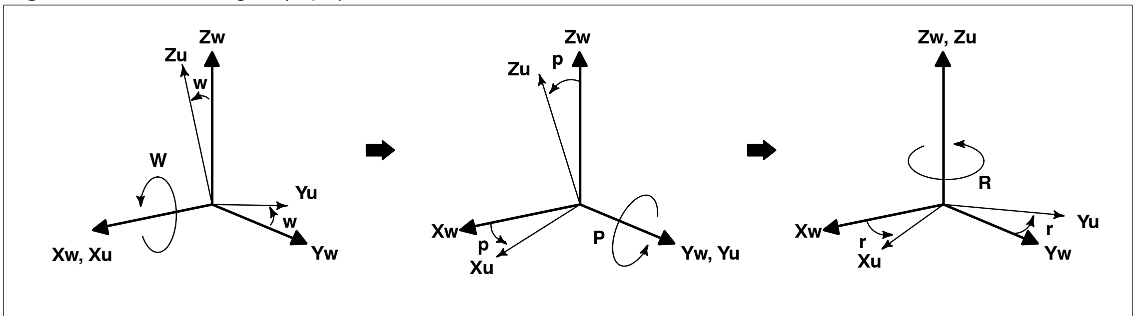
Figure 3–30. Four Point Method



Direct List Method

Enter the following values directly:the value (x,y,z) which specifies the origin of the user frame and is the coordinate values of the world frame and the rotating angle (w,p,r) around the x–,y–,and z–axis of the world frame.

Figure 3–31. Meaning of (w,p,r) used in direct list method



Xw, Yw, Zw World coordinate system
Xu, Yu, Zu User coordinate system

Procedure 3–13 Setting Up User Frame Using Three Point Method

Step

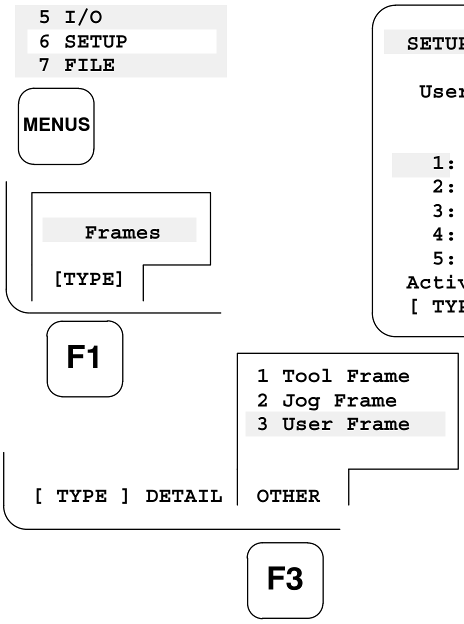
1 Press the MENUS key. The screen menu is displayed.

2 Select "6 (SETUP)."

3 Press the F1 key, TYPE. The screen change menu is displayed.

4 Select Frames.

5 Press F3, OTHER and then select User Frame. The user frame list screen is displayed.

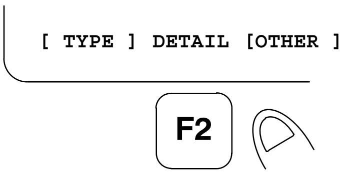


User frame list screen

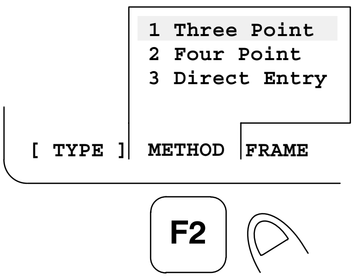
```
SETUP Frames                          JOINT  30 %

 User Frame Setup/ Direct Entry     1/5

       X       Y       Z    Comment
  1:    0.0    0.0     0.0 *************
  2:    0.0    0.0     0.0 *************
  3:    0.0    0.0     0.0 *************
  4:    0.0    0.0     0.0 *************
  5:    0.0    0.0     0.0 *************
Active UFRAME $MNUFRAMNUM[1]=0
[ TYPE ] DETAIL [OTHER ]  CLEAR  SETIND
```

6 Move the cursor to the line of the user frame number you want to set.

7 Press F2,DETAIL. The user frame setup screen of the selected frame number is displayed.



8 Press F2,METHOD and then select Three Point.



User frame setup screen (Three Point Method)

```
SETUP Frames                          JOINT  30 %
 User Frame Setup/ Three Point      1/4
 Frame Number: 1
   X:     0.0    Y:     0.0    Z:     0.0
   W:     0.0    P:     0.0    R:     0.0
  Comment:********************
   Orient Origin Point:  UNINIT
   X Direction Point:    UNINIT
   Y Direction Point:    UNINIT
Active UFRAME $MNUFRAMNUM[1]=0
[ TYPE ][METHOD] FRAME
```

9 To add a comment:

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

```
                 JOINT  30 %
    [            ]
```

ENTER

b Select the method of naming the comment.

c Press the appropriate function keys to add the comment.

d When you are finished, press ENTER key.

```
SETUP Frames                     JOINT  30 %

  Comment:     REFERENCE FRAME

[ TYPE ][METHOD]  FRAME
```

10 Record each approach point:

a Move the cursor to each Approach point.

b Jog the robot to the position you want to record.

c Press and hold the SHIFT key and press F5, RECORD to record the current position as the approach point. As for the taught reference point, RECORDED is displayed.

```
    X Direction Point:

FRAME   MOVE_TO RECORD
```

SHIFT F5

```
SETUP Frames                     JOINT  30 %
   Orient Origin Point:  RECORDED
   X Direction Point:    RECORDED
   Y Direction Point:    UNINIT
Active UFRAME $MNUFRAMNUM[1]=0
[ TYPE ][METHOD] FRAME   MOVE_TO RECORD
```

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

```
SETUP Frames                     JOINT  30 %
 User Frame Setup/ Three Point     4/4
 Frame Number: 1
   X: 1243.6    Y:    0.0   Z:   10.0
   W:    0.1    P:    2.3   R:    3.2
  Comment:     REFERENCE FRAME
   Orient Origin Point:  USED
   X Direction Point:    USED
   Y Direction Point:    USED
Active UFRAME $MNUFRAMNUM[1]=0
[ TYPE ][METHOD] FRAME   MOVE_TO RECORD
```

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

```
FRAME   MOVE_TO RECORD
```

SHIFT F4

12 To see each recorded position data, move the cursor to each reference position item and press the ENTER key. The position detail screen of each position data is displayed. To return to the previous screen, press the PREV key.

13 To display the user frame list screen, press the PREV key. You can see the settings for all user frames.

PREV

```
SETUP Frames                     JOINT  30 %
 User Frame Setup/ Three Point     1/9
       X      Y       Z    Comment
   1: 1243.6   0.0    43.8 REFERENCE FR>
   2:    0.0   0.0     0.0 *************
   3:    0.0   0.0     0.0 *************
   4:    0.0   0.0     0.0 *************
   5:    0.0   0.0     0.0 *************
   6:    0.0   0.0     0.0
   7:    0.0   0.0     0.0
   8:    0.0   0.0     0.0
   9:    0.0   0.0     0.0
Active UFRAME $MNUFRAMNUM[1]=0
[ TYPE ] DETAIL [OTHER ]  CLEAR  SETIND
```

14 To make the set user frame effective, press F5 (SETIND), then enter the frame number.

```
[OTHER ]  CLEAR  SETIND
```

F5

CAUTION
If you do not press F5, SETIND, the user frame will not be effective.

NOTE To select the number of a coordinate system to be used, the jog menu can also be used. See Section 5.2.3 "Moving the robot jog feed."

CAUTION
After all coordinate systems are set, the setting information should be saved in external storage (floppy disk, for example) in case the information needs to be re–loaded. Otherwise, the current setting information would be lost when it is changed.

15 To delete the data of the set frame, move the cursor to the desired frame and press F4,CLEAR.

```
[OTHER ]  CLEAR  SETIND
```

F4

Procedure 3–14 Setting User Frame Using Four Point Method

Step

1 Display the user frame list screen (Refer to the three point method)

```
SETUP Frames                        JOINT  30 %
 User Frame Setup/ Three Point     2/9
       X       Y       Z    Comment
  1: 1243.6    0.0    43.8 REFERENCE FR>
  2:     0.0    0.0    0.0 *************
  3:     0.0    0.0    0.0 *************
  4:     0.0    0.0    0.0 *************
  5:     0.0    0.0    0.0 *************
  6:     0.0    0.0    0.0 *************
  7:     0.0    0.0    0.0 *************
  8:     0.0    0.0    0.0 *************
  9:     0.0    0.0    0.0 *************
Active UFRAME $MNUFRAMNUM[1]=0
[ TYPE ] DETAIL [OTHER ]  CLEAR  SETIND
```

2 Move the cursor to the user frame number line you want to set.

3 Press F2,DETAIL. The user frame setup screen of the selected frame number is displayed.

```
[ TYPE ] DETAIL [OTHER ]
```

F2

4 Press F2,METHOD

5 Select Four Point. The user frame setup / four point screen is displayed.

```
1 Three Point
2 Four Point
3 Direct Entry

[ TYPE ] METHOD FRAME
```

F2

User frame setup screen (Four Point Method)

```
SETUP Frames                        JOINT  30%
 User Frame Setup/ Four Point       1/5
 Frame Number: 2
   X:     0.0     Y:     0.0     Z:     0.0
   W:     0.0     P:     0.0     R:     0.0
  Comment:********************
   Orient Origin Point:  UNINIT
   X Direction Point:    UNINIT
   Y Direction Point:    UNINIT
   System Origin:        UNINIT
Active UFRAME $MNUFRAMNUM[1]=0
[ TYPE ][METHOD] FRAME
```

6 Add a comment and teach the reference point. For details, refer to TCP auto set (Three Point Method).

```
SETUP Frames                        JOINT  30%
 User Frame Setup/ Four Point       5/5
 Frame Number: 2
   X: 1243.6     Y:   525.2     Z:   43.9
   W:   0.123    P:    2.34     R:    3.2
  Comment:            RIGHT FRME
   Orient Origin Point:  USED
   X Direction Point:    USED
   Y Direction Point:    USED
   System Origin:        USED
Active UFRAME $MNUFRAMNUM[1]=0
[ TYPE ][METHOD] FRAME   MOVE_TO RECORD
```

7 Press the PREV key. The user frame list screen is displayed. You can see all the user frame settings.

PREV

```
SETUP Frames                      JOINT  30 %
 User Frame Setup/ Four Point      2/5
       X      Y      Z    Comment
  1: 1243.6   0.0   43.8 REFERENCE FR>
  2: 1243.6 525.2   43.8   RIGHT FRAME
  3:    0.0   0.0    0.0 *************
  4:    0.0   0.0    0.0 *************
  5:    0.0   0.0    0.0 *************
  6:    0.0   0.0    0.0
  7:    0.0   0.0    0.0
  8:    0.0   0.0    0.0
  9:    0.0   0.0    0.0
Active UFRAME $MNUFRAMNUM[1]=0
[ TYPE ] DETAIL [OTHER ]  CLEAR  SETIND
```

8 To make the set user frame effective, press F5 (SETIND), then enter the frame number.

```
[OTHER ]  CLEAR  SETIND
```

F5

CAUTION

If you do not press F5, SETIND, the user frame will not be effective.

NOTE To select the number of a coordinate system to be used, the jog menu can also be used. See Section 5.2.3 "Moving the robot jog feed."

CAUTION

After all coordinate systems are set, the setting information should be saved in external storage (floppy disk, for example) in case the information needs to be re–loaded. Otherwise, the current setting information would be lost when it is changed.

9 To delete the data of the set frame, move the cursor to the desired frame and press F4,CLEAR.

```
[OTHER ]  CLEAR  SETIND
```

F4

Procedure 3–15 Setting User Frame Using Direct List Method

Step

1 Display the user frame list screen (Refer to the three point method).

```
SETUP Frames                         JOINT  30 %
  User Frame Setup/ Four Point        3/9
        X       Y       Z     Comment
   1: 1243.6    0.0    43.8 REFERENCE FR>
   2: 1243.6 525.2    43.8   RIGHT FRAME
   3:     0.0   0.0     0.0 *************
   4:     0.0   0.0     0.0 *************
   5:     0.0   0.0     0.0 *************
   6:     0.0   0.0     0.0 *************
   7:     0.0   0.0     0.0 *************
   8:     0.0   0.0     0.0 *************
   9:     0.0   0.0     0.0 *************
Active UFRAME $MNUFRAMNUM[1]=1
[ TYPE ] DETAIL [OTHER ]  CLEAR  SETIND
```

2 Move the cursor to the user frame number line you want to set.

3 Press F2,DETAIL or press the ENTER key. The user frame setup screen of the selected frame number is displayed.

4 Press F2,METHOD.

5 Select Direct List. The user frame setup / direct list is displayed.

User frame setup screen (Direct List Method)

```
[ TYPE ] DETAIL [OTHER ]
```

F2

```
1 Three Point
2 Four Point
3 Direct Entry
[ TYPE ] METHOD FRAME
```

F2

```
SETUP Frames                        JOINT  30 %
  User Frame Setup/ Direct Entry     1/7
  Frame Number: 3
   1  Comment:   ********************
   2      X:                      0.000
   3      Y:                      0.000
   4      Z:                      0.000
   5      W:                      0.000
   6      P:                      0.000
   7      R:                      0.000
          Configuration:  N D B, , 0
Active UFRAME $MNUFRAMNUM[1]=0
[ TYPE ][METHOD] FRAME  MOVE_TO RECORD
```

6 Add a comment and enter the coordinate values. For details, refer to tool frame (Direct List Method).

```
SETUP Frames                        JOINT  30 %
  User Frame Setup/ Direct Entry     4/7
  Frame Number: 3
   1  Comment:          LEFT FRAME
   2      X:                  1243.6
   3      Y:                  -525.2
   4      Z:                    43.9
   5      W:                     0.123
   6      P:                     2.34
   7      R:                     3.2
          Configuration:  N D B, , 0
Active UFRAME $MNUFRAMNUM[1]=0
[ TYPE ][METHOD] FRAME  MOVE_TO RECORD
```

7 To display the user frame list screen, press the PREV key. You can see the settings of all the user frame.

PREV

```
SETUP Frames                    JOINT  30 %
 User Frame Setup/ Three Point     3/5
       X      Y      Z    Comment
  1: 1243.6   0.0   43.8 REFERENCE FR>
  2: 1243.6 525.2   43.8   RIGHT FRAME
  3: 1243.6 -525.2   43.8    LEFT FRAME
  4:     0.0   0.0    0.0
  5:     0.0   0.0    0.0
  6:     0.0   0.0    0.0
  7:     0.0   0.0    0.0
  8:     0.0   0.0    0.0
  9:     0.0   0.0    0.0
Active UFRAME $MNUFRAMNUM[1]=0
[ TYPE ] DETAIL [OTHER ]  CLEAR  SETIND
```

8 To get the set user frame as effective, press F5,SETIND.

```
[OTHER ]  CLEAR  SETIND
```

F5

> **CAUTION**
>
> If you do not press F5, SETIND, the user frame will not be effective.

NOTE To select the number of a coordinate system to be used, the jog menu can also be used. See Section 5.2.3 ”Moving the robot by jog feed.”

> **CAUTION**
>
> After all coordinate systems are set, the setting information should be saved in external storage (floppy disk, for example) in case the information needs to be re–loaded. Otherwise, the current setting information would be lost when it is changed.

9 To delete the data of the set frame, move the cursor to the desired frame and press F4,CLEAR.

```
[OTHER ]  CLEAR  SETIND
```

F4

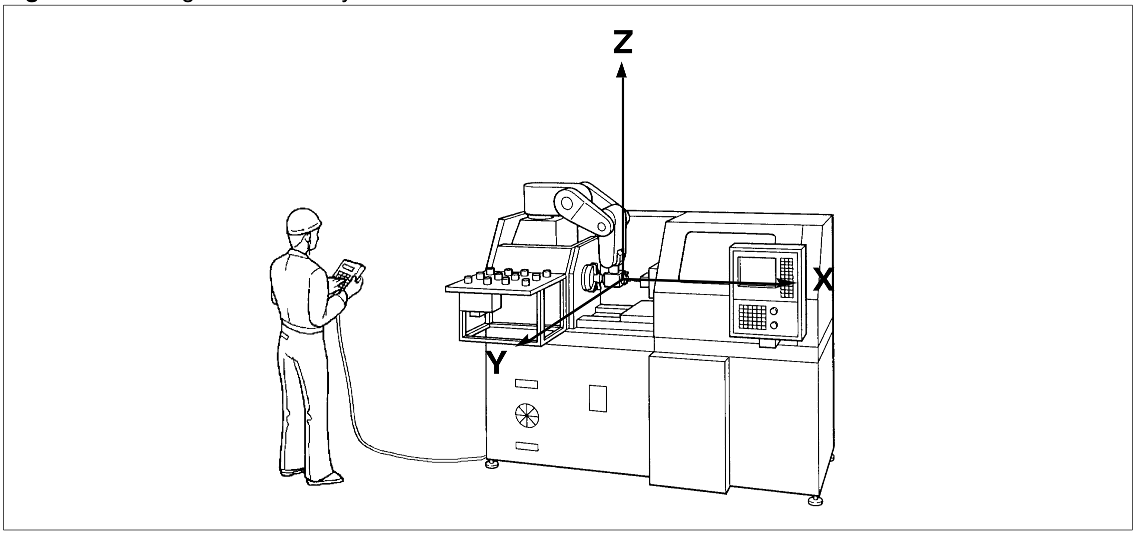
3.8.3 Setting a jog coordinate system

A jog coordinate system is a Cartesian coordinate system defined in a work space by the user. It is used to efficiently move the robot by Cartesian jog in the work space. (See Section 5.2.3.)

The jog coordinate system is defined by (x, y, z) indicating the position of the zero point, and (w, p, r) indicating the angular displacement around the X–, Y–, and Z–axes on the world coordinate system.

NOTE You need not take care of the jog frame origin,since it is used only when the jog frame is selected as the manual–feed coordinate system. The zero point of the jog coordinate system has no special meaning. Select any convenient position for defining the jog coordinate system.

Figure 3–32. Jog Coordinate System



The following system variables are changed by setting the jog frame with the frame setup screen.

- $JOG_GROUP [1] . $JOGFRAME is set the jog frame you want to used.

Five jog frames can be set and they can be switched according to the situation. It is substituted by the world frame when undefined.

Jog frame can be set by two methods.

Three Point Method

Three reference points need be taught. They are the start point of the x–axis,the positive direction of the x–axis,and one point on the x–y plane. The start point of the x–axis is used as the origin of the frame. Refer to Figure 3–29.

Direct List Method

The origin position x,y and z of the jog frame in the world frame and the rotating angle w,p, and r around the x–,y–,and z–axis of the world frame can be input directly. Refer to Figure 3–31.

Procedure 3–16 Setting Up Jog Frame Using Three Point Method

Step

1 Press the MENUS key. The screen menu is displayed.

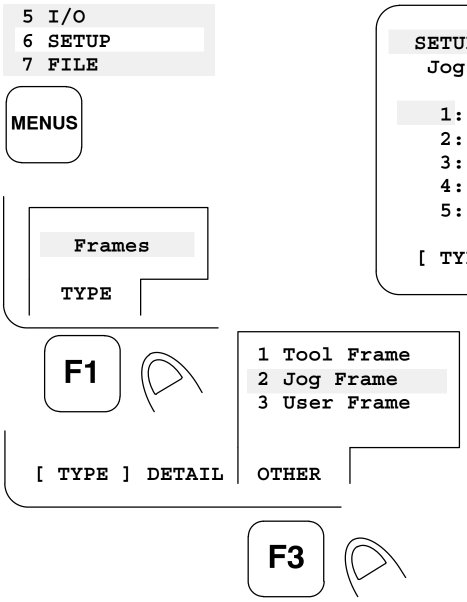
2 Select 6 (SETUP).

3 Press the F1 key, [TYPE]. The screen change menu is displayed.

4 Select Frames.

5 Press F3, OTHER

6 Select Jog Frame. Jog frame entry screen is displayed.

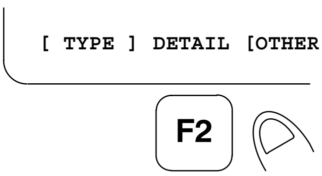


Jog frame list screen

```
SETUP Frames                    JOINT  30 %
  Jog Frame Setup / Three Point      1/5
        X       Y        Z     Comment
    1:     0.0    0.0     0.0 *************
    2:     0.0    0.0     0.0 *************
    3:     0.0    0.0     0.0 *************
    4:     0.0    0.0     0.0 *************
    5:     0.0    0.0     0.0 *************

[ TYPE ] DETAIL [OTHER ]  CLEAR  JGFRM
```

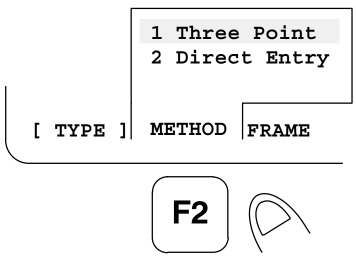
7 Move the cursor to the line of the jog frame number you want to set.

8 Press F2,DETAIL. The jog frame setup screen of the selected frame number is displayed.



9 Press F2,METHOD.

10 Select Three Point.



Jog frame setup screen (Three Point Method)

```
SETUP Frames                    JOINT  30 %
  Jog Frame Setup / Three Point      1/4
  Frame Number: 1
    X:     0.0     Y:     0.0    Z:     0.0
    W:     0.0     P:     0.0    R:     0.0
   Comment:********************
    Orient Origin Point:   UNINIT
    X Direction Point:     UNINIT
    Y Direction Point:     UNINIT

[ TYPE ][METHOD] FRAME
```

11 Add a comment and teach the reference point. For details, refer to TCP auto set(Three Point Method).

```
 SETUP Frames                      JOINT  30 %
  Jog Frame Setup / Three Point     4/4
  Frame Number: 1
    X: 1243.6    Y:     0.0   Z:    10.0
    W:     0.1   P:     2.3   R:     3.2
   Comment:          WORK AREA 1
    Orient Origin Point:  RECORDED
    X Direction Point:    RECORDED
    Y Direction Point:    UNINIT

 [ TYPE ][METHOD] FRAME   MOVE_TO RECORD
```

12 Press the PREV key. The jog frame list screen is displayed. You can see all the jog frame settings.

13 To display the user frame list screen, press the PREV key. You can see the settings for all user frames.

PREV

```
 SETUP Frames                      JOINT  30 %
  Jog Frame Setup / Three Point     1/5
          X      Y      Z    Comment
   1: 1243.6 525.2   60.0   WORK AREA 1
   2:    0.0   0.0    0.0 *************
   3:    0.0   0.0    0.0 *************
   4:    0.0   0.0    0.0 *************
   5:    0.0   0.0    0.0 *************

 [ TYPE ] DETAIL [OTHER ]  CLEAR  JGFRM
```

14 To make the set jog frame effective, press F5 (JGFRM), then enter the frame number.

```
[OTHER ]  CLEAR  JGFRM
```

F5

CAUTION

If you do not press F5, JGFRM the jog frame will not be effective.

NOTE To select the number of a coordinate system to be used, the jog menu can also be used. See Section 5.2.3 "Moving the robot jog feed."

CAUTION

After all coordinate systems are set, the setting information should be saved in external storage (floppy disk, for example) in case the information needs to be re–loaded. Otherwise, the current setting information would be lost when it is changed.

15 To delete the data of the set frame, move the cursor to the desired frame and press F4,CLEAR.

```
[OTHER ]  CLEAR  JGFRM
```

F4

Procedure 3–17 Setting Up Jog Frame Using the Direct List Method

Step

1 Display the jog frame list screen(Refer to the three point method).

```
 SETUP Frames                     JOINT  30 %
  Jog Frame Setup / Three Point      2/5
        X      Y      Z     Comment
   1: 1243.6 525.2   60.0   WORK AREA 1
   2:     0.0   0.0    0.0 *************
   3:     0.0   0.0    0.0 *************
   4:     0.0   0.0    0.0 *************
   5:     0.0   0.0    0.0 *************

 [ TYPE ] DETAIL [OTHER ]  CLEAR  JGFRM
```

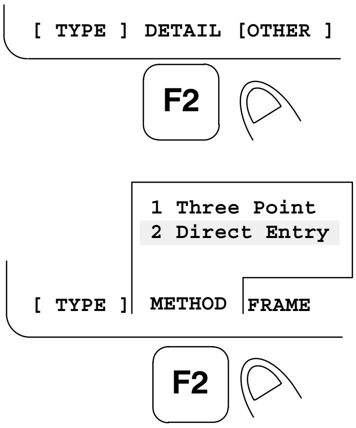
2 Move the cursor to the jog frame number line you want to set.

3 Press F2,DETAIL or press the ENTER key. The jog frame setup screen of the selected frame number is displayed.

4 Press F2,METHOD.

5 Select Direct Entry.

Jog Frame Setup Screen (Direct Entry Method)




```
 SETUP Frames                     JOINT  30 %
  Jog Frame Setup / Direct Entry     1/8
  Frame Number: 2
   1  Comment:    ********************
   2      X:                       0.000
   3      Y:                       0.000
   4      Z:                       0.000
   5      W:                       0.000
   6      P:                       0.000
   7      R:                       0.000
   8      Configuration:  N D B, , 0

 [ TYPE ][METHOD] FRAME   MOVE_TO   RECORD
```

6 Add a comment and teach the reference point. For details, refer to TCP auto set(Three Point Method).

```
 SETUP Frames                     JOINT  30 %
  Jog Frame Setup / Direct Entry     4/8
  Frame Number: 2
   1  Comment:            WORK AREA 2
   2      X:                    1003.000
   3      Y:                    -236.000
   4      Z:                      90.000
   5      W:                       0.000
   6      P:                       0.000
   7      R:                       0.000
   8      Configuration:  N D B, , 0

 [ TYPE ][METHOD] FRAME   MOVE_TO   RECORD
```

7 Press the PREV key. The jog frame list screen is displayed. You can see all the jog frame settings.

PREV

```
 SETUP Frames                        JOINT  30 %
  Jog Frame Setup / Three Point      2/5
        X       Y      Z    Comment
   1: 1243.6 525.2   60.0   WORK AREA 1
   2: 1003.0-236.0   90.0   WORK AREA 2
   3:     0.0   0.0    0.0 *************
   4:     0.0   0.0    0.0 *************
   5:     0.0   0.0    0.0 *************

 [ TYPE ] DETAIL [OTHER ]  CLEAR  JGFRM
```

8 To make the set jog frame effective, press F5 (JGFRM), then enter the frame number.

```
[OTHER ]  CLEAR  JGFRM
```

F5

CAUTION
If you do not press F5,JGFRM,the set frame will not be effective.

NOTE To select the number of a coordinate system to be used, the jog menu can also be used. See Section 5.2.3 "Moving the robot by jog feed."

CAUTION
After all coordinate systems are set, the setting information should be saved in external storage (floppy disk, for example) in case the information needs to be re–loaded. Otherwise, the current setting information would be lost when it is changed.

9 To delete the data of the set frame, move the cursor to the desired frame and press F4,CLEAR.

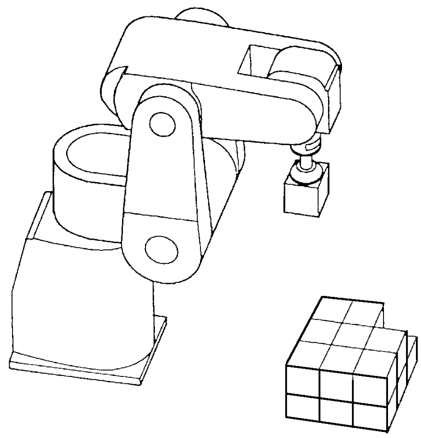
```
[OTHER ]  CLEAR  JGFRM
```

F4

3.9 Setting a Reference Position

A reference position is a fixed (predetermined) position that is frequently used in a program or when the robot is moved by jog feed. The reference position is a safe position, which is usually distant from the operating area of the machine tool or peripheral equipment. Three reference positions can be specified.

Figure 3–33. Reference Position



When the robot is at the reference position, a predetermined digital signals, SDO, is output. If the reference position is invalidated, the DO signal is not output.

When the robot is at reference position 1, the reference position output signal (ATPERCH) of the peripheral device I/O is output.

For this function, the reference position settings can be disabled so that the signal is not output.

To make the robot move to the reference position, make the program which specifies the return path and execute this program. At this time, also specify the oreder in which axes returns to the reference position in the program. Moreover, it is convenient to set the return program as a macro instruction.(See Section 9.1, ”Macro instruction”)

Specify the reference position on the reference position setting screen [6 (SETUP). Ref Position].

Procedure 3–18 Setting a reference position

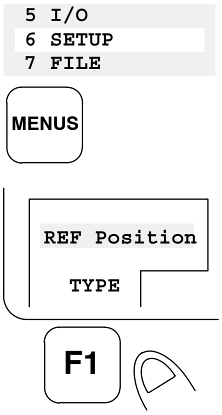
Step

1 Press the MENUS key.

2 Select SETUP.

3 Press the F1 key, TYPE.

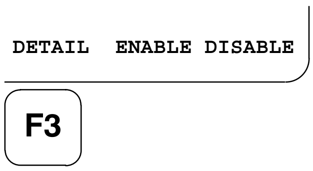
4 Select “Ref Position.” The reference position selection screen is displayed.



Reference Position Selection Screen

```
 REF POSN                                JOINT 30%
                                              1/3
   NO    End/Dsbl    @Pos    Comment
    1    DISABLE    FALSE   [                ]
    2    DISABLE    FALSE   [                ]
    3    DISABLE    FALSE   [                ]


 [TYPE]             DETAIL    ENABLE    DISABLE
```

5 Press the F3 key, DETAIL. The detailed reference position screen is displayed.



Detailed Reference Position Screen

```
 REF POSN                               JOINT 30%
  Reference Position                        1/12
  Ref.Position Number:  1
   1  Comment:           [**********]
   2  Enable/Disable:   DISABLE
   3  Sinal definition:SDO[ 0]
   4  J1        0.0      +/-       0.0
   5  J2        0.0      +/-       0.0
   6  J3        0.0      +/-       0.0
   7  J4        0.0      +/-       0.0
   8  J5        0.0      +/-       0.0
   9  J6        0.0      +/-       0.0
  [TYPE]                         RECORD
```

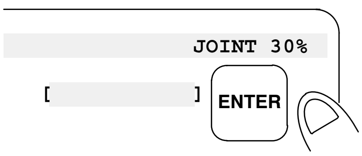
6 To enter a comment, follow these steps:

a Place the cursor on the comment line and press the ENTER key.

b Determine whether the comment is entered by words, alphabetic characters, or katakana.

c Press the corresponding function key and enter the desired comment.

d After entering the comment, press the ENTER key.

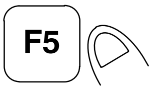



```
 REF POSN                                JOINT 30%
 Reference Position                          1/12
   1  Comment:             [Refpos1***]

 [TYPE]
```

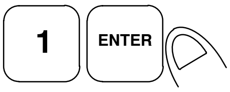
7 In the “Signal definition” line, specify the digital output signal to be output when the tool is at the reference position.

```
                 JOINT 30%
                   3/12
   DO[ 0]

             DO      RO
```

```
 REF POSN                          JOINT 30%

   3  Signal definition: RO[ 0]

[TYPE]                      DO       RO
```



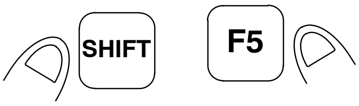

```
                 JOINT 30%
                   3/12
   RO[ 0]
```




```
 REF POSN                          JOINT 30%

   3  Signal definition:  RO[ 1]

[TYPE]                             RECORD
```

8 To teach the reference position, place the cursor on the setting fields J1 to J9. While pressing the SHIFT key, press the F5 key, RECORD. The current position is recorded as the reference position.

```
              RECORD
```




```
 REF POSN                          JOINT 30%

   4  J1        0.0      +/-       0.0

[TYPE]                             RECORD
```

9 To enter the numeric value of the reference position directly, place the cursor on the setting fields J1 to J9 and enter the coordinates of the reference position. Enter the coordinates in the left column and allowable errors in the right column. Moreover, the value entered to the setting field which specifies an unused axis is ignored.

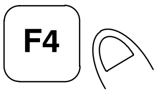
```
 REF POSN                          JOINT 30%
 Reference Position                    1/12
  Ref.Position Number:  1
   1  Comment:          [Refpos1    ]
   2  Enable/Disable:   ENABLE
   3  Sinal definition:  RO[ 1]
   4  J1      129.000   +/-       2.000
   5  J2      -31.560   +/-       2.000
   6  J3        3.320   +/-       2.000
   7  J4      179.240   +/-       2.000
   8  J5        1.620   +/-       2.000
   9  J6       00.000   +/-       0.000
[ TYPE ]                        RECORD
```

10 After the reference position is specified, press the PREV key. The reference position selection screen is displayed again.


```
 REF POSN                          JOINT 30%
                                       1/3
  NO    Enb/Dsbl   @Pos    Comment
   1    DISABLE    FALSE   [Refpos1        ]
   2    DISABLE    FALSE   [               ]
   3    DISABLE    FALSE   [               ]


[TYPE]             DETAIL    ENABLE    DISABLE
```

11 To enable or disable the reference position output signal, place the cursor on the ENABLE/DISABLE field and press the corresponding function key.

```
 REF POSN

   NO  Enb/Dsbl  @Pos
    1   DISABLE   FALSE

    ENABLE     DISABLE
```




```
 REF POSN                         JOINT 30%
                                       1/3
   NO   Enb/Dsbl   @Pos   Comment
    1   ENABLE    FALSE  [Refpos1         ]

[TYPE]            DETAIL   ENABLE   DISABLE
```

3.10 Joint Operating Area

The software restricts the operating area of the robot according to a specified joint operating area. The standard operating area of the robot can be changed by specifying the joint operating area.

Specify the joint operating area at [6 SYSTEM Axis Limits] on the joint operating area setting screen.

WARNING

The robot operating area should not be controlled only by the joint moving range function. Limit switches and mechanical stoppers should be used together with the function. Otherwise, injury or property damage would occur.

WARNING

The mechanical stoppers should be adjusted to the software settings. Otherwise, injury or property damage would occur.

CAUTION

Changing the joint moving range will affect the robot operating area. Before the joint moving range is changed, the expected effect of the change should be carefully studied in order to prevent possible trouble. Otherwise, the change would produce unpredictable results. For example, an alarm might occur at a position programmed earlier.

UPPER

Specifies the upper limit of the joint operating area, which is the limit of the motion in the positive direction.

LOWER

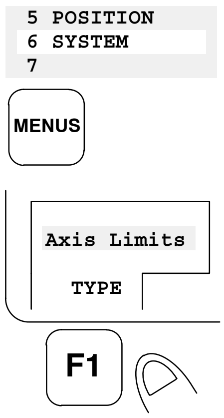
Specifies the lower limit of the joint operating area, which is the limit of the motion in the negative direction.

Enabling the new setting

After a new joint operating area is specified, turn the controller off and on again to enable the new setting.

Procedure 3–19 Setting the joint operating area

Step
1 Press the MENUS key. The screen menu is displayed.
2 Select 6 (SYSTEM).
3 Press F1 (TYPE). The screen change menu is displayed.
4 Select Axis Limits. The joint operating area setting screen is displayed.



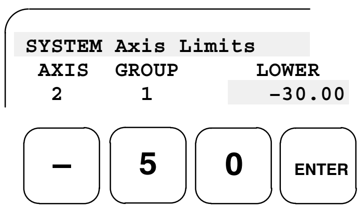
Joint Operating Area Setting Screen

```
SYSTEM Axis Limits            JOINT   30 %
 AXIS  GROUP       LOWER      UPPER  1/16

   1      1       -160.00     160.00 dg
   2      1        -30.00     150.00 dg
   3      1       -156.50     206.10 dg
   4      1       -120.00     120.00 dg
   5      1          0.00       0.00 mm
   6      0          0.00       0.00 mm
   7      0          0.00       0.00 mm
   8      0          0.00       0.00 mm
   9      0          0.00       0.00 mm

[ TYPE ]
```

> **WARNING**
> **The robot operating area should not be controlled only by the joint moving range function. Limit switches and mechanical stoppers should be used together with the function. Otherwise, injury or property damage would occur.**

NOTE Value 0.000 means that the robot does not have the corresponding axis.

5 Place the cursor on the target axis limits field, and enter a new value from the teach pendant.




```
SYSTEM Axis Limits            JOINT   30 %
 AXIS  GROUP       LOWER      UPPER  2/16
  2      1         -50.00     100.00 dg

[ TYPE ]
```

6 Repeat the above step for all the axes.
7 To make the set information effective, turn the controller off and on again in cold start mode (Section 5.2.1).

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

3.11 User Alarm

In the user alarm setup screen, the message displayed when the user alarm is generated is set.

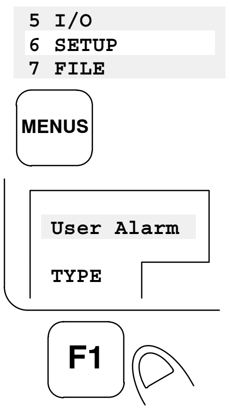
The user alarm is the alarm which is generated when the user alarm instruction is executed. (See Section 4.14.2 ”User alarm instruction”)

Settings for user alarm is done in the user alarm setup screen[6 SETUP.User Alarm].

Procedure 3–20 Setting Up the User Alarm

Step

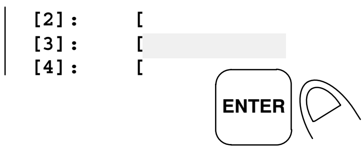
1 Select the MENUS key. The screen menu is displayed.
2 Select 6(SETUP).
3 Press the F1 key, TYPE. The screen change menu is displayed.
4 Select User Alarm. The user alarm setup screen is displayed.



User Alarm Setup Screen

```
 Setting/User Alarm           JOINT  30 %
                                    1/200
 Alarm No.           User Message
   [1]:    [                            ]
   [2]:    [                            ]
   [3]:    [                            ]
   [4]:    [                            ]
   [5]:    [                            ]
   [6]:    [                            ]
   [7]:    [                            ]
   [8]:    [                            ]
   [9]:    [                            ]
 [ TYPE ]
```

5 Move the cursor to the line of the user alarm number you want to set and press the ENTER KEY. Enter the message with the function keys.




```
 Setting/User Alarm           JOINT  30 %
  1 Upper Case
  2 Lower Case
  3 Punctuation
  4 Options
 Setting/User Alarm
 Alarm No.           User Message
   [1]:    [                            ]
   [2]:    [                            ]
   [3]:    [WORK                        ]
 Old Value:
  ABCDEF  GHIJKL  MNOPQR  STUVWX  YZ_@*
```

6 When you are finished to input the message of the user alarm,press the ENTER key. The user alarm message has been set.

```
 Setting/User Alarm           JOINT  30 %
                                    3/200
 Alarm No.           User Message
   [1]:    [                            ]
   [2]:    [                            ]
   [3]:    [NO WORK                     ]
   [4]:    [                            ]
   [5]:    [                            ]
   [6]:    [                            ]
   [7]:    [                            ]
   [8]:    [                            ]
   [9]:    [                            ]
 [ TYPE ]
```

3.12 Variable Axis Areas

On the variable axis area setting screen, multiple (up to three) sets of stroke limits can be set for the J1 axis and an additional axis.

The variable axis area function allows the user to switch from one set of stroke limits to another during program execution.

* This function is offered by the specific robot only.

Upper limit

Indicates the upper limit for a joint operating area. Operating area in the plus direction.

Lower limit

Indicates the lower limit for a joint operating area. Operating area in the minus direction.

After changing an upper or lower limit, turn off the power to the control unit and then turn it on with a cold start. With a cold start, the new upper or lower limit takes effect and the selected joint operating area is returned to the standard value ($PARAM_GROUP.$SLMT_**_NUM).

> **CAUTION**
>
> Changing a joint operating area affects the operating area of the robot. To avoid problems, it is necessary to thoroughly consider the effect of a change in the joint operating area before making the change.

Procedure 3–21 Setting a variable axis area

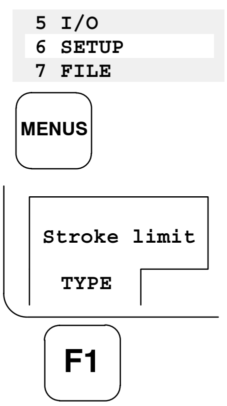
Step

1 Press MENUS. The screen menu appears.

2 Select SETUP.

3 Press F1 ”TYPE.” The screen switching menu appears.

4 Select Stroke limit. The variable axis area setting screen appears.



Variable Axis Area Setting Screen

```
 Stroke limit setup              JOINT 30%
      GROUP:1             AXIS:J1
No.   Lower>-180.0      UPPER<180.0
 1:     0.0 deg           0.0 deg
 2:     0.0 deg           0.0 deg
 3:     0.0 deg           0.0 deg
Default
 0: -180.0 deg        180.0 deg

Active limit:
& MRR_GRP[1]. $SLMT_J1_NUM=0

[TYPE]    GROOP#    AXIS#
```

5 Position the cursor to the desired axis area. Enter new values using the numeric keys on the teach pendant.

- The upper and lower limits must be within the stroke limits of the system. (–>Section 3.8, ”Joint Operating Area”). If an attempt is made to set a value outside the limits, the upper or lower limit is fixed to the system default value.
- To switch from one motion group to another, use the F2 key (group #).
- To set an additional axis, press the F3 key (axis #) to switch to the additional axis setting screen.

6 To make the settings effective, turn off the power and then back on. When the power is turned on for the first time after the settings have been changed, a cold start is automatically performed.

Procedure 3–22 Using a variable axis area

Condition ■ A proper axis area has been set and is effective.

Step 1 To switch to the joint operating area that has been set on the variable axis area setting screen during program execution, use the parameter instruction (→Section 4.14.7, ”Parameter instruction”). For example, after the following program has been executed

```
PRG1                                    JOINT 30%
  1:    $MRR_GRP[1]. $SLMT_J1_NUM=1
  2:    $PARAM_GROUP[1]. $SLMT_J1_NUM=1




[INST]                                  [EDCMD]
```

Value No.1 is used for the joint operating area for the J1 axis. To switch to another joint operating area for the additional axis, use the following command:

```
PRG1                                    JOINT 30%
  3:    $MRR_GRP[1]. $SLMT_J1_NUM=2
  4:    $PARAM_GROUP[1]. $SLMT_J1_NUM=2




[INST]                                  [EDCMD]
```

3.13 Preventive Interference Area Function

The preventive interference area function is a function that automatically stops the robot when a move instruction that causes the robot to enter the preset interference area is issued, the robot will stop if another robot or peripheral device is located in that interference area and, after confirming that the other robot or peripheral device has moved out of the interference area, automatically releases the robot from the stopped state to restart its operation.

Communication between a robot and a peripheral device is accomplished with a set of interlock signals (one signal for each of input and output). One set of interlock signals is allocated to one interference area. Up to three interference areas can be defined.

The relationship between the interlock signals and the robot is as described below.

Output signal

The output signal is off when the tool endpoint is located inside the interface area. It is on when it is located outside the area.

State	Output signal
Safe (tool endpoint located outside the interference area)	On
Dangerous (tool endpoint located inside the interference area)	Off

Input signal

When the input signal is off, and the robot attempts to enter the interference area, the robot enters the hold state. When the input signal is turned on, the robot is released from the hold state, automatically restarting its operation.

CAUTION

The robot decelerates to stop at the point where the tool endpoint enters the interference area, so that the robot actually stops at a position inside the interface area. The faster the operating speed of the robot, the deeper the robot enters the interference area. Consider this and other factors, such as the tool size, to ensure that a sufficiently large interface area is set.

To set up the preventive interference area function, use the SETUP Space function.

To set up the following items, use the Rectangular Space/DETAILED SCREEN screen.

Table 3–5. Items of the preventive interference area function (area details screen)

Item	Description
Enable/disable	Enables and disables this function. To change the settings of the other items, this function must be disabled for the area for which the settings of the items are to be changed.
Comment	Allows the user to enter a comment of up to 10 characters.
Output signal	Sets up the output signal.
Input signal	Sets up the input signal.
Driority	When two robots use this function, this item specifies which robot is to enter the interference area first if the two robots attempt to enter the interference area at the same time. The robot for which High is set enters the interface area first. When the robot completes its operation and moves out of the interference area, the robot for which Low is set enters the interference area. The setting for one robot must be different from that for the other. **NOTE** If High or Low is set for both robots, and the robots attempt to enter the interference area at the same time, they both enter the stopped (deadlock) state. If this occurs, perform the recovery operation described below and check that the settings are correct. 1 Perform an emergency stop on both robots. **WARNING** **If an emergency stop is not performed on both robots, one robot will automatically start its operation when the other moves out of the interference area. This is very dangerous.** 2 Check that there are no objects or by standers that a robot could hit. 3 Disable this function. 4 Move either robot out of the interference area, using a jog operation.
inside/outside	Specifies whether the inside or outside of a rectangular parallelepiped is to be an interference area.

To set up the following items, use the Rectangular Space/SPACE SETUP screen.

Table 3–6. Items of the preventive interference area function (area setting screen)

Item	Description
BASIS VERTEX	Position of the vertex of a rectangular parallel pipe that is to become the reference.
SIDE LENGTH/SECOND VERTEX	If SIDE LENGTH is selected, specify the lengths of the sides of a rectangular parallelepiped from the reference vertex along the X, Y, and Z axes in the user coordinate system. (The sides of the rectangular parallel pipe must be parallel to the respective axes of the user coordinate system.) If SECOND VERTEX is selected, the rectangular parallel pipe having the reference vertex and the diagonal vertex, specified here, becomes an interference area.

Procedure 3–23 Setting up the preventive interference area function

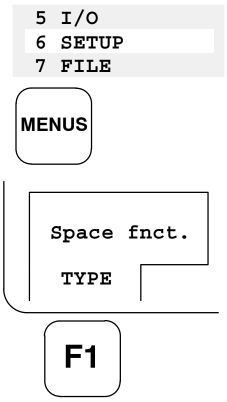
Step

1 Press MENUS. The screen menu appears.

2 Select SETUP.

3 Press F1 ”TYPE.” The screen switching menu appears.

4 Select Space fnct. The area list screen appears.



Area List Screen

```
Rectangular Space                JOINT 30%
LIST SCREEN
No. End/Dsbl     Comment         Usage
 1  ENABLE    [              ] Common Space
 2  DISABLE   [              ] Common Space
 3  DISABLE   [              ] Common Space



[TYPE]          DETAIL   ENABLE   DISABLE
```

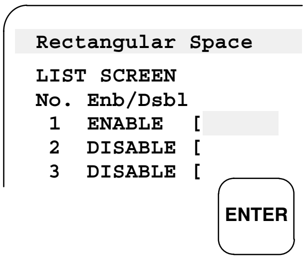
5 The area list screen allows the user to enable and disable each interface area with the appropriate function key. To enter a comment, use the procedure below:

a. Move the cursor to the desired comment line and press the Enter key.

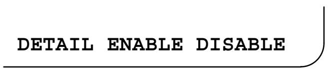
b. Specify which of alphabetic or katakana characters are to be used to enter a comment.

c. Press the appropriate function key to enter a comment.

d. When a comment is entered, press the Enter key.



6 To set up an item other than Enb/Dsbl or Comment, press F3 (DETAIL). The details screen appears.




```
Rectangular Space
DETAILED SCREEN
   SPACE:1              GROUP:1
   USAGE:            Common Space
  1 Enable/Disable:     ENABLE
  2 Comment:           [**********]
  3 Output Signal:      DO[0]
  4 Input Signal:       DI[0]
  5 Priority:           High
  6 Inside/Outside:     Inside

[TYPE]  SPACE            ENABLE   DISABLE
```

7 Position the cursor to the desired item. Change the setting of the item using the function or numeric keys.

8 To set an area, press SPACE. The area setting screen appears.

[TYPE] SPACE

```
 Rectangular Space                JOINT 30%
 SPACE SETUP                            1/4
    SPACE:1                  GROUP:1
   UFRAME:0                  UTOOL:1
 1:BASIS VERTEX             [SIDE LENGTH]
 2:X 0.0 mm                  0.0 mm
 3:Y 0.0 mm                  0.0 mm
 4:Z 0.0 mm                  0.0 mm



[TYPE]  OTHER                       RE10RD
```

9 The reference vertex and the side lengths or diagonal vertex can be set in either of two ways:

a. Position the cursor to the X, Y, and Z coordinate fields and enter the desired coordinates directly using the numeric keys.

b. Move the robot to a vertex of a rectangular parallelepiped, then read the current position of the robot with SHIFT key +F5 RECORD.

RE10RD

F5

SHIFT

NOTE If UF or UT is to be changed, perform operation b first. By means of this operation, the current UF or UT value is selected.

NOTE When the user coordinate system values are changed, the spatial position of the interference area does not change. When the user coordinate system values have been changed and an interference area is to be defined in the new user coordinate system, use SHIFT key +F5 RECORD to set an interference area again.

10 After setting the area, press PREV. The area details screen reappears. To return to the area list screen, press PREV again.

PREV

3.14 System Config Menu

The System Config Menu includes some important components which should be set when the system is established.

In the system config menu,the following items can be set.

Table 3–7. System config menu

ITEMS	DESCRIPTIONS
Use HOT START (Hot Start)	When the hot start is set to TRUE, hot start is done at turning on the controller. (Default setting = FALSE)
I/O power fail recovery	Specifies whether or how to perform I/O power failure recovery if the hot start function is enabled and how to perform simulated recovery if the hot start function is disabled. There are four power failure recovery modes, as described below. – NOT RECOVER I/O power failure recovery is not performed regardless of whether the hot start function is enabled. All outputs are turned off, and the simulated state is reset. – RECOVER SIM Simulated–state recovery is performed regardless of whether the hot start function is enabled, and the simulated state is reset, but all actual outputs and simulated inputs/outputs are turned off. – UNSIMULATE I/O power failure recovery is performed, but all the simulated states are reset. This is equivalent to NOT RECOVER if the hot start function is disabled, because the output states are not recovered. – RECOVER ALL I/O power failure recovery is performed if the hot start function is enabled. The output and simulated states are recovered to the states that existed immediately before the power is turned off. If the hot start function is disabled, RECOVER ALL is equivalent to RECOVER SIM, because the output states are not recovered. **CAUTION** Even if power failure handling is enabled, the output signal is turned off without being recovered in the following cases: ● When the I/O allocation is changed before the power is turned off. ● When the fuse of an I/O unit blows, or when an I/O unit is turned off. ● When the I/O unit configuration is changed.
Autoexec program for Cold start **Autoexec program for Hot start**	Specifies the name of the auto–start program for the hot start. The specified program is executed immediately after the power is turned on. If it does not end within 15 seconds, it will be aborted. **CAUTION** The program automatically executed at power of is executed just before the servo power is turned on. Therefore the robot can not be moved by this program. Set the program which initializes the condition of setup and I/O of the system. You should set the name of program which sets up the system,initializes I/O...etc. Moreover,the attributes should be set as follows on the program detail screen. Group Mask: [* , * , * , * , *] Ignore pause: [ON]
HOT START done signal	Specifies the digital signal (SDO) that is to be output at the hot start. If the hot start is not performed, the digital signal is turned off. This function is disabled if 0 is specified.
Restore selected program	Specifies whether the program selected at turning off the controller is selected after turning on the controller when the cold start is done. When this is set to TRUE,the program selected at the power off is selected after the power on again. When this is set to FALSE,the program is not selected after power on again. This is set to TRUE in standard setting.
Disable UI signals	Selects whether a UI signal is valid or invalid. When this is set to FALSE,the peripheral input signals (UI[1 to 8]) is disabled. See Section 3.3 ”Peripheral I/O”.
START for CONTINUE only	If this item is enabled, the external start signal (START) starts only those programs that have been paused. →See Peripheral I/O.
CSTOPI for ABORT	If this item is enabled, those programs that are currently running are forcibly terminated immediately upon the input of CSTOPI. →See Peripheral I/O.
Abort all programs by CSTOPI	Specifies whether all programs are to be forcibly terminated with the CSTOPI signal in a multitasking environment. If this item is set to TRUE, the CSTOPI input signal functions as follows: ● If RSR is selected for the RSR/PNS item, all programs are forcibly terminated. ● If PNS is selected for the RSR/PNS item, the selected program is forcibly terminated. If no program is selected, however, all programs are forcibly terminated.

Table 3–7. (Cont'd) System config menu

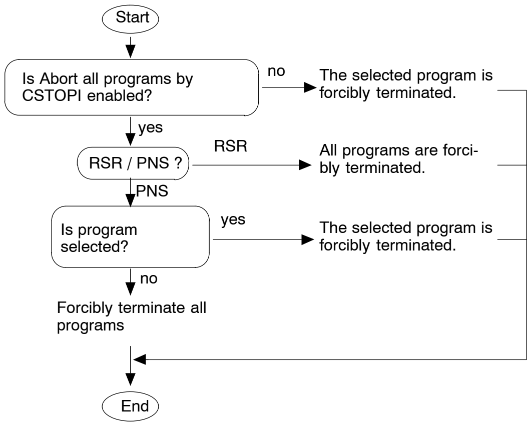
ITEMS	DESCRIPTIONS
	 If this item is set to FALSE, the CSTOPI input signal causes only the currently selected program to be forcibly terminated. (Default setting)
PROD–START depend on PHSTROBE	If this item is enabled, the PROD_START input is enabled only when the PNSTROBE input is on. By enabling this item, it is possible to prevent a program that should not be started from being started accidentally due to noise or a sequence error when that program is displayed on the teach pendant.
Detect FAULT_RESET signal	Specifies whether the reset signal is detected the instant it rises or falls. When this setting is changed, turn the controller off and on again to use the new information. At this time the cold start is done automatically. The falling edge is detected by standard setting.
Use PPABN signal	Specifies if the pneumatic pressure alarm(*PPABN) is detected for each motion group. Move the cursor to this line and press ENTER key. The setup screen for each motion group is displayed. When *PPABN signal is not used,set this invalid. When this setting is changed,turn off the controller,and turn on the controller to use the new information. At this time it atart when HOT START is disable.
WAIT timeout	Specifies the period of time used in the conditional wait instruction(WAIT ..., TIMEOUT LBL[...]). The period of time is 30 second.
RECEIVE timeout	For this item, set the limit time for register receive instruction RCV R[...] LBL[...] (can be specified only when the sensor interface option is specified).
Return to top of program	Specifies whether the cursor is moved to the top of the program or not when it is finished to be executed. When this setting is TRUE,the cursor stays at the last line without returning to the top of the program when the program is finished to be executed. The default setting is TRUE. It is effective in the standard.
Original program name (F1 to F5)	Specifies the words which is displayed as the soft key at registering a program. It is convenient to set the words used many times as the program name to this.
Default logical command	It is possible to enter the screen to which standard instruction function key is set by pushing the input key from the condition that there is a cursor in setting a standard instruction. – **Name** Specifies the name which is displayed as the function key title.(Up to 7 characters) – **Lines** Specifies the number of the logic command registered in on function key. The default logical command up to four can be registered in one function key. When the Lines is set to 0,the function of teaching the default logical command is invalid.
Maximum of ACC instruction	Specifies the maximum of the override value used in the acceleration override motion option(ACC ...). The default value is 150.
Minimum of ACC instruction	Specifies the minimum of the override value used in the acceleration override motion option(ACC ...).
WJNT for default motion	Adds the Wjnt motion option to all linear and circular default motion instructions or delete it from them. – Pressing the F4 (ADD) key adds the Wjnt motion option to all the linear and circular default motion instructions and changes the screen display from "DELETE" (or ******) to "ADD." – Pressing the F5 (DELETE) key deletes the Wjnt motion option from all the linear and circular default motion instructions and changes the screen display from "ADD" (or ******) to "DELETE."
Auto display of alarm menu	Toggles the function for automatically displaying the alarm screen between FALSE and TRUE. The default setting is FALSE. If the setting of this item is changed, the power must be turned off and then back on. ● FALSE: Does not display the alarm screen automatically. ● TRUE: Displays the alarm screen automatically.
Force Message	Specifies whether the user screen is to appear automatically when a message instruction is executed in a program.

Table 3–7. (Cont'd) System config menu

ITEMS	DESCRIPTIONS
Hand broken	Enables and disables hand breakage (*HBK) detection. When multiple robots are used, hand breakage detection can be enabled and disabled for two robots. Press the Enter key with the cursor positioned on this line. Then, the screen for enabling or disabling hand breakage detection for each robot appears. On this screen, move the cursor to ENABLE or DISABLE, then press the ENABLE (F4) or DISABLE (F5) key to enable or disable hand breakage detection. When hand breakage detection is enabled, and the *HBK signal is off, alarm "–SRVO–006 Hand broken" is issued. See Appendix D–2, "RECOVERY FROM THE HAND BREAKAGE ALARM," and release the alarm. When the *HBK signal is off, and this signal is not to be used, disable the hand breakage detection. When hand breakage detection is disabled although a hand is installed, and the *HBK signal is used, "SRVO 302 Set hand broken to ENABLE" is displayed if the *HBK signal is on. Enable hand breakage detection. If the *HBK signal is turned off when hand breakage detection is disabled, "SRVO 300 Hand broken / HBK disabled" is issued. In this case, this alarm can be released by pressing the reset key. By default, hand breakage detection is enabled.
Reset CHAIN FAILURE detection	Resets a chain abnormality alarm (servo 230 or 231) when it is issued. For details on the chain abnormality alarm and for how to make hardware checks, refer to the maintenance manual. <Reset procedure> 1) Check for any hardware problem. 2) Press the emergency stop button on the teach pendant. (Input an emergency stop signal other than the emergency stop signal currently generated.) 3) Turn the emergency stop button on the teach pendant to release the emergency stop condition. 4) Move the cursor to this line, then press the F4 (TURE) key. 5) Press the reset button on the teach pendant.
Remote / Local setup	Select the method for setting the remote signal (SI[2]) that switches between remote mode and local mode of the system. – Remote : Keeps SI[2] on (remote mode) at all times. – Local : Keeps SI[2] off (local mode) at all times. – External I/O : Reflects the external signal status on SI[2]. When selecting this item, specify an external signal for External I/O (ON : Remote) on the next line. – OP panel key: When the R–J3i MODEL B controller is used, this item cannot be selected.
External I/O (ON : Remote)	When External I/O (ON : Remote) is selected in Remote / Local setup above, specify an external signal to be used here. Choose from SDI, SDO, RDI, RDO, UI, and UO.
Allow force I/O in AUTO mode	Enables or disables signal setting from TP when AUTO mode is set. By default, setting is enabled. – Yes: Enables signal setting. – No: Disables signal setting.
Allow chg. ovrd. in AUTO mode	Enables or disables override change from TP when AUTO mode is set. By default, change is enabled. – Yes: Enables override change. – No: Disables override change.
Signal to set in AUTO mode	If the three–mode switch is set to AUTO mode, a specified SDO is turned on. When 0 (default) is set, this function is disabled. When the setting has been changed, the power must be turned off then back on.
Signal to set in T1 mode	If the three–mode switch is set to T1 mode, a specified SDO is turned on. When 0 (default) is set, this function is disabled. When the setting has been changed, the power must be turned off then back on.
Signal to set in T2 mode	When the three–mode switch is set to T2 mode, a specified SDO is turned on. When 0 (default) is set, this function is disabled. When the setting has been changed, the power must be turned off then back on.

Table 3–7. (Cont'd) System config menu

ITEMS	DESCRIPTIONS
Signal to set if E–STOP	When an emergency stop (TP external emergency stop, operator's panel) is applied, a specified SDO is output. When 0 (default) is set, this function is disabled. When the setting has been changed, the power must be turned off then back on.

Procedure 3–24 Setting The System

Step

1 Select the MENUS key. The screen menu is displayed.

2 Select 6(SYSTEM) in the next page.

3 Press the F1 key, [TYPE]. The screen change menu is displayed.

4 Select Config. The system configuration screen is displayed.

System Configuration Screen

```
 5 I/O
 6 SETUP
 7 FILE
```

MENUS

```
   Config

  TYPE
```

```
System/Config                      JOINT 30%
                                        1/37
  1 Use HOT START:                     FALSE
  2 I/O power fail recovery: RECOVER ALL
  3 Autoexec program               [********]
            for Cold start:
  4 Autoexec program               [********]
            for Hot start:
  5 HOT START done signal:            DO[0]
  6 Restore selected program:         TRUE
  7 Enable UI signals :               TRUE
  8 START for CONTINUE only :        FALSE
  9 CSTOPI for ABORT :               FALSE
 10 Abort all programs by CSTOPI : FALSE
 11 PROD_START depend on PNSTROBE :FALSE
 12 Detect FAULT_RESET signal :       FALL
 13 Use PPABN signal :            <*GROUPS*>
 14 WAIT timeout :                 30.00 sec
 15 RECEIVE timeout :              30.000 sec
 16 Return to top of program :        TRUE
 17 Original program name (F1) : [PRG     ]
 18 Original program name (F2) : [MAIN    ]
 19 Original program name (F3) : [SUB     ]
 20 Original program name (F4) : [TEST    ]
 21 Original program name (F5) : [*******]
 22 Default logical command :  <*DETAIL*>
 23 Muximum of ACC instruction :   150
 24 Minimum of ACC instruction :      0
 25 WJNT for default motion :      ******
 26 Auto display of alarm menu :   FALSE
 27 Force Message :                ENABLE
 28 Reset CHAIN FAILURE detection : FALSE
 29 Allow Force I/O in AUTO mode  : TRUE
 30 Allow chg. ovrd. in AUTO mode : TRUE
 31 Signal to set in AUTO mode DOUT [  0]
 32 Signal to set in T1 mode DOUT   [  0]
 33 Signal to set in T2 mode DOUT   [  0]
 34 Signal to set if E-STOP DOUT    [  0]
 35 Hand broken :                 <*GROUPS*>
 36 Remote / Local setup :          Remote
 37 External I/O (ON : Remote) : DI [  0]

[TYPE]                  [CHOICE]
```

5 Move the cursor to the field you want to set and enter the new value by using the numerical key or using the function key on the teach pendant. As for the field which should be set character string, move the cursor to it and press the ENTER key. Then the character input becomes possible.

NOTE As for the setting of ”Use PPABN signal:”, ”Hand Broken:” or ”Default logical command:”,move the cursor to ”<*GROUPS*> or ”<*DETAIL*>” and press the ENTER key. Then each setting screen is displayed. Press the PREV key to get out of these screens.

```
13 Use PPABN signal:           <********>
14 WAIT timeout:                30.00 sec
15 RECEIVE timeout:             30.00 sec
```

ENTER

6 When you change the setting that the cold start must be done after a setting is changed, the following message is displayed. In that case perform the cold start.(See Section 5.2,"Turning on the Power and Jog Feed")

```
Please power on again
[ TYPE ]
```

3.15 Setting the General Items

[6 SETUP General] has the following items.

- Break on hold
- Current language
- Ignore Offset command
- Ignore Tool–offset

Table 3–8. Setting the general items

ITEMS	DESCRIPTIONS
Break on hold	Specifies whether to issue an alarm and turn off the servo alarm when the HOLD key is pressed. – If the function is DISABLED, no alarm is issued when the operation is halted by the HOLD key (standard setting). – If the function is ENABLED, an alarm is issued and the servo power is turned off, when the operation is halted by the HOLD key. To be ENABLE this function power need to be on again. **WARNING** **Not all axes are equipped with a brake. The brake on hold function has no effect on an axis without brake even if the function is enabled. Before the brake on hold function is enabled, it should be checked which axis has a brake. Otherwise, injury would occur.**
Current language	The current language is set to ”DEFAULT” by standard setting. Changing the current language requires special work. Usually, the standard setting should be used.
Ignore Offset command	Specifies whether to ignore the offset command (Section 4.3.5 ”Additional motion instructions”). – If the function is DISABLED, the robot moves to the position for which the offset command has been executed (standard setting). – If the function is ENABLED, the robot moves to the taught position (for which the offset command has not been executed). See Section 4.3.5 ”Additional motion instructions” for details of the Offset command.
Ignore Tool–offset	Specifies whether to ignore the tool offset command (Section 4.3.5 ”Additional motion instructions”). – If the function is DISABLED, the robot moves to the position for which the tool offset command has been executed (standard setting). – If the function is ENABLED, the robot moves to the taught position (for which the tool offset command has not been executed). See Section 4.3.5 ”Additional motion instructions” for details of the Tool–offset command.

Procedure 3–25 Setting the general items

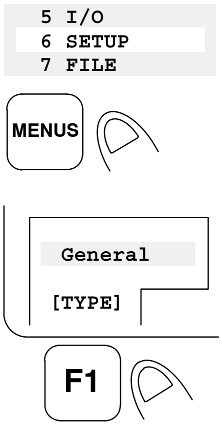
Step

1 Press the MENUS key. The screen menu is displayed.

2 Select 6 (SYSTEM).

3 Press F1 [TYPE]. The screen change menu is displayed.

4 Select General.



General item setting screen

```
SETUP General                  JOINT  30 %
                                      1/4
  1 Break on hold:          DISABLED
  2 Current language:       DEFAULT
  3 Ignore Offset command:  DISABLED
  4 Ignore Tool_offset:     DISABLED

[ TYPE ]                 ENABLED DISABLED
```

5 Place the cursor on the target field, and select the function key menu.

6 If the value for the break on hold function is re–set, to make the new setting effective, turn the controller off and on again in cold start mode. The setting of the other functions is made effective immediately when they are re–set.

3.16 Other Settings

The other settings are specified at [6 SYSTEM Variables] on the system variable screen.

- Override restore function

Override restore function

The override restore function is a function that decreases the speed override to a prescribed value when a safety fence is opened and the *SFSPD input is turned off, but restores the speed override immediately when the safety fence is closed. This function is effective under the following conditions:

- $SCR.$RECOV_OVRD = TRUE. (A control start is required.)
- The system is in remote control state.
- The speed override is not changed while the safety fence is open.

Other items are set up on the system variable screen, [6 SYSTEM Variables]. To specify system variables, see the appropriate appendix (→Appendix D, ”System Variables”).

4. PROGRAM STRUCTURE

This chapter describes the program structure and program instructions.

☐ Contents of this chapter

A handling application program consists of user–coded instructions for performing handling work, and other associated information. A program contains program information specifying how handling work is to be performed, and also contains program detail information defining program attributes.

Figure 4–1. Program Information Screen

```
 Program detail                 JOINT  30 %
                                      1/6
  Creation Date:            10-MAR-1998
  Modification Date:        11-MAR-1998
  Copy Source:        [****************]
  Positions: FALSE  Size:      312 Byte
   1  Program name:          [SAMPLE3 ]
   2  Sub Type:       [           None]
   3  Comment:        [SAMPLE PROGRAM 3]
   4  Group Mask:           [1,*,*,*,*]
   5  Write protect:         [      OFF]
   6  Ignore pause:          [      OFF]
    END   PREV  NEXT
```

Program detail information consists of the following information items:

- Attribute–related information items such as a creation date, modification date, a copy source file name, presence/absence of position data, and program data size.
- Information items related to an execution environment such as a program name, subtype, comment, group mask, write protection and interruption disable.

Figure 4–2. Program Selection Screen

Memory available capacity

```
 Select                         JOINT  30 %
              58740 bytes free      1/7
  No.  Program name        Comment
   1  SAMPLE1           [SAMPLE PROGRAM 1]
   2  SAMPLE2           [SAMPLE PROGRAM 2]
   3  SAMPLE3           [SAMPLE PROGRAM 3]
   4  PROG001           [PROGRAM001      ]
   5  PROG002           [PROGRAM001      ]
   6  CLAMP1            [CLAMP OPEN      ]
   7  CLAMP2            [CLAMP CLOSE     ]
 [ TYPE ] CREATE DELETE  MONITOR [ATTR ]>
 COPY     DETAIL  LOAD    SAVE    PRINT   >
```

Attribute

Program name

Figure 4–3. Program Edit Screen

Program name
Line number
Program statement
Motion instruction
Macro instruction
Abort instruction
Program end symbol

```
 SAMPLE1                          JOINT 10%
                                      1/8
 1: J  P[1] 100% FINE
 2:  HAND1CLOSE
 3: J  P[2] 70% CNT50
 4: L  P[3] 500mm/sec CNT10
 5:  HAND1OPEN
 6: L  P[4] 500mm/sec CNT10
 7:  HAND1CLOSE
 8:  ABORT
 [End]

  POINT                          TOUCHUP >
```

A program consists of the following information:

- Line number assigned to each program command
- Motion instructions specifying how and where the robot is to move
- Program instructions including the following:
  - Palletizing instructions for performing the palletizing function
  - Instructions for storing numerical data in registers (register instructions)
  - Instructions for storing robot position data in position registers (position register instructions)
  - I/O instructions to output and input signals to and from peripheral devices
  - Branch instructions for changing the flow of program control when a defined condition is satisfied (IF, JMP/LBL, CALL/END)
  - Wait instructions for suspending program execution
  - Skip condition instruction for operating the robot until a signal is received. If the signal is not received, a branch to a specified command occurs. If the signal is received, the next command is executed, cancelling the operation.
  - Program comments
  - Other instructions
- Program end symbol indicating that the program contains no more instructions

Program detail information is set on the program information screen. (See Subsection 5.3.1 and 5.5.) A program is registered on the program registration screen (See Subsection 5.3.1.) A program is created and changed on the program edit screen. (See Sections 5.3 and 5.4.)

4.1 Program Detail Information

Program detail information names a program and defines the attributes of the program. Program detail information consists of the following items:

- Attribute–related information items such as a creation date, modification date, a copy source file name, presence/absence of position data, and program data size.
- Information items related to an execution environment such as a program name, subtype, comment, group mask, write protection and interruption disable.

The program information screen is used to set program detail information. The program information screen is displayed by selecting F2 (DETAIL) on the program selection screen. (For program detail information setting, see Subsection 5.3.1 and 5.5)

Moreover, program comment,a write protection,a modification date, a memory size of the program,and a copy source settings, can be displayed on the selection screen by pressing F5,[ATTR] and selecting the item from a pull up menu.

```
 1 SELECT
 2 EDIT
```

MENUS

```
Program detail                    JOINT  30 %
                                         1/6
 Creation Date:             10-MAR-1998
 Modification Date:         11-MAR-1998
 Copy Source:        [****************]
 Positions: FALSE  Size:       312 Byte
  1  Program name:           [SAMPLE3 ]
  2  Sub Type:       [             None]
  3  Comment:        [SAMPLE PROGRAM 3]
  4  Group Mask:            [1,*,*,*,*]
  5  Write protection
  6  Interruption disable
   END  PREV  NEXT
```

4.1.1 Program name

A program name is specified to identify a program stored in the memory of the control unit. A single control unit cannot contain two or more programs with the same program names.

Length

A program name must consist of one to eight characters. A unique name must be assigned to each program.

Usable characters

Character: Alphabetic characters, Number: 0 to 9. No program name can start with a number. Symbol: Underscore (_) only. The at mark (@) and asterisk (*) cannot be used.

Informative name

A program should be named so that purpose or function of the program can be known from its name.

When a program for spot workpiece A is to be named, for example, “HAND–A” is a good name to assign to the program since it indicates the function of the program.

NOTE Observe the following when writing a program for automatic operation using RSR or PNS. Otherwise, the program will not run.

- A program using RSR must be named RSRnnnn, where nnnn is a 4–digit number. Example: RSR0001.
- A program using PNS must be named PNSnnnn, where nnnn is a 4–digit number. Example: PNS0001.

4.1.2 Program comment

When a new program is created, a program comment can be added to the program name. A program comment is used to describe additional information to be displayed on the selection screen together with the program name.

Length

A program comment must consist of one to sixteen characters.

Usable characters

Character: Alphabetic characters, Number: 0 to 9 Symbol: Underscore (_), at mark (@), and asterisk (*)

Informative comment

A program comment should describe the purpose or function of the program.

4.1.3 Subtype

Subtype is used to set a type of program. The following subtypes are available:

- **Job (JB)**: This represents a main program that can be started using a device such as a teach pendant. Process programs are called in a main program for execution.
- **Process (PR)**: This represents a subprogram that is called by a job program for execution of a particular job.
- **Macro (MR)**: This represents a program for executing a macro instruction. The subtype of a program registered on the macro instruction setting screen is automatically set to MR.
- **State**: Specify this when creating a conditional program with the state monitoring function.

4.1.4 Group mask

A motion group sets up an operation group of a program. An operation group represents a group of different axes (motors) used for independent robots, positioning tables, and other jigs.

NOTE A motion group must be set before the program is executed.

The R–J3*i* Mate control unit can divide up to 8 axes into up to three operation groups and control those groups simultaneously. A single group can control up to 8 axes (multimotion function). If the system has only one operation group, the default motion group is group 1 (1, *, *, *, *).

For a program that has no motion group (that is, a program involving no robot motion), this item is to be specified as (*, *, *, *, *). A program that has no motion group can be started even when the system is not ready for operation.

The system is ready for operation when the following ready conditions are satisfied:

- The peripheral I/O, ENBL input, is on.
- The peripheral I/O, SYSRDY output, is on (With the servo power is on).

4.1.5 Write protection

Write protection specifies whether the program can be modified.

- When this item is set to ON, no data can be added to the program, and the program cannot be modified; that is, the program is write protected. When a program has been created, and its operation is confirmed, the user can set this item to ON to prevent the program from being modified by the user or someone else.

NOTE When this item is set to ON, other items in the program detail information (Program name, Comment, Sub Type, Group Mask, Ignore pause) cannot be changed.

- When this item is set to OFF, the program can be modified; that is, program instructions can be added to the program, and existing instructions can be modified. Write protection is normally set to OFF as standard.

4.1.6 Interruption disable

Interruption disable (ignore pause) prevents a program being executed and not having the motion group from being interrupted by an alarm (with a severity of SERVO or lower), emergency stop, or halt. When these signals are to be ignored, set interruption disable to ON.

When interruption disable is set to ON, a program being executed can only interrupted by an abort instruction in the program or an alarm with a severity higher than SERVO. (See Subsection 4.13.2.)

> **WARNING**
> **When interruption disable is set to ON, a program being executed cannot be interrupted by pressing the emergency stop or halt button on the teach pendant or operator's panel.**

Procedure 4–1 Program Detail Information

Step

1 Press the MENUS key. The screen menu is displayed.

2 Select 1(SELECT). The program selection screen is displayed.
The program selection screen can be displayed by pressing the SELECT key without using above steps.

```
1 SELECT
2 EDIT
```

MENUS

```
Select                              JOINT  30 %
              58740 bytes free      1/7
 No.  Program name        Comment
  1  SAMPLE1          [SAMPLE PROGRAM 1]
  2  SAMPLE2          [SAMPLE PROGRAM 2]
  3  SAMPLE3          [SAMPLE PROGRAM 3]
  4  PROG001          [PROGRAM001      ]
  5  PROG002          [PROGRAM001      ]
  6  CLAMP1           [CLAMP OPEN      ]
  7  CLAMP2           [CLAMP CLOSE     ]

[ TYPE ] CREATE DELETE  MONITOR [ATTR ]>
COPY    DETAIL LOAD    SAVE    PRINT   >
```

3 **Switching the screen using sub type**
To specify the program to be displayed for the sub type,press F1,[TYPE] and select the sub type of the program you want to display.

- All : All the programs are displayed.
- Job : Only job programs are displayed.
- Process : Only process programs are displayed.
- Program : All the programs except the macro are displayed.
- Macro : Only macro programs are displayed.

```
1 All
2 Jobs
3 Processes
4 TP Programs
5 Macro

TYPE
```

F1

```
SAMPLE1           LINE 1         ABORTED
Select                              JOINT  30 %
              61276 bytes free      1/4
  1  SAMPLE1          [SAMPLE PROGRAM 1]
  2  SAMPLE2          [SAMPLE PROGRAM 2]

[ TYPE ] CREATE DELETE  MONITOR [ATTR ]>
```

4 **Switching the display using the attribute**
To specify the program attribute to be displayed,press F5,[ATTR] and select the attribute type of the program you want to display.

- Comment : The comment is displayed.
- Protection : The setting of the write protection is displayed.
- Last Modified : The latest date of the modification is displayed.
- Size : The number of the line and the program size are displayed.
- Copy Source : The name of the copy source program is displayed.

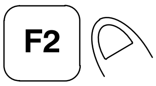
```
1 Comment
2 Protection
3 Last Modified
4 Size
5 Copy Source

             ATTR
```

F5

```
Select                              JOINT  30 %
              58740 bytes free      1/2
 No.  Program name          Size
  1  SAMPLE1          [      32/      839]
  2  SAMPLE2          [      12/     1298]

[ TYPE ] CREATE DELETE  MONITOR [ATTR ]>
```

5 **Program Detail Screen**
Press NEXT,> and press F2,DETAIL in the next page. The program detail screen is displayed.

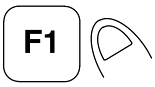
```
COPY     DETAIL  LOAD
```




```
Program detail                    JOINT  30 %
                                         1/6
 Creation Date:              10-MAR-1998
 Modification Date:          11-MAR-1998
 Copy Source:         [*****************]
 Positions: FALSE  Size:         312 Byte
  1  Program name:            [SAMPLE3  ]
  2  Sub Type:        [             None]
  3  Comment:         [SAMPLE PROGRAM 3]
  4  Group Mask:           [1,*,*,*,*]
  5  Write protect:        [      OFF]
  6  Ignore pause:         [      OFF]
   END  PREV  NEXT
```

6 When you finish setting the program header information, press F1,END.

```
END  PREV  NEXT
```



4.2 Line Number, Program End Symbol, and Argument

Line number

A line number is automatically inserted in front of an instruction when it is added to a program. When an instruction is deleted, or an instruction is moved to another location, the lines of the program are renumbered in ascending order; that is, the first line is numbered as 1, the second line is numbered as 2, and so forth.

When a program is to be modified, the cursor can be used to specify a line or a range of lines for movement or deletion by line number.

The user can make the cursor move to a desired line number by specifying a line number (with the ITEM key).

Program end symbol

The program end symbol ([End]) is automatically displayed on the line after the last instruction of a program. Whenever a new instruction is added, the program end symbol moves downward on the screen. As a result, it is always displayed on the last line.

When the execution of a program reaches the program end symbol after the last instruction in the program is executed, the program execution automatically returns to the first line of the program for termination.However, when the setting of ”Return to top of program” is FALSE, the cursor stays at the last line of the program after program execution is completed. (See Section 3.14 ”System Config Menu”.)

A description of the program instructions required to create and change a program follows. (For how to create a program, see Section 5.3. For how to change a program, see Section 5.4.)

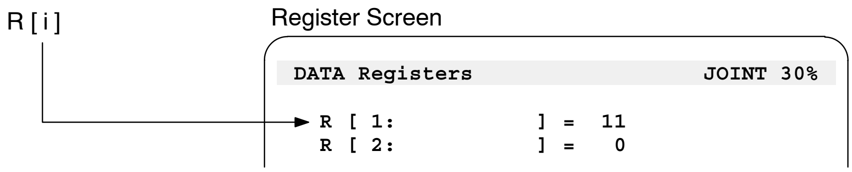
Argument i

Argument i is an index used in teaching control instructions (program instructions other than motion instruction). Some arguments are specified directly; others are specified indirectly. In direct specification, an integer from 1 to 32767 is usually specified. The range of values used depends on the type of instruction. In indirect specification, the register number of a register is specified.

Figure 4–4. Format of Argument i

Argument i

- Direct specification : Number.(Example: R[i])
- Indirect specification : Uses the value of the register with register number i as the argument.(Example: R[R[i]])



Procedure 4–2 Program Edit Screen

Condition ■ The teach pendant must be enabled.

Step

1 Display the program selection screen.

2 Move the cursor to the program you want to edit and press ENTER key. The program edit screen is displayed.

```
Select
                 61276
 No.  Program name
  1  SAMPLE1
  2  SAMPLE2
  3  SAMPLE3
  4  PROG001
```

ENTER

```
SMPLE1                          JOINT  30 %
                                       1/6
     1:J P[1] 100% FINE
     2:J P[2] 70% CNT50
     3:L P[3] 1000cm/min CNT30
     4:L P[4] 500mm/sec FINE
     5:J P[5] 100% FINE
 [End]

 POINT                             TOUCHUP>
```

3 **Moving the cursor**

To move the cursor,use the arrow keys such as up,down right,and left. To move quickly through the information, press and hold the SHIFT key and press the down or up arrow keys.

4 To select the line number, press the ITEM key and enter the line number you want to move the cursor.

ITEM 5 ENTER

```
SMPLE1                          JOINT  30 %
                                       5/6
     4:L P[4] 500mm/sec FINE
     5:J P[5] 100% FINE
 [End]
```

5 **Entering the numerical value**

To enter the numerical value,move the cursor to the argument and press the numerical value keys. When you are finished,press the ENTER key.

```
PROG2

   9:L P[5] 100% FINE
  10:  DO[...]=...
```

1 ENTER

```
PROG2                           JOINT  30 %
                                     10/11
   10:  DO[  1]=...
 [End]
 Enter value
         DIRECT INDIRECT[CHOICE]
```

6 To use the indirect addressing with the register,press F3,INDIRECT.

```
         DIRECT INDIRECT
```

F3

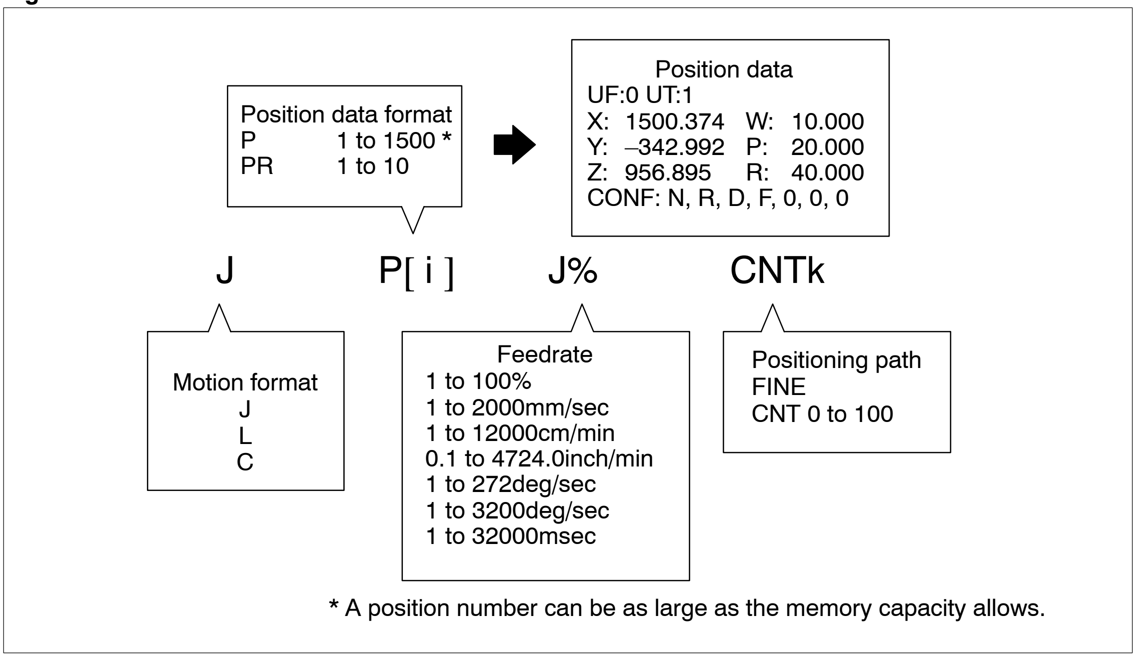
```
PROG2                           JOINT  30 %
                                     10/11
   10:  DO[R[1]]=...
 [End]Enter value

         DIRECT INDIRECT[CHOICE]
```

4.3 Motion Instructions

A motion instruction moves a robot tool to a specified point within the operating area at a specified feedrate and in a specified traveling mode. The items listed below must be specified in a motion instruction. The format of a motion instruction is shown in Figure 4–5.

- Motion format: Specifies how to control the path of motion to a specified position.
- Position data: Teaches a position to which the robot is to move.
- Feedrate: Specifies the feedrate of the robot.
- Positioning path: Specifies whether to position the robot at a specified point.
- Additional motion instruction: Specifies the execution of an additional instruction while the robot is in motion.

Figure 4–5. Motion Instructions



In teaching a motion instruction, a standard motion instruction is selected using either the F1 or F5 function key. (For modifying a standard motion instruction, see Subsection 5.3.2. For teaching a motion instruction, see Subsection 5.3.3. For changing a motion instruction, see Subsection 5.4.2.)

```
POINT   SINGLE   DUAL   BACKUP   TOUCHUP >
```

- Press F1 POINT to program an operation instruction.
- Press F5 TOUCHUP to re–program programmed position data.

4.3.1 Motion format

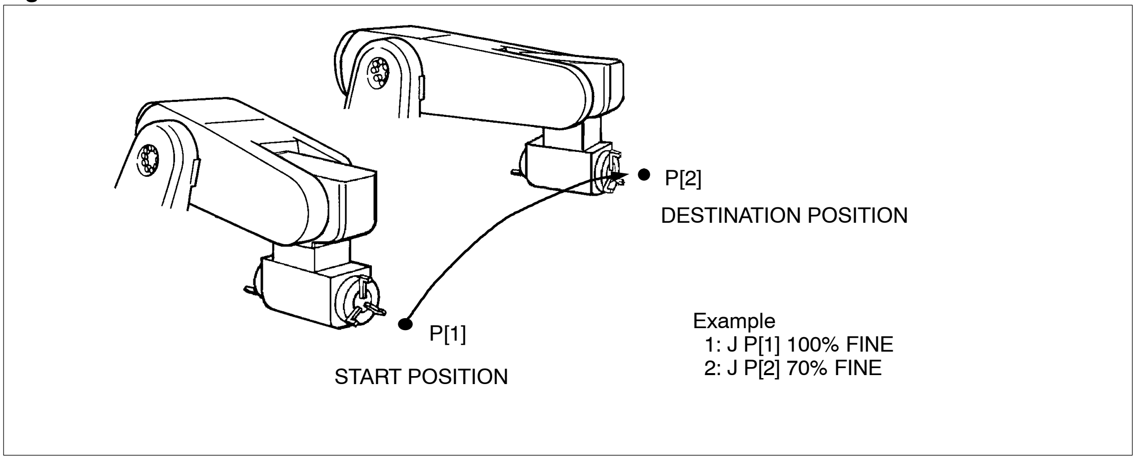
For the motion format, the path of motion to a specified position is specified. Three options are available: joint motion, which does not exercise path/attitude control and linear motion and circular motion, which exercise path/attitude control.

- Joint motion (J)
- Linear motion (including the rotation motion)(L)
- Circular motion (C)

Joint motion J

The joint motion mode is the basic mode for moving the robot to a specified position. The robot accelerates along or about all axes, moves at a specified feedrate, decelerates, and stops at the same time. The path of motion is usually non–linear. The motion format is specified to teach an end point. A percentage of a maximum feedrate is specified as the feedrate of joint motion. The attitude of a tool being moved is not controlled.

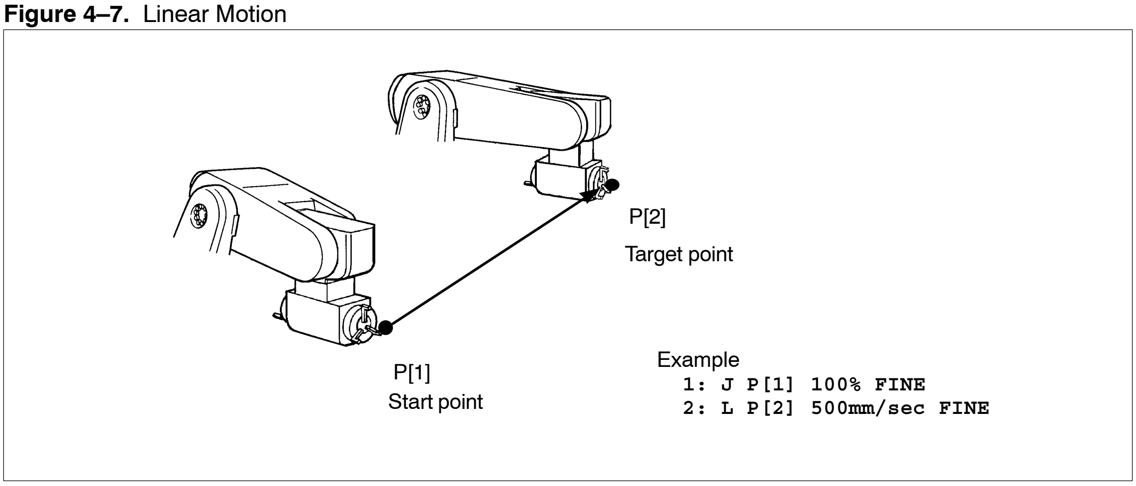
Figure 4–6. Joint Motion



Linear motion L

The linear motion mode controls the path of tool center point (TCP) motion from a start point to an end point; the tool center point moves linearly. The motion format is specified to teach an end point. For linear feedrate specification, a desired option must be chosen from mm/sec, cm/min, and inch/min. The attitude of a tool being moved is controlled by distinguishing the attitude at a start point from the attitude at a target point.
The LR Mate 100*i* MODEL B uses five–axis control, so that attitude control is exercised only when the flange surface is facing directly upward or downward.

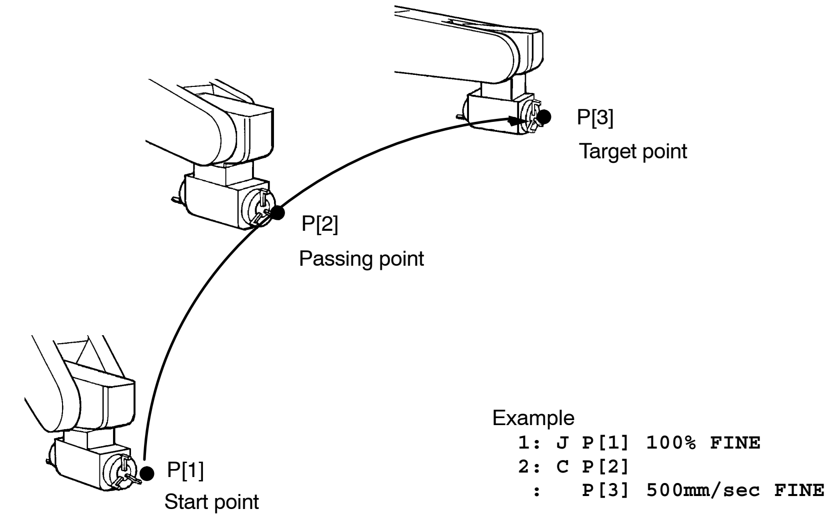
Figure 4–7. Linear Motion



Circular motion

The circular motion mode controls the path of tool center point motion from a start point to an end point through a passing point.
Both a passing point and a target point are taught in one instruction.
For circular feedrate specification, a desired option must be chosen from mm/sec, cm/min and inch/min.
The attitude of a tool being moved is controlled by distingushing the attitude at a start point from the attitude at a target point.

Figure 4–8. Circular motion

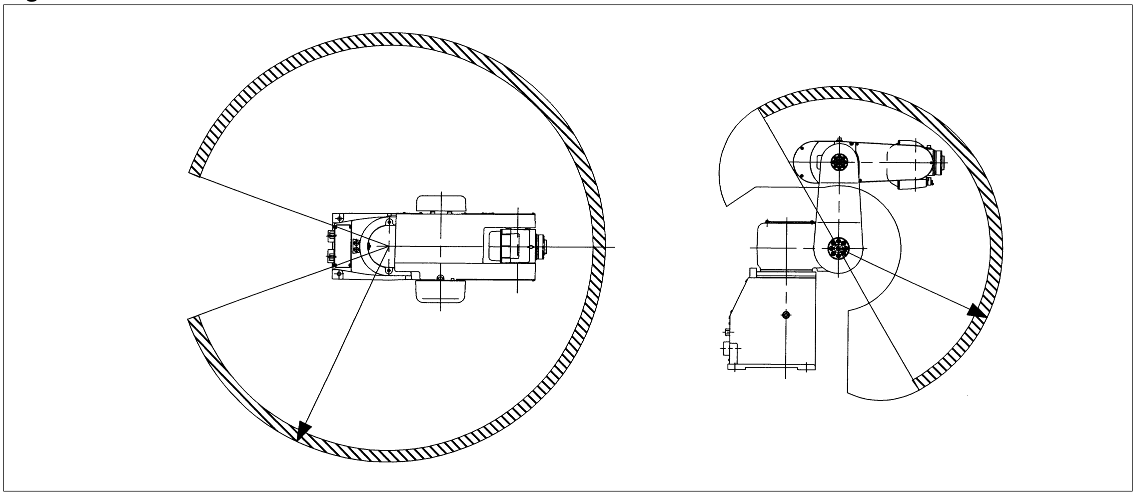


Motion specific to the LR Mate 100*i* MODEL B

Since the LR Mate 100*i* MODEL B is a five–axis robot, it has several unassumable attitudes. By this reason, the structural restrictions described below are applied to the execution of attitude control motion (linear motion/circular motion). To overcome these restrictions, joint motion or wrist joint motion (WITHOUT ATTITUDE) can be used.

- In linear jog feed, when the robot moves it assumes the attitude which most closely approximates the unassumable attitude. Care must be taken because this may cause the attitude of the tool to change very quickly. Rotational jog feed is impossible because the W, P, and R keys for rotation are disabled.
- The LR Mate 100*i* MODEL B can exercise attitude control only when the flange surface is facing directly upward or downward, and the Z–axis in the tool coordinate system is perpendicular to the flange surface. In other cases, care must be taken because when the robot moves it assumes the attitude which most closely approximates the unassumable attitude.
- When the flange surface faces directly upward or downward, the configuration of the wrist changes: the configuration is NOFLIP when the flange surface is facing the robot, and the configuration is FLIP when it is not. I n other cases, attitude control is impossible, which causes the configuration at an end point to differ from that at a start point. In such a case, use joint motion or wrist joint motion (WITHOUT ATTITUDE).
- When the control point (intersection of the rotation center axes of the J4–axis and J5–axis) passes near the rotation center axis of the J1–axis, the J1–axis or J5–axis may turn at high speed, so the user must be careful.
- A stroke limit error occurs if the angular displacement per linear or circular motion is 180, or more with respect to the J5–axis. In such a case, the angular displacement must be changed to less than 180, per linear or circular motion.
- Precision in path control and palletizing tends to decrease near the periphery of the operating area of the robot when the J2–axis arm is aligned with the J3–axis arm. Therefore, the user should avoid teaching near the periphery. If teaching near the periphery is unavoidable, the user should teach extra points.

Figure 4–9. Area with Decreased Precision



4.3.2 Position data

Position data includes the positions and attitudes of the robot. When a motion instruction is taught, position data is written to the program at the same time.

Position data is classified into two types. One type consists of joint coordinates in a joint coordinate system. The other type consists of Cartesian coordinates representing tool positions and attitudes in work space. Standard position data uses Cartesian coordinates.

Cartesian coordinates

Position data consisting of Cartesian coordinates is defined by four elements: the position of the tool center point (origin of the tool coordinate system) in a Cartesian coordinate system, the inclination of the axis along which the tool moves (tool coordinate system), configuration, and a Cartesian coordinate used. A Cartesian coordinate system may be a world coordinate system. How to select the coordinate systems is explained later in this subsection.

Figure 4–10. Position Data (Cartesian Coordinates)

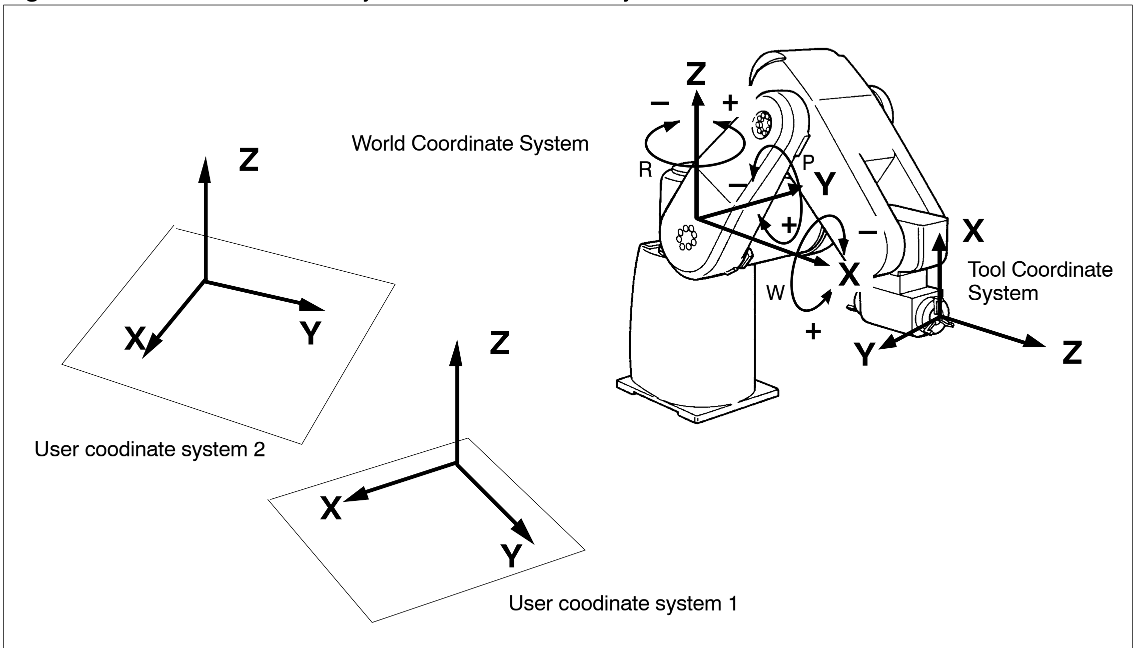
UF , UT , (X , Y , Z , W , P , R) , Configuration

- UF: User coordinate system number
- UT: Tool coordinate system number
- X , Y , Z: Position
- W , P , R: Attitude
- Configuration: Configuration

Position and attitude

- The position (x,y,z) represents the three–dimensional position of the tool center point (origin of the tool coordinate system) in the Cartesian coordinate system.
- The attitude (w,p,r) represents angular displacements about the X–axis, Y–axis, and Z–axis in the Cartesian coordinate system.

Figure 4–11. World Coordinate System/Tool Coordinate System



Configuration

A configuration represents the attitude of the robot. Several configurations are available which meet the condition of Cartesian coordinates (x,y,z,w,p,r). The turn number and joint placement of each axis must be specified.

Figure 4–12. Configuration

J4 J5 J6

(F , , , T , , 0 ,)

Joint placement — Turn number

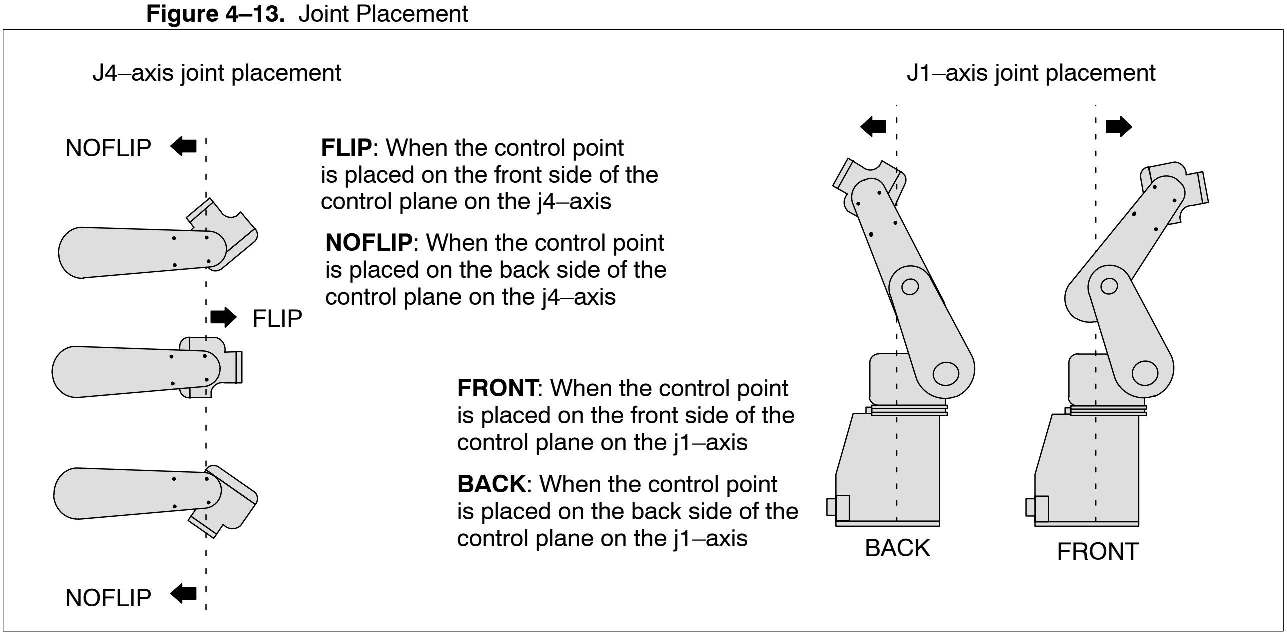
- **F**LIP / **N**OFLIP — Wrist up/down
- FRON**T** / **B**ACK — Arm front/back
- 1: 180° to 539°; 0: –179° to 179°; –1: –539° to –180°

Joint placement

Joint placement specifies the placement of the wrist and arm. This specifies which side the control point of the wrist and arm is placed on against the control plane. When a control point is placed on the control plane, the robot is said to be placed at a singular, or to be taking a peculiar attitude. At the singular, since the configuration can not be decided to one by the specified cartesian coordinate values, the robot can not move.

- J4–axis joint placement specifies on which side of the control plane the wrist control point (flange surface center point) is placed.
- J1–axis joint placement specifies on which side of the control plane the arm control point (intersection of the rotation center axes of the J4–axis and J5–axis) is placed.
- In linear motion or circular motion attitude control is possible only when the J4–axis is placed at a singular. This means that attitude control is enabled only when the flange surface is facing directly upward or downward, and the Z–axis in the tool coordinate system is perpendicular to the flange surface.
- During linear or circular motion, the tool cannot pass through a singular point (the joint placement cannot be changed). In this case, execute a joint motion. To pass through a singular point on the wrist axis, a wrist joint motion (Wjnt) can also be executed.

Figure 4–13. Joint Placement



Turn number

Turn number represents the number of revolutions of the J5–axis. Each axis returns to the original position after one revolution. So, specify how many turns have been made. Turn number is 0 when each axis is at an attitude of 0°.

When programmed linear motion or circular motion is executed, the robot tool moves toward the target point while adopting an attitude very similar to that at the start point. The number of revolutions performed at the target point is selected automatically. The actual number of revolutions performed at the target point may differ from the number specified in the position data.

Cartesian coordinate system reference

In playback of position data consisting of Cartesian coordinates, a Cartesian coordinate system reference checks the coordinate system number of a Cartesian coordinate system to be used.

If the coordinate system number (a number from 0 to 10) specified in the position data does not match the coordinate system number currently selected, the program is not executed for safety, and an alarm is issued.

A coordinate system number is written into position data in position teaching.

To change a coordinate system number after it has been written, use the tool replacement/coordinate replacement shift function.

Tool coordinate system number (UT)

The tool coordinate system number specifies the coordinate system number of a mechanical interface coordinate system or tool coordinate system. Thus, the coordinate system of the tool is determined.

- 0 : The mechanical interface coordinate system is used.
- 1 to 10 : The tool coordinate system of a specified tool coordinate system number is used.
- F : The coordinate system of the tool coordinate system number currently selected is used.

User coordinate system number (UF)

The user coordinate system number specifies the coordinate system number of a world coordinate system or user coordinate system. Thus, the coordinate system of work space is determined.

- 0 : The world coordinate system is used.
- 1 to 9 : The tool coordinate system of a specified tool coordinate system number is used.
- F : The coordinate system of the tool coordinate system number currently selected is used.

Detail position data

To display the detail position data, position the cursor to the position number, then press the F5 (POSITION) key.

```
 SAMPLE1

   1: J  P[1] 100%
   2: J  P[2] 70%

COMMENT CHOICE POSITION
```


```
Position Detail                 JOINT 30%
P[2]  GP:1   UF:0  UT:1 CONF: N  T,0
 X: 1500.374  mm    W:  40.000  deg
 Y: -242.992  mm    P:  10.000  deg
 Z:  956.895  mm    R:  20.000  deg
SAMPLE1
```

Switching the coordinate system check function

The coordinate system check function allows the user to perform FWD/BWD execution easily between two points with different coordinate system numbers. By changing the setting of the following system variable, this function can be switched to one of three specifications.

Setting of the system variable	Description
$FRM_CHKTYP = –1	Disables FWD/BWD execution between two points having different coordinate system numbers.
$FRM_CHKTYP = –2	Enables FWD/BWD execution between two points having different numbers.
$FRM_CHKTYP = 2	Enables FWD/BWD execution between two points having different numbers, and changes the current coordinate system number ($MNUFRAME_NUM or $MNUTOOL_NUM) to the number specified in the position data in the program.

The system variable is explained, using a specific program as an example.

Example 1: UTOOL_NUM = 1
2: JP [1] 100% FINE (specified with P [1] UT = 1)
3: JP [2] 100% FINE (specified with P [2] UT = 2)

– If $FRM_CHKTYP = –1,
  FWD: An alarm is generated if the coordinate system numbers differ on the third line.
  BWD: If the currently selected tool coordinate system number is 2, an alarm is generated when the second line is executed after BWD execution on the third line.
– If $FRM_CHKTYP = –2,
  FWD: An alarm is not generated on the third line. The third line is executed with a tool coordinate system number of 2. (Operation is performed at the specified position.)
  BWD: As with FWD, an alarm is not generated.
– If $FRM_CHKTYP = 2,
  An alarm is not generated in the same way as for $FRM_CHKTYP = –2.
  FWD: An alarm is not generated on the third line. The third line is executed with a tool coordinate system number of 2. Immediately after the start of the operation for the third line, the tool coordinate system number of the system is changed to 2.
  BWD: An alarm is not generated on the second line. Immediately after the start of the operation for the second line, the tool coordinate system number of the system is changed to 1.

NOTE Regardless of the value of $FRM_CHKTYP, BWD operation between arcs having different coordinate system numbers result in an alarm.

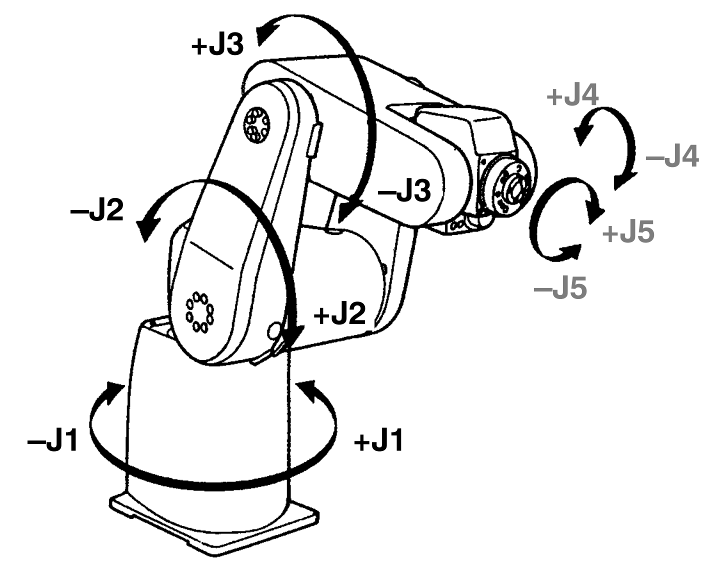
Joint coordinates

Position data consisting of joint coordinates is defined using angular displacements with respect to the joint coordinate system on the base side of each articulation.

Figure 4–14. Position Data (Joint Coordinates)

$$(\underbrace{J1 , J2 , J3}_{\text{Main axis}} , \underbrace{J4 , J5}_{\text{Wrist axis}} , \underbrace{E1 , E2 , E3}_{\text{Additional axis}})$$

Figure 4–15. Joint Coordinate System



detail position data

Detailed position data is displayed when F5 POSITION is pressed. You can chose between Cartesian coordinates and axial coordinates by pressing F5 [REPRE].

```
SAMPLE1

   1: J  P[1] 100%
   2: J  P[2] 70%

      CHOICE POSITION
```

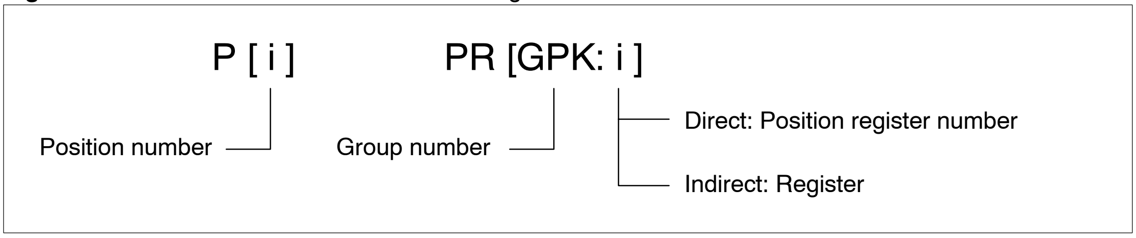
F5

```
Position Detail                 JOINT 30 %
P[2]   UF:0  UT:1
 J1:     .125  deg    J4:  -95.000  deg
 J2: 23.590  deg    J5:      .789  deg
 J3: 30.300  deg
SAMPLE1
```

Position variable and position register

In a motion instruction, position data is represented by a position variable (P[i]) or position register (PR[i]). Usually, a position variable is used.

Figure 4–16. Position Variable and Position Register



Example

```
1: J  P[12] 30% FINE
2: L  PR[1] 300mm/s CNT50
3: L  PR[R[3]] 300mm/s CNT50
```

Position variable

The position variable is the variable usually used to hold position data. In motion instruction teaching, position data is automatically saved.

When Cartesian coordinates are taught, the following Cartesian coordinate system and coordinate system number are used:

- Coordinate system of the tool coordinate system number currently selected (UT = 1 to 10)
- World coordinate system (UF = 0)(When $USE_UFRAME is FALSE)

In playback, the following Cartesian coordinate system and coordinate system number are used:

- Coordinate system of the user coordinate system number currently selected (UF = 0 to 9) (When $USE_UFRAME is TRUE)

NOTE The system variable $USE_UFRAME cannot be used if the user coordinate system input function option is not provided.

When a position is copied,

- Coordinate system with the specified tool coordinate system number (UT = 1 to 10)
- Coordinate system with the specified user coordinate system number (UF = 0 to 9)

Position register

The position register functions as a general–purpose register for holding position data. (For position teaching using a position register, see Section 7.4.)

When Cartesian coordinates are taught, the following Cartesian coordinate system and coordinate system number are used:

- Coordinate system of the tool coordinate system number currently selected (UT = F)
- Coordinate system of the user coordinate system number currently selected (UF = F)

In playback, the following Cartesian coordinate system and coordinate system number are used:

- Coordinate system of the tool coordinate system number currently selected (UT = F)
- Coordinate system of the user coordinate system number currently selected (UF = F)

Position number

The position number is used to reference a position variable. A position number is automatically assigned each time a motion instruction is taught and it is reflected in the program. For example, the first position number assigned is P[1], the second P[2], and so on.

When a motion instruction is added, it is assigned the position number obtained by incrementing the position number assigned to the most recently added motion instruction by one, regardless of where the newly added instruction is placed in the program. However, this is not the case when a position number is changed.

When a position is deleted, the position numbers of other taught points remain unchanged. However, this is not the case when a position number is changed. (For changing a position number, see Section 5.4 ”Changing a Program”.)

A comment consisting of up to 16 characters can be described for a position number or position register number. To add a comment,press the ENTER key when the cursor is at the position number or position register number.

Example

```
4: J  P[11: APPROACH POS ] 30% FINE
5: L  PR[1: WAIT POS  ] 300mm/s CNT50
```

4.3.3 Feedrate

The feedrate specifies the speed at which the robot moves. During program execution, the feedrate is controlled by feedrate overriding. A feedrate override value of 1% to 100% can be used.

The unit used to specify a feedrate depends on the motion format taught with a motion instruction.

NOTE The programmed travelling speed cannot exceed the allowable range of the robot. If a speed exceeding the range is programmed, a warning alarm would be issued.

J P[1] 50% FINE

When the motion type is joint, a feedrate is specified as the following:

- A percentage from 1% to 100% of the maximum feedrate is to be specified.
- When the unit is sec,specify the value from 0.1 to 3200sec as the time took for motion. This specification is required,when the time took for motion is important. An operation cannot sometimes takes place in a specified time.
- When the unit is msec,specify the value from 1 to 32000msec as the time took for motion.

L P[1] 100mm/sec FINE

If the specified motion format is linear motion or circular motion, specify a feedrate as follows:

- When the unit is mm/sec, specify a feedrate from 1 to 2000 mm/sec.
- When the unit is cm/min, specify a feedrate from 1 to 12000 cm/min.
- When the unit is inch/min, specify a feedrate from 0.1 to 4724.4 inch/min.
- When the unit is sec, specify the value from 0.1 to 3200sec as the time took for motion.
- When the unit is msec,specify the value from 1 to 32000msec as the time took for motion.

L P[1] 50deg/sec FINE

When the mode of motion is rotation about the tool center point, specify an angular displacement as follows:

- When the unit is deg/sec, specify an angular displacement from 1 to 272 deg/sec.
- When the unit is sec, specify the value from 0.1 to 3200sec as the time took for motion.
- When the unit is msec,specify the value from 1 to 32000msec as the time took for motion.

Specifying the feedrate with a register

The feedrate can be specified with a register. This allows the user to specify the feedrate for an operation instruction after calculating the feedrate using a register. The feedrate can also be specified externally, using group input (GI) or data transfer, for example.

CAUTION

This function allows the user to change the feedrate of a robot freely by setting a register. This means that the robot may operate at an unexpected speed depending on the specified register value. When using this function, therefore, specify the register value with great care during both teaching and operation.

Format in which an operation instruction is displayed when the feedrate is specified with a register

- Joint J P[1] R[i]% FINE
- Linear L P[1] R[i]mm/sec FINE
- Arc C P[1]
  P[2] R[i]mm/sec FINE
- Pallet operation instruction
  J PAL_1[A_1] R[i]% FINE
  J PAL_1[BTM] R[i]% FINE
  J PAL_1[R_1] R[i]% FINE

NOTE The pallet operation instruction is a software option of palletizing.

- Pallet operation instruction
  J PAL_1[A_1] R[i]% FINE
  J PAL_1[BTM] R[i]% FINE
  J PAL_1[R_1] R[i]% FINE

NOTE The pallet operation instruction is a software option of palletizing.

Operation group instruction

Asynchronous operation group
GP1 JP[1] R[i]% FINE
GP2 JP[1] R[i]% FINE

NOTE The operation group instruction is a software option of multimotion.

The feedrate for a standard operation instruction is also supported.

Search/replace functions

– Search function
  The search function is not supported.
  Search using register items cannot be performed.

– Replace function
  Replacement is possible with the operation statement modification item.
  Replacement using register items cannot be performed.

The additional axis feedrate for an operation addition instruction is not supported.

In program editing, a range check is not performed on the feedrate (register value).

The feedrate (register value) is not automatically converted when the feedrate unit is changed.

If the feedrate specification for an operation statement is made with a register, the read–ahead of execution is stopped. (It is possible to specify whether to stop read–ahead using a system variable. This is described later.)

If the value entered in the register is not within the upper and lower limits, or if the value is of a type other than those appropriate to a feedrate (integer/real), an alarm is generated during execution.

Unit	Allowable range	
%	1 to 100	Integer
sec	0.1 to 3200.0	(*1) Real/effective up to the first decimal place.
msec	1 to 32000	(*1) Integer
mm/sec	1 to 2000	(*1) Integer
cm/min	1 to 12000	(*1) Integer
inch/min	0.1 to 4724.2	(*2) Real/effective up to the first decimal place.
deg/sec	1 to 272	(*3) Integer

The allowable range (maximum value) differs depending on the robot type.

*1: System variable $MPR_GRP.$SPPEDLIM
*2: System variable $MPR_GRP.$SPPEDLIM/10
*3: System variable $MPR_GRP.$ROTSPEEDLIM * 180/3.1415

Read–ahead can be enabled.

If the feedrate specification for an operation statement is made with a register, the read–ahead of execution is stopped. It is possible to specify whether to stop read–ahead using the following system register. The default is FALSE (read–ahead is stopped).

$RGSPD_PREXE = TRUE: Enables read–ahead.
= FALSE: Disables read–ahead.

NOTE If the read–ahead of the register feedrate is enabled with the above system variable, it is possible that the new value is not reflected in the operating speed, causing the robot to move with the old value, depending on the timing at which the register value is changed. If read–ahead of the register feedrate is enabled, it is necessary to take appropriate measures such as interlocking or not changing the value of the register used for the feedrate during program execution.

```
10: R [1] = 100
11:J P[5] R[1]% FINE
12:R[1]=10
14:J P[6] R[1]% FINE
```

If read–ahead is enabled, 100 on line 10, not 10 on line 12, is used for the operating speed on line 14.

4.3.4 Positioning path

The positioning path defines the method of ending robot operation in a motion instruction. Three positioning path modes are available:

- FINE positioning path
- CNT positioning path
- Corner distance specification: Available only when the high–precision focus function is ordered. For an explanation of corner distance specification, see Section 4.3.6.

FINE positioning path J P[i] 50% FINE

When the FINE positioning path is specified, the robot stops at a target point before moving to the next target point.

CNT positioning path J P[i] 50% CNT50

When the CNT positioning path is specified, the robot approaches a target point but does not stop at the point and moves to the next point.

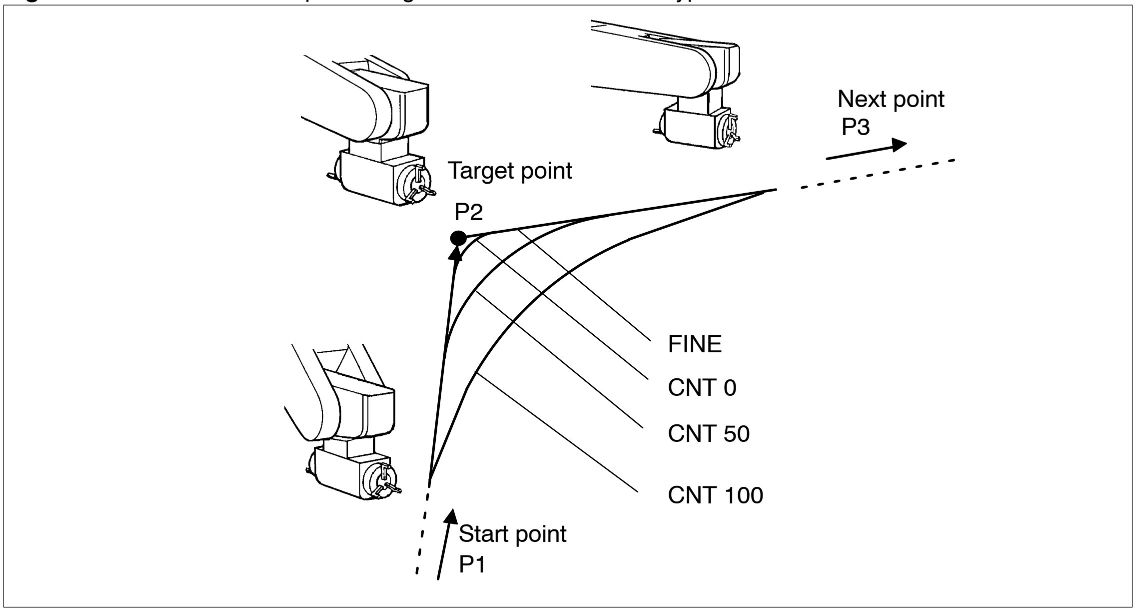
How closely the robot must approach a target point can be defined by specifying a value from 0 to 100.

When 0 is specified,the robot moves the nearest path to the destination position but moves to the next target point without stopping at the target point. When 100 is specified, the robot moves along the farthest path to the target point because the robot does not decelerate near the target point and it starts to move to the next target point soon.

NOTE When an instruction such as a wait instruction is taught, the robot stops at the target point to execute that instruction.

NOTE Several short–distance, high–speed motions that are performed continuously with CNT specified may be decelerated, even if the specified CNT value is 100.

Figure 4–17. Robot motion path using continuous termination type



4.3.5 Additional motion instructions

An additional motion instruction causes the robot to perform a particular job. The following additional motion instructions are available:

- Wrist joint motion instruction (Wjnt)
- Acceleration override instruction (ACC)
- Skip instruction (Skip,LBL[i])
- Offset condition instruction (Offset)
- Direct offset condition (Offset,PR[i])
- Tool offset instruction (Tool_Offset)
- Direct tool offset instruction (Tool_Offset, PR[i])
- Incremental instruction (INC)
- Simultaneous EV instruction (EV i%)
- Independent EV instruction (Ind.EV i%)
- Path instruction (PTH)
- Pre–execution instruction (pre–execution/post–execution)
  (→Section 9.8, "Pre–execution Instruction")
- Corner speed ratio add instruction
  For an explanation of the corner speed ratio add instruction, see Section 4.3.6.

When teaching an additional motion instruction, move the cursor after the motion instruction, then press the F4 [CHOICE] to display the list of additional motion instructions. Then select a desired additional motion instruction.

```
                JOINT 30%
                     4/5
 500mm/sec CNT10

         CHOICE
```

F4

```
Motion Modify                   JOINT 30%
1 No option          5 Incremental
2 Wrist Joint        6 Skip,LBL[ ]
3 Offset/Frames      7
4 Offset.PR[  ]      8
PROGRAM1
```

Wrist joint motion instruction L P[i] 50% FINE Wjnt

```
Motion Modify
  1 No option      5 In
  2 Wrist Joint    6 Sk
  3 Offset         7
  4 Offset,PR[ ]   8
```

The wrist joint motion instruction specifies a path control operation that does not control the attitude of the wrist. (In the standard mode, the attitude of the wrist is controlled until the end of the motion.) The wrist joint motion instruction is used when a linear motion or circular motion is specified.

When the wrist joint motion instruction is used, the attitude of the wrist changes during the motion. However, the tool center point can move along a programmed path without causing the wrist axis to invert due to a wrist axis singular point.

Acceleration override

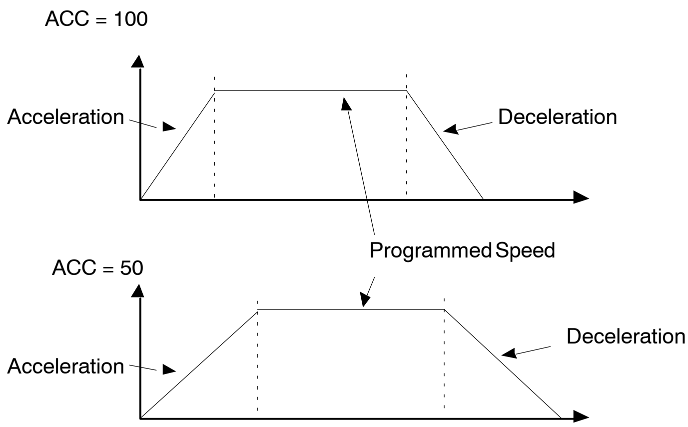
J P[1] 50% FINE ACC80

```
Motion modify
  1 No option
  2 Wrist Joint
  3 ACC
  4 Skip,LBL[]
 PROGRAM1
```

This instruction specifies the percentage of the acceleration/deceleration rate during motion. When the acceleration override is reduced, acceleration time will be long (Acceleration and deceleration are done slowly). To perform a potentially dangerous operation such as hot water scooping, use an ACC value of less than 100%. When acceleration override is raised, acceleration time will be short (Acceleration and decelerate are done quickly). For portions where the operation is felt to be very slow, use an ACC value greater than 100%.

The time used for motion from a starting point to a destination point depends on the acceleration override. The acceleration override value ranges from 0 to 150%. Acceleration override is programmed at the destination position.

Figure 4–18. Acceleration Override



CAUTION

If the acceleration override value is large, awkward movement and vibration may occur. This may cause a servo alarm. If this occurs with an operation instruction to which an acceleration/deceleration override instruction is added, either reduce the acceleration/deceleration override value or delete the accelerate/deceleration override instruction.

Skip instruction

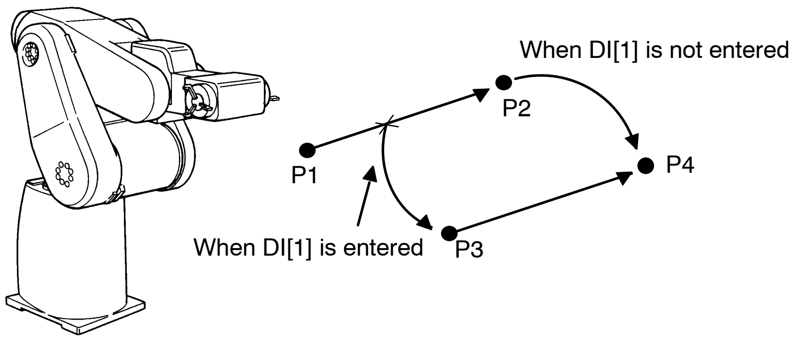
SKIP CONDITION [I/O] = [value]   J P[1] 50 FINE Skip,LBL[3]

```
             JOINT 30%
 5 Incremental
 6 Skip,LBL[  ]
 7
 8
```

A skip instruction causes a jump to a branch destination label if the skip condition is not satisfied. If the skip condition is satisfied while the robot is moving to a target point, the robot cancels the motion and program execution proceeds to the program statement on the next line. If the skip condition is not satisfied, program execution skips (jumps) to the line of the branch destination label after completion of the robot motion.

The skip condition instruction specifies, in advance, a skip condition (condition for executing a skip instruction) to be used with it. Before a skip instruction can be executed, a skip condition instruction must be executed. A skip condition once specified is valid until the execution of the program is completed, or the next skip condition instruction is executed. (For the branch instructions, see Section 4.7. For the skip condition instruction, see Section 4.9.)

Figure 4–19. Skip Instruction



Example

```
1: SKIP CONDITION DI[1] = ON
2: J  P[1] 100% FINE
3: L  P[2] 1000mm/sec FINE Skip, LBL[1]
4: L  P[3] 50% FINE
5:    LBL[1]
6: J  P[4] 50% FINE
```

High–speed skip

Function outline

(1) The position of the robot when the skip conditions are met can be stored in programmed position registers.

(2) Digital servo control stops the robot quickly by developing the maximum torque of the motor when the robot detects that the skip conditions are met.

Use method

The high–speed skip function can be used in program teaching. There is no need to set system variables.

Program teaching

a) Teaching skip conditions

The skip conditions for the high–speed skip function are taught in the same way as the ordinary skip function.

b) Teaching a high–speed skip instruction (an additional operation instruction)

In the same way as the ordinary line skip instruction, select the high–speed skip instruction from the additional operation instruction menu.

```
Motion modify            JOINT 30%    High–speed skip instruction
 1 Skip, LBL, PR                      Ordinary skip instruction
 3 Skip, LBL,
```

c. Specify the label, position register, and position storage format.

```
Skip, LBL[10], PR[5]=LPOS
                    or JPOS
```

[Sample program]

```
              :
8: SKIP CONDITION SDI[3]=ON
              :
10: LP[2]500mm/sec FINE
11: LP[3]100mm/sec
  :  SKIP, LBL[10], PR[5]=LPOS
              :
              :
30: LBL[10]
```

Explanation of the execution example

When SDI[3] is turned on during execution of the 11th line, the current position is stored in a form of Cartesian coordinates.

When SDI[3] is not turned on during execution of the 11th line, a branch to LBL[10] is made after the execution of the 11th line ends. In this case, no position data is stored in PR[5].

4. Limitations and notes

<1> Position read error

As the programmed operation speed is slower, the position read accuracy under skip conditions becomes higher. (As a guideline, an error of about 1.5 mm is generated for 100 mm/sec. The error is proportional to the speed.)

OFFSET instruction

Offset,PR[2] (UFRAME [1])

J P[1] 50% FINE Offset

```
        JOINT  30 %
5 Offset
6 Offset,PR[  ]
7 Incremental
8 ---next page---
```

The OFFSET instruction alters positional information programmed at the destination position by the offset amount specified by a position register,and moves the robot to the altered position.

The offset condition is specified by the OFFSET CONDITION instruction.

The OFFSET CONDITION instruction specifies the offset amount used by the OFFSET instruction in advance. The OFFSET CONDITION instruction has to be specified before the OFFSET instruction is executed. The specified offset condition is available until the program is finished or the next OFFSET CONDITION instruction is executed:

As for the offset condition, the following elements should be specified:

- The position register specifies the shifting direction and the shift amount.
- When the positional information is expressed in the joint frame,the shift amount of each axis is applied.
- When the positional information is expressed in the Cartesian coordinate system,the user frame by which the offset condition is decided should be specified.(See Section 4.12, ”FRAME INSTRUCTION) When it is not specified,the user frame (UF) being selected now is used.(See Section 4.10, ”OFFSET CONDITION INSTRUCTION”)

> **CAUTION**
>
> If teaching is made by joint coordinates, changing the user coordinate system does not affect the position variables and position registers. If teaching is performed in cartesian coordinates, and the user coordinate system input option is not used, the position variable is not influenced by the user coordinate system. In other cases, both the position variable and position register are influenced by the user coordinate system.

The setting values of the tool frame number (UT) and the configuration (CONF:) are ignored.

When you teach or edit the positional information of the motion instruction with the OFFSET option,you teach the position minus the offset amount.

When you teach or edit the positional information of the motion instruction with the OFFSET option,the following prompt message is displayed.

- Subtract offset data from current pos?
  - Yes The positional information subtracted the offset data is taught.
  - No The positional information is directly taught.
- Enter PR index of offset data :
  - Enter the number of the position register specified in the OFFSET CONDITION instruction.
- Enter uframe no of offset data :
  - Enter the number of the user frame which is used when the offset amount is subtracted.

When the positional information is manually edited with the numerical keys,you can not teach the positional information minus the offset amount.

Moreover, even if the position teaching by which the amount of the subtracted correction is effective, the current position will be taken in the following cases.

- The specified position register is non–initialization.
- ”Ignore Offset command” is set to ENABLED. (See Section 3.15 ”Setting the general items”)

When ”Ignore Offset command” is set to ENABLED, the current position is directly taught as the positional information (The prompt message is not displayed) and the robot stops at the teaching position even if the OFFSET instruction is executed.

When the offset amount is changed after the program is paused while the OFFSET instruction is in progress,this change is reflected to the motion after the program is resumed. But,when you change the number of a position register in the OFFSET CONDITION instruction,this change is not reflected in the motion.

The robot moves to the offset position at the backward execution.(See Section 6.3.2,”Step test”)

This is the same as the following explanation for the direct offset condition instruction.

Direct offset condition instruction
J P[1] 50% FINE Offset,PR[2]

```
          JOINT  30 %
 5 Offset
 6 Offset,PR[  ]
 7 Incremental
 8 ---next page---
```

The direct offset condition instruction alters positional information by the offset amount directly specified in the position register without using the offset condition specified in the OFFSET CONDITION instruction. The reference frame is specified by the number of the user frame currently selected.

> **CAUTION**
> If teaching is made by joint coordinates, changing the user coordinate system does not affect the position variables and position registers. If teaching is performed in cartesian coordinates, and the user coordinate system input option is not used, the position variable is not influenced by the user coordinate system. In other cases, both the position variable and position register are influenced by the user coordinate system.

When you change or edit the motion instruction with the direct offset condition option,you can teach the positional information minus the offset amount.

When you teach or edit the motion instruction with the direct offset condition option,the following prompt message is displayed:

- Subtract offset data from current pos?
  - Yes Subtract the offset data from the taught position.
  - No The positional information from the directly taught position.

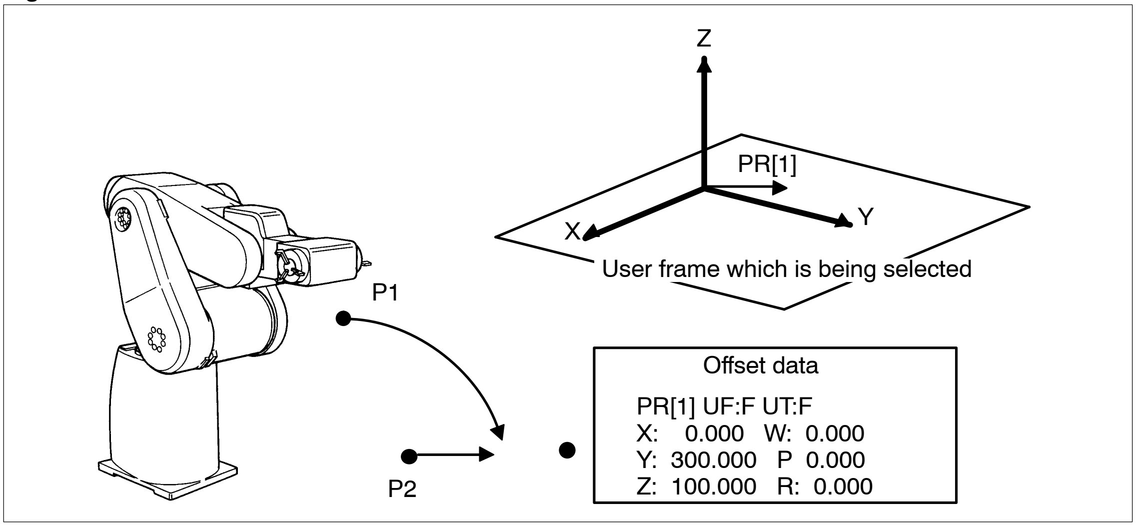
When the positional information is manually edited with the numerical keys,you can not teach the positional information minus the offset amount.

Moreover, even if the position teaching by which the subtracted offset amount is effective, the current position will be taught as it is in the following cases:

- The specified position register is non–initialized.
- The position register number used by direct offset condition instruction is non–initialized.
- ”Ignore Offset command” is set to ENABLED.(See Section 3.15 ”Setting the general items”)

When ”Ignore Offset command” is set to ENABLED,the current position is directly taught as the positional information (The prompt message is not displayed) and the robot stops at the teaching point even if the offset instruction is executed.

Figure 4–20. Offset Instruction



Example 1

```
1: OFFSET CONDITION PR[1]
2: J  P[1] 100% FINE
3: L  P[2] 500mm/sec FINE Offset
```

Example 2

```
1: J  P[1] 100% FINE
2: L  P[2] 500mm/sec FINE Offset, PR[1]
```

Tool offset instruction

TOOL_OFFSET_CONDITION PR[2] (UTOOL[1])

J P[1] 50% FINE Tool_offset

```
         JOINT  30 %
5 Tool_Offset
6 Tool_Offset,PR[
7 Incremental
8 ---next page---
```

A tool offset instruction moves the robot to the position shifted from the target position, recorded in the position data, by the offset specified in the tool offset conditions. The condition when the offset is applied is specified by a tool offset condition instruction.

A tool offset condition instruction specifies the offset condition used in a tool offset instruction. Execute a tool offset condition instruction before executing the corresponding tool offset instruction. Once the tool offset condition has been specified, it remains effective until the program terminates or the next tool offset condition instruction is executed.

Note the following when specifying tool offset conditions.

- The position register specifies the direction in which the target position shifts, as well as the amount of the shift.
- The tool coordinate system is used for specifying offset conditions.
- When the number of a tool coordinate system is omitted, the currently selected tool coordinate system is used.

When a motion statement which includes a tool offset instruction is taught or a certain position is modified, the position to which the offset is not to be applied can be taught.

When a motion statement which includes a tool offset instruction is taught or a certain position is modified, the system prompts the operator to respond to enter data in response to the following messages.

- Subtract tool offset data?
  - Pressing the YES soft key subtracts the tool offset from the position data and the robot is taught the new position.
  - Pressing the NO soft key stores the current position as the position data.
- Enter PR index of tool offset data?
  - Specify the position–register number specified by the tool offset condition instruction.
- Enter tool no. of tool offset data?
  - Specify the number of the tool coordinate system in which the offset is to be specified.

When the position data is manually modified with the numeric keys, the position is taught without subtracting the offset.

Even when teaching of the position from which the offset is subtracted is enabled, the current position is stored in the following cases.

- When the specified position register has not yet been initialized
- When ”Ignore Tool–offset” is set to ENABLED. (See Section 3.15 ”Setting the general items”.)

When ”Ignore Tool–offset” is set to ENABLED, the current position is taught as position data (no prompt messages are output) and the robot is moved to the taught position, even if a tool offset instruction is executed.

When the robot is temporarily stopped during the execution of a tool offset instruction and the shift distance is modified, the modified distance is used in the resumed movement. When a position register number specified by a tool offset condition instruction is modified, the modified number is not used.

In backward execution (See Section 6.3.2, ”Step test”), the robot is moved to the position to which the offset has been applied. This also applies to the direct tool offset instruction, described next.

Direct tool offset instruction

J P[1] 50% FINE Tool_Offset, PR[2]

```
        JOINT  30 %
5 Tool_Offset
6 Tool_Offset,PR[
7 Incremental
8 ---next page---
```

The robot moves according to the offset stored in the specified position register, ignoring the tool offset conditions specified by the tool offset condition instruction. The currently selected tool coordinate system is used.

When a motion statement which includes a direct tool offset instruction is taught or a certain position is modified, the position to which the offset is not to be applied can be taught.

When a motion statement which includes a direct tool offset instruction is taught or a certain position is modified, the system prompts the operator to enter data in response to the following messages.

- Subtract tool offset data?
  - Pressing the YES soft key subtracts the tool offset from the position data and the robot is taught the new position.
  - Pressing the NO soft key stores the current position as position data.

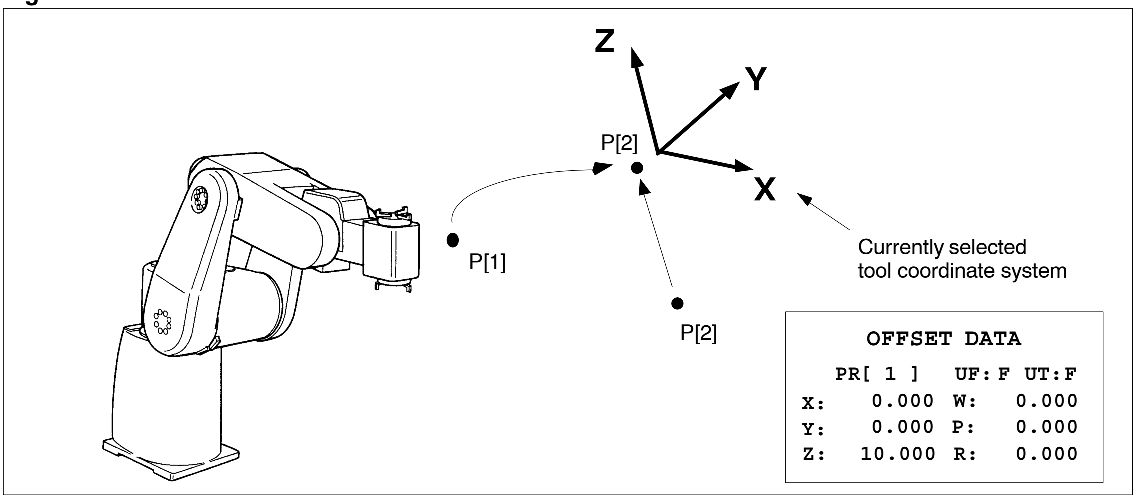
When the position data is manually modified with the numeric keys, the position is taught without subtracting the offset.

When teaching of the position from which the offset is subtracted is enabled, the current position is stored in the following cases.

- When the specified position register has not yet been initialized
- When the direct tool offset instruction has not specified the number of a position register
- When ”Ignore Tool–offset” is set to ENABLED. (See Section 3.15 ”Setting the general items”.)

When ”Ignore Tool–offset” is set to ENABLED, the current position is taught as position data (no prompt messages are output) and the robot is moved to the taught position even if a tool offset instruction is executed.

Figure 4–21. Tool Offset Instruction



Example 1

```
1: TOOL_OFFSET CONDITION PR[1]
2: J P[1] 100% FINE
3: L P[2] 500mm/sec FINE Tool_Offset
```

Example 2

```
1: J P[1] 100% FINE
2: L P[2] 500mm/sec FINE Tool_Offset, PR[1]
```

Incremental instruction

J P[1] 50% FINE INC

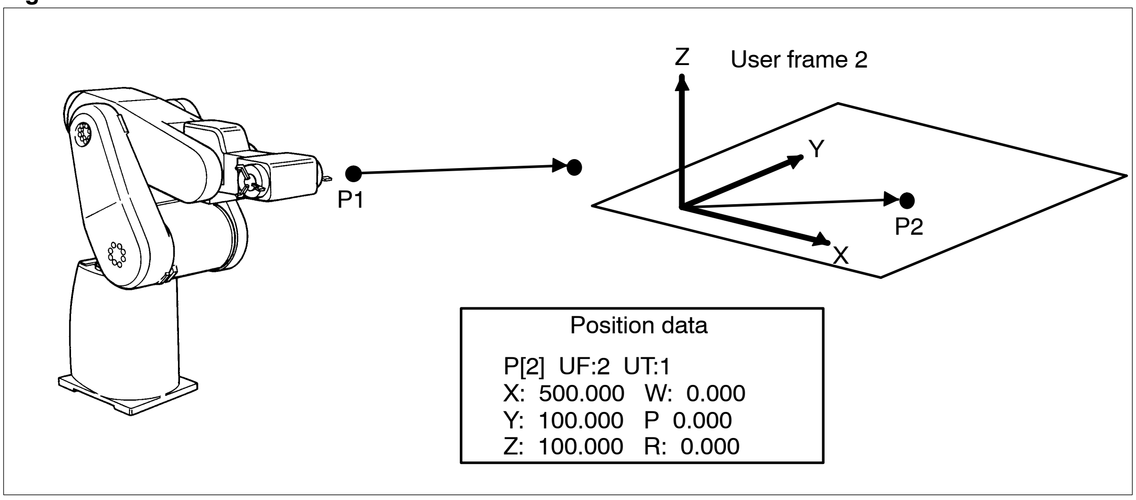
```
         JOINT  30 %
5 Offset
6 Offset,PR[  ]
7 Incremental
8 ---next page---
```

The incremental instruction uses the positional data in the motion instruction as the incremental amount from the current position,and causes the robot to move to the destination position that the incremental amount is added to the current position. This means that the incremental motion amount from the current position is recorded in the positional data in the motion instruction.

The incremental condition is specified by the following elements:

- When the positional data is joint frame value,the incremental amount of each axis is applied.
- When the positional variable (P[]) is used as the positional data,the reference user frame is specified by the number of the user frame which is specified in the positional data. However, the frame is verified.(For the cartesian coordinate system reference, See Section 4.3.2)
- When the position register is used as the position data, the reference frame is the current user frame.
- When the INC instruction is used with the OFFSET instruction,the type of the positional data in the motion instruction should be corresponding to the type of the positional register for the offset. In this case,the offset amount is used as the offset amount of the specified incremental amount.

Figure 4–22. Incremental instruction



Example

```
1: J  P[1] 100% FINE
2: L  P[2] 500mm/sec FINE INC
```

Note the following when teaching the incremental instruction (See Section 5.3.4, ”Teaching an supplementary motion instruction”):

- Adding the INC option causes the positional data to be non–initialized.
- When the motion instruction with the INC option is taught,the positional data is set to be non–teaching.
- Editing the position in the motion instruction with the INC option removes the INC option automatically.

When the motion instruction with the INC option is paused and the position data is changed,that changes is not immediately reflected. To move the robot to the changed position,resume the program from the just previous motion instruction.

Simultaneous EV instruction

J P[1] 50% FINE EV 50%

```
Motion modify
  1 Independent EV
  2 Simultaneous EV
  3
  4
PROGRAM1
```

The additional axis speed instruction (synchronous) moves the robot in sync with the additional axis.

When this instruction is used, the robot and additional axis operations are synchronized as follows:

- If the robot operation time is longer than the additional axis operation time, the additional axis operation is synchronized with the robot operation.
- If the additional axis operation time is longer than the robot operation time, the robot operation is synchronized with the additional axis operation.

The extended axis speed is specified as a ratio (1% to 100%) to the maximum travel speed of the extended axis.

Independent EV instruction (Ind.EV i%)

J P[1] 50% FINE Ind.EV 50%

```
Motion modify
  1 Independent EV
  2 Simultaneous EV
  3
  4
PROGRAM1
```

The additional axis speed instruction (asynchronous) moves the robot synchronously with the additional axis.

When this instruction is used, the robot and the additional axis start moving at the same time, but stop at different times because they are not synchronized.

The extended axis speed is specified as a ratio (1% to 100%) to the maximum travel speed of the extended axis.

If a motion statement is not accompanied with either extended axis speed instruction, the extended axis moves in synchronization with the speed of the robot.

Path instruction

J P[1] 50% Cnt10 PTH

```
Motion modify
  1 Independent EV
  2 Simultaneous EV
  3 PTH
  4
PROGRAM1
```

This function is designed to improve the performance of continuous motion (the termination type is Cnt1 to Cnt100) when the robot moves through a short distance.

In a motion where the robot moves through a short distance, the robot speed cannot be increased to the speed specified by a motion statement. For this reason, in an operation statement for which the positioning format is "FINE," operation planning for such an operation must be based on the "attainable speed," the speed that the robot can actually attain, rather than the specified speed. (Motion planning entails calculating the path along which the robot will travel, before actual operation.)

By using this instruction, operation planning is performed using the "attainable speed" in a CNT operation.

The use of this function enables the following effects in normal operation:

- Improvement in cycle time
- Improvement in path accuracy

This function is more effective as the movement distance is shorter and the Cnt value is smaller (the value n in Cntn is smaller).

When using this function, note the following:

In the following cases, use of the PTH instruction may actually incur a longer cycle time:

Before using this function, therefore, confirm its effect.

- A large Cn t value is specified in a motion statement.
- A motion statement causes the robot to move through a long distance.
- Successive Cnt motion statements appear.

CAUTION

Some motion instructions that use the PTH switch might cause jerky motion or vibration. If the motion is attached to PTH has a vibration, delete the PTH motion option.

4.3.6 High–precision locus function (option)

The high–precision locus function improves the operating performance of the robot in linear and arc operations having a CNT or corner distance specification. Linear positioning operations and joint operation are the same regardless of whether this function is provided.

- Constant locus

  By using the high–precision focus function, the robot can move along a constant locus without using the override. For a program taught and tested with a low override value, the robot moves along the same locus even if the override is at 100%.

- Restart operation after a temporary/emergency stop

  When a restart operation is performed after a temporary or emergency stop, this function causes the robot to move along the same locus as that along which it would move if the stop was not performed. Although the robot moves along the same locus, the orientation may differ slightly. For the restart operation function, the following system variable must be specified:

  – $MH_ORGRSM.$RET_PTH_ENA=TRUE (handling tool)

- Locus precision improvement

  This function improves locus precision for linear and arc operations.

- Constant speed

  This function can maintain the specified speed when the robot moves around a corner. If the specified speed cannot be maintained due to the operating performance of the robot mechanism, the speed decreases automatically. If the speed around the corner decreases, adjustment is possible using the corner positioning format and the corner speed ratio add instruction.

  The corner positioning format and the corner speed ratio add instruction can be edited from the teach pendant as part of a normal program.

- Corner distance specification

  This function allows the user to specify the distance between a specified point around a corner and the actual pass point. To specify the distance, use a corner distance specification, not CNT. If CNT 100 is specified, the minimum corner distance that allows the robot to move at a constant speed is assumed.

NOTE Control of the locus/speed for an operation for which CNT format is specified differs depending on whether the high–precision locus function is added. A positioning teaching operation does not differ depending on whether the high–precision locus function is added.

Corner distance specification format L P[1] 100mm/sec CDy.

To adjust the distance between a specified point and the path during the execution of an operation instruction, use the corner distance specification format. When the corner distance specification format is used, the corner distance must be specified.

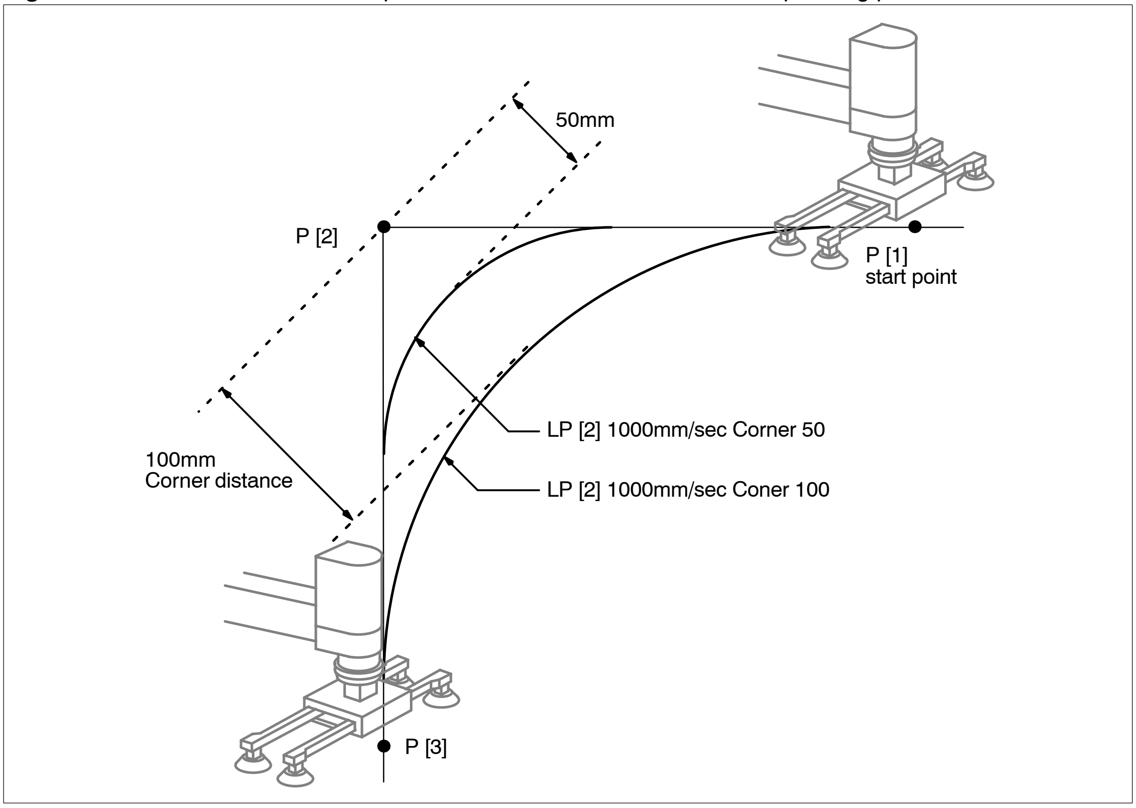
A corner distance is the distance between a specified point and the path during operation.

Note the following when specifying a corner distance:

- A corner distance is specified in mm units.
- A corner distance must fall within the range of 0 mm to 1000 mm.
- With a shorter corner distance, the robot passes nearer the specified point.
- With a longer corner distance, the robot passes farther from the specified point.
- A corner can never be more than half the distance between the preceding and succeeding specified points. When an operation subject to this control is performed, the corner distance is restricted even if the corner distance in the program is increased. When an operation subject to the restriction is performed, the robot decelerates around the corner. This restriction affects both the CNT positioning format and corner distance positioning format.

Fig.4–23 shows the relationship between a corner distance and an operating path.

Figure 4–23. Shows the relationship between a corner distance and an operating path



When a corner distance is specified, an attempt is made to maintain the specified speed around a corner while remaining within the operating capabilities of the robot. If the specified speed cannot be maintained, the robot automatically decelerates. To adjust the corner speed, use the corner speed ratio add instruction.

Corner speed ratio add instruction

When the high–precision locus function is enabled, an attempt is made to maintain the specified speed while remaining within the operating capabilities of the robot. If the specified speed cannot be maintained, the robot automatically decelerates. If the speed decreases around a corner and the speed around the corner needs adjustment, use the corner speed ratio add instruction.

Specifying the corner speed ratio add instruction

- A ratio of 100 causes the robot to operate in exactly the same way as when the corner speed ratio add instruction is not specified.
- If the corner speed ratio add instruction is specified with a parameter greater than 100, the speed around the corner becomes faster than that when the corner speed ratio add instruction is not specified. The speed cannot, however, exceed the specified speed.
- If the corner speed ratio add instruction is specified with a parameter smaller than 100, the speed around the corner becomes slower than that when the corner speed ratio add instruction is not specified.

CAUTION

When the corner speed ratio add instruction is used with a parameter greater than 100, vibration may occur during the operation of the robot. If this occurs, either delete the corner speed ratio add instruction or specify a ratio of 100.

Restrictions

Some instructions may cause the robot to decelerate regardless of the positioning format for an operation instruction. For such instructions, corner distance positioning and the corner speed ratio add instruction are ignored and positioning equivalent to CNT 0 is performed.

Instructions that cause the robot to decelerate if Lock position register Instruction/Unlook position register instruction is not used:

- Position register instructions
  PR[], PR[]INC
- Position compensation add instructions
  OFFSET, TOOL_OFFSET

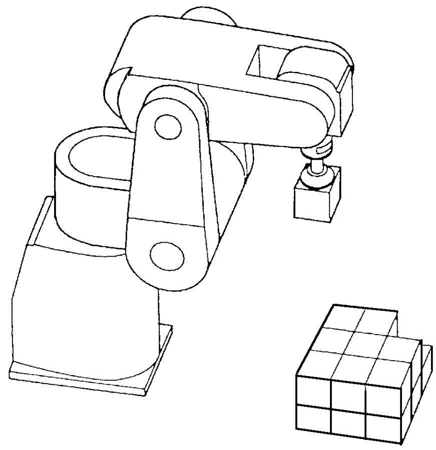
Instructions that always cause the robot to decelerate:

- Coordinate system instructions
  UFRAME_NUM, UFRAME, UTOOL_NUM, UTOOL
- – Branch instructions
  IF SELECT CALL (if the calling and called programs use different motion groups)
- Wait instruction
  WAIT
- Another instruction
  Parameter instruction
- Program control instructions
  ABORT
- Macros
  (if the calling program and the called macro use different motion groups)
- Skip condition instructions
- Tracking instructions
- Sensor instructions
  RCV
- Palletizing instructions
  PALLETIZIHG_B, PL[]

4.4 Palletizing Instructions

Palletizing is a function for orderly stacking of workpieces by only teaching several representative points. (See Section 10.1.)

Figure 4–24. Palletizing



The following palletizing instructions are available:

- Palletizing instruction PALLETIZING–B, BX, E, EX
- Palletizing motion instruction L PAL_i [BTM] 100 % FINE
- Palletizing end instruction PALLETIZING–END–_i

4.4.1 Palletizing instruction

Based on the value held in the palletizing register, the palletizing instruction calculates the position of the current stack point from a stacking pattern and the position of the current path from a path pattern. It then writes the found values into the position data of a palletizing motion instruction.

Figure 4–25. Palletizing Instruction

```
PALLETIZING–[pattern]_i
     B, BX, E, EX ——┘    └—— Palletizing number(1 to 16)
```

Palletizing is divided into four palletizing patterns according to the piling and path patterns (See Section 10.2).

Palletizing patterns

For pallet instructions, palletizing is divided into the following palletizing patterns.

Table 4–1. Palletizing Patterns

Pattern	Allowable palletizing
B	Simple piling pattern and single path pattern
BX	Simple piling pattern and multiple path patterns
E	Complex piling pattern and single path pattern
EX	Complex piling pattern and multiple path patterns

4.4.2 Palletizing motion instruction

The palletizing motion instruction is a motion instruction that uses three path points – an approach point, stack point, and retraction point – as position data. This instruction is dedicated to palletizing. Each palletizing instruction rewrites such position data.

Figure 4–26. Palletizing Motion Instruction

J PAL_i [Path point] 100% FINE

Palletizing number (1 to 16)

Types of path points

A_n	:	Approach point n = 1 to 8
BTM	:	Stack point
R_n	:	Retraction point n = 1 to 8

4.4.3 Palletizing end instruction

The palletizing end instruction increments or decrements the palletizing register.

Figure 4–27. Palletizing End Instruction

PALLETIZING–END_i

Palletizing number(1 to 16)

Example

```
1:    PALLETIZING-B_3
2: L  PAL_3[ A_1 ] 100mm/sec CNT10
3: L  PAL_3[ BTM ] 50mm/sec FINE
4:    HAND1 OPEN
5: L  PAL_3[ R_1 ] 100mm/sec CNT10
6:    PALLETIZING-END_3
```

4.5 Register Instructions

The register instructions perform arithmetic operations on registers. The following register instructions are available:

```
Instruction                         JOINT 30%
1 Registers                 5 JMP/LBL
2 I/O                       6 CALL
3 IF/SELECT                 7 Palletizing
4 WAIT                      8 ---next page---
PROGRAM
```

- Register instructions
- Position register instructions
- Position register axis instructions
- Polletizing register instructions

In register operations, polynomial operations such as those shown below are possible:

Example 1: R[2]=R[3]–R[4]+R[5]–R[6]
2: R[10]=R[2]*[100/R[6]

The following restrictions are imposed:

- Up to five operators can be written on a single line.

Example 1: R[2]=R[3]+R[4]+R[5]+R[6]+R[7]+R[8]
Up to five operators

- The ”+” and ”–” operators can be mixed on a single line. So can the ”*” and ”/” operators. ”+” and ”–” cannot, however, be mixed with ”*” and ”/”.

4.5.1 Register instructions

A register instruction performs an arithmetic operation on registers. A register is a variable for holding an integer or a decimal fraction. (For registers, See Section 7.3.) Two hundred registers are provided.

R[i] = (value)

The instruction, R[i] = (value), loads a value into a specified register.

Figure 4–28. Instruction R[i] = (value)

R [i] = (value)

Register number(1 to 32)

- Constant
- R [i] : Value of R[i]
- PR [i, j] : Value of position register element [i, j]
- GI [i] : Group input signal
- GO [i] : Group output signal
- AI [i] : Analog input signal
- AO [i] : Analog output signal
- SDI [i] : System digital input signal
- SDO [i] : System digital output signal
- RDI [i] : Robot digital input signal
- RDO [i] : Robot digital output signal
- SI [i] Operation panel input signal
- SO [i] : Operation panel output signal
- UI [i] : Peripheral device input signal
- UO [i] : Peripheral device output signal
- Timer [i] : Value of program timer [i]
- Timer overflow [i] : Overflow flag of program timer [i]
  0: The timer has not overflowed.
  1: The timer has overflowed.

* The timer overflow flag is cleared with the timer [i] = reset instruction.

Example
```
1: R[1] = RDI[3]
2: R[R[4]] = AI[R[1]]
```

R[i] = (value) + (value)

The instruction, R[i] = (value) + (value), loads the sum of two values into a specified register.

R[i] = (value) – (value)

The instruction, R[i] = (value) – (value), loads the difference between two values into a specified register.

R[i] = (value) * (value)

The instruction, R[i] = (value) * (value), loads the product of two values into a specified register.

R[i] = (value) / (value)

The instruction, R[i] = (value) / (value), loads the quotient of two values into a specified register.

R[i] = (value) MOD (value)

The instruction, R[i] = (value) MOD (value), loads the remainder (value after decimal point) of the quotient of two values into a specified register.

R[i] = (value) DIV (value)

The instruction, R[i] = (value) DIV (value), loads the integer of the quotient of two values into a specified register.
R [i] = (x – (x MOD y)) / y

Figure 4–29. Arithmetic Register Instruction

Register number(1 to 32)

R [i] = (value) (operator) (value)

Operator:
- +
- –
- *
- /
- MOD
- DIV

Value:
- AR[i]
- Constant
- R [i] : Value of R[i]
- PR [i, j] : Value of position register element [i, j]
- GI [i] : Group input signal
- GO [i] : Group output signal
- AI [i] : Analog input signal
- AO [i] : Analog output signal
- SDI [i] : System digital input signal
- SDO [i] : System digital output signal
- RDI [i] : Robot digital input signal
- RDO [i] : Robot digital output signal
- SI [i] Operation panel input signal
- SO [i] : Operation panel output signal
- UI [i] : Peripheral device input signal
- UO [i] : Peripheral device output signal
- Timer [i] : Value of program timer [i]
- Timer overflow [i] : Overflow flag of program timer [i]
  0: The timer has not overflowed.
  1: The timer has overflowed.

* The timer overflow flag is cleared with the timer [i] = reset instruction.

Example

```
3: R[3:flag] = SDI[4]+PR[ 1, 2 ]
4: R[ R[4] ] = R[1]+1
```

4.5.2 Position register instructions

A position register instruction performs an arithmetic operation on position registers. A position register instruction can load position data, the sum of two values, or the difference of two values, into a specified position register. A position register instruction uses the same format as a register instruction.

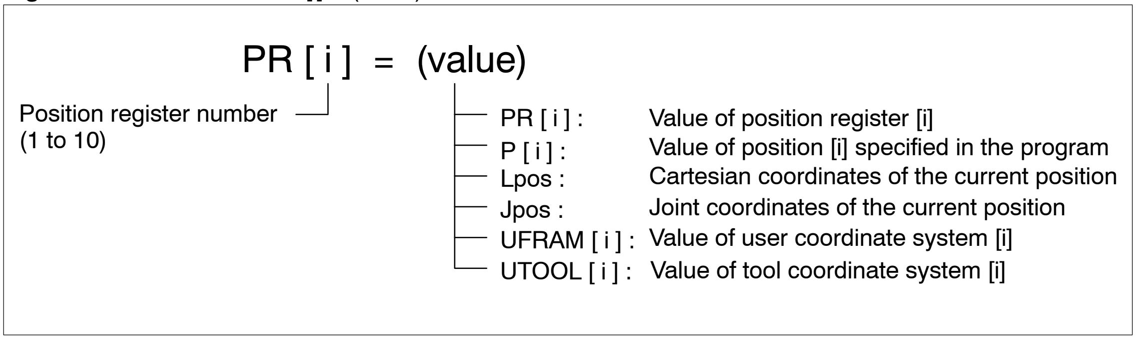
A position register is a variable for holding position data (x,y,z,w,p,r). (For position registers, See Section 7.4.) Ten position registers are provided.

NOTE Before using position register instructions, lock position registers by specifying LOCK PREG. When position register instructions are used with the position registers unlocked, operation may become tight. For the LOCK PREG instruction, see Section 9.6, "POSITION REGISTER LOOK–AHEAD EXECUTION FUNCTION."

PR[i] = (value)

The instruction, PR[i] = (value), loads position data into a specified position register.

Figure 4–30. Instruction PR[i] = (value)



Example

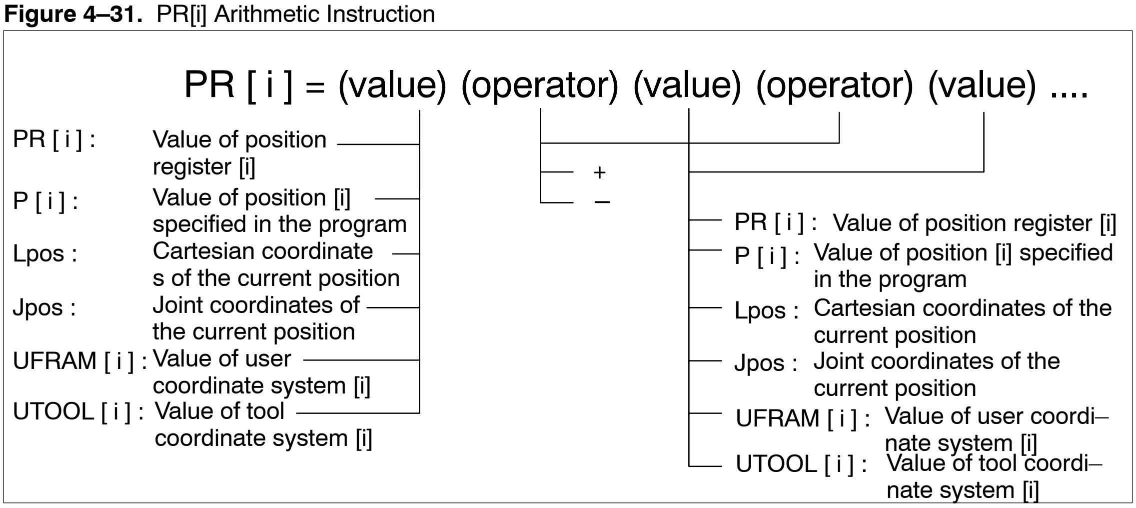
```
1: PR[1] = Lpos
2: PR[ R[4] ] = UFRAME[ R[1] ]
3: PR[9] = UTOOL[1]
```

PR[i] = (value) + (value)

The instruction, PR[i] = (value) + (value), loads the sum of two values into a specified register.

The instruction, PR[i] = (value) – (value), loads the difference of two values into a specified register.

Figure 4–31. PR[i] Arithmetic Instruction



Example

```
4: PR[3] = PR[3]+Lpos
5: PR[4] = PR[ R[1] ]
```

4.5.3 Position register axis instructions

A position register axis instruction performs an arithmetic operation on position register elements. i of PR[i,j] represents a position register number, and j of PR[i,j] represents a position register element number. The position register axis instructions can load the value of one position data element, or the sum, difference, product, or quotient of two values into a specified position register element. A position register axis instruction uses the same format as a register instruction.

Figure 4–32. Format of PR[i,j]

PR [i, j]

Position register number (1 to 10)

Position register element number

Cartesian coordinate system:
- 1 = X
- 2 = Y
- 3 = Z
- 4 = W
- 5 = P
- 6 = R

Joint coordinate system:
- 1 = J1
- 2 = J2
- 3 = J3
- 4 = J4
- 5 = J5
- 6 = J6
- n = Jn

PR[i,j] = (value)

The instruction, PR[i,j] = (value), loads the value of a position data element into a position register element.

Figure 4–33. Instruction PR[i,j] = (value)

PR [i, j] = (value)

Position register number (1 to 10)

- AR[i]
- Constant
- R [i] : Register [i]
- PR [i, j] : Position register element [i, j]
- GI [i] : Group input signal
- GO [i] : Group output signal
- AI [i] : Analog input signal
- AO [i] : Analog output signal
- SDI [i] : System digital input signal
- SDO [i] : System digital output signal
- RDI [i] : Robot digital input signal
- RDO [i] : Robot digital output signal
- SI [i] Operation panel input signal
- SO [i] : Operation panel output signal
- UI [i] : Peripheral device input signal
- UO [i] : Peripheral device output signal
- Timer [i] : Value of program timer [i]
- Timer overflow [i] : Overflow flag of program timer [i]
  0: The timer has not overflowed.
  1: The timer has overflowed.

* The timer overflow flag is cleared with the timer [i] = reset instruction.

Example

```
1: PR[ 1, 2 ] = R[3]
2: PR[ 4, 3 ] = 324.5
```

PR[i] = (value) + (value)

The instruction, PR[i,j] = (value) + (value), loads the sum of two values into a specified position register element.

PR[i] = (value) – (value)

The instruction, PR[i,j] = (value) – (value), loads the difference of two values into a specified position register element.

PR[i] = (value) * (value)

The instruction, PR[i,j] = (value) * (value), loads the product of two values into a specified position register element.

PR[i] = (value) / (value)

The instruction, PR[i,j] = (value) / (value), loads the quotient of two values into a specified position register element.

R[i] = (value) MOD (value)

The instruction, R[i] = (value) MOD (value), loads the remainder (value after decimal point) of the quotient of two values into a specified register.

R[i] = (value) DIV (value)

The instruction, R[i] = (value) DIV (value), loads the integer of the quotient of two values into a specified register.
R [i] = (x – (x MOD y)) / y

Figure 4–34. PR[i,j] Arithmetic Instruction

Position register number(1 to 10)

PR [i] = (value) (operator) (value) (operator) ....

Operator:
- +
- –
- *
- /
- MOD
- DIV

Value:
- AR[i]
- Constant
- R [i] : Register [i]
- PR [i, j] : Position register element [i, j]
- GI [i] : Group input signal
- GO [i] : Group output signal
- AI [i] : Analog input signal
- AO [i] : Analog output signal
- SDI [i] : System digital input signal
- SDO [i] : System digital output signal
- RDI [i] : Robot digital input signal
- RDO [i] : Robot digital output signal
- SI [i] Operation panel input signal
- SO [i] : Operation panel output signal
- UI [i] : Peripheral device input signal
- UO [i] : Peripheral device output signal
- Timer [i] : Value of program timer [i]
- Timer overflow [i] : Overflow flag of program timer [i]
  0: The timer has not overflowed.
  1: The timer has overflowed.

* The timer overflow flag is cleared with the timer [i] = reset instruction.

Example

```
1: PR[ 3, 5 ] = R[3]+DI[4]
2: PR[ 4, 3 ] = PR[ 1, 3 ]-3.528
```

4.5.4 Arithmetic palletizing register instructions

An arithmetic palletizing register instruction performs an arithmetic operation on palletizing registers. The arithmetic palletizing register instructions are load position data, the sum of two values, or the difference between two values. An arithmetic palletizing register instruction uses the same format as a register instruction.

A palletizing register holds palletizing register elements (j,k,l). A program can use up to 16 palletizing registers. (See Section 7.6.)

Palletizing register element

For palletizing register elements, elements to be loaded into a palletizing register, or elements to be operated on are specified. Three methods of element specification are available; – Direct specification: A numerical value is directly specified. – Indirect specification: The value of register [i] is specified. – Omitted: The asterisk (*) specifies that no modification is to be made.

Figure 4–35. Format of Palletizing Register Elements

[i , j , k]

Palletizing register elements

Direct specification	: Number of rows, columns, and layers(1 to 127)
Indirect specification	: Value of register [i]
Omited	: Asterisk (*) to specify that no modification is be made

PL[i] = (value)

The instruction, PL[i] = (value), loads (assigns) palletizing register elements into a specified palletizing register.

Figure 4–36. Instruction PL [i] = (value)

PL [i] = (value)

Palletizing register number (1 to 32)

PL [i] : Palletizing register [i]
[i , j , k] : Palletizing register elements

Example

```
1: PL[1] = PL[3]
2: PL[2] = [ 1, 2, 1 ]
3: PL[ R[3] ]= [ *, R[1], 1 ]
```

PL[i] = (value) (operator) (value)

The instruction, PL[i] = (value) (operator) (value), performs an arithmetic operation, then loads the result of the operation into a specified palletizing register.

Figure 4–37. Instruction PL [i] = (value) (operator) (value)

Palletizing register number(1 to 32)

PL [i] = (value) (operator) (value)

+
–

PL [i] : Palletizing register [i]
[i , j , k] : Palletizing register elements

Example

```
1: PL[1] = PL[3]+[ 1, 2, 1 ]
2: PL[2] = [ 1, 2, 1 ]+[ 1, R[1], * ]
```

4.6 I/O Instructions

The I/O (input/output signal) instructions are used to change the state of a signal output to peripheral devices and read the state of an input signal.

```
Instruction                     JOINT 30%
1 Registers           5 JMP/LBL
2 I/O                 6 CALL
3 IF/SELECT           7 Palletizing
4 WAIT                8 ---next page---
PROGRAM
```

- (System) digital I/O instruction
- Robot (digital) I/O instruction
- Analog I/O instruction
- Group I/O instruction

NOTE As for the I/O signal,the logical number need to be allocated to the physical number before using it.(For configuring I/O, See Section 3.1.)

4.6.1 Digital I/O instructions

The digital input signal (SDI) and digital output signal (SDO) are input/output signals that can be controlled by the user.

R[i] = SDI[i]

The instruction, SR[i] = DI[i] loads, the state of a digital input signal (on = 1/off = 0) into a specified register.

Figure 4–38. Instruction R[i] = SDI[i]

R [i] = SDI [i]

Register number (1 to 32)

Digital input signal number

Example
```
1: R[1] = SDI[1]
2: R[ R[3] ] = SDI[ R[4] ]
```

SDO[i] = ON/OFF

The instruction, SDO[i] = ON/OFF, turns on or off a specified digital output signal.

Figure 4–39. Instruction SDO[i] = ON/OFF

SDO [i] = (value)

Digital output signal number

ON : Turns on the digital output signal.
OFF: Turns off the digital output signal.

Example
```
3: SDO[1] = ON
4: SDO[ R[3] ] = OFF
```

SDO[i] = PULSE,[WIDTH]

The SDO[i] = PULSE, [TIME] instruction inverts the current status of a specified digital output for a specified duration. When no duration is specified, pulse output is executed for the duration specified with $DEFPULSE (0.1–second units).

Figure 4–40. Instructions SDO[i] = PULSE,(WIDTH)

SDO [i] = PULSE, (value)

Digital output signal number

Pulse width (sec) (0.1 to 25.5 sec)

Example

```
5: SDO[1] = PULSE
6: SDO[2] = PULSE, 0.2sec
7: SDO[ R[3] ] = PULSE, 1.2sec
```

SDO[i] = R[i]

The instruction, SDO[i]=R[i],turns on or off a specified digital output signal according to the value of a specified register. When the value of the specified register is 0, the digital output signal is turned off. When the value of the specified register is other than 0, the digital output signal is turned on.

Figure 4–41. Instruction SDO[i] = R[i]

SDO [i] = R [i]

Digital output signal number

Register number (1 to 32)

Example

```
7: SDO[1] = R[2]
8: SDO[ R[5] ] =R [ R[1] ]
```

4.6.2 Robot I/O instructions

The robot input signal (RDI) and robot output signal (RDO) are input/output signals that can be controlled by the user.

SR[i] = RI[i]

The instruction, R[i] = RI[i], loads the state of a robot input signal (on = 1/off = 0) into a specified register.

Figure 4–42. Instruction R[i] = RI[i]

R [i] = RDI [i]

Register number (1 to 32)

Robot input signal number

Example

```
1: R[1] = RDI[1]
2: R[ R[3] ] = RDI[ R[4] ]
```

RDO[i] = ON/OFF

The instruction, ROD[i] = ON/OFF, turns on or off a specified robot digital output signal.

Figure 4–43. Instruction RDO[i] = ON/OFF

RDO [i] = (value)

Robot output signal number

ON : Turns on the robot output signal.
OFF: Turns off the robot output signal.

Example

```
3: RDDO[1] = ON
4: RDDO[ R[3] ] = OFF
```

RDO[i] = PULSE,[WIDTH]

The RDO[i] = PULSE,[TIME] instruction inverts the current status of a specified digital output for a specified duration. When no duration is specified, pulse output is executed for the duration specified with $DEFPULSE (0.1–second units).

Figure 4–44. Instruction of RDO[i] = PULSE,[WIDTH]

RDO [i] = PULSE, [WIDTH]

Robot output signal number

Pulse width (sec) (0.1 to 25.5 sec)

Example

```
5: RDO[1] = PULSE
6: RDO[2] = PULSE, 0.2sec
7: RDO[ R[3] ] = PULSE, 1.2sec
```

RDO[i] = R[i]

The instruction, RDO[i] = R[i], turns on or off a specified robot output signal according to the value of a specified register. When the value of the specified register is 0, the robot output signal is turned off. When the value of the specified register is other than 0, the robot output signal is turned on.

Figure 4–45. Instruction RDO[i] = R[i]

RDO [i] = R [i]

Robot output signal number

Register number (1 to 32)

Example

```
7: RDO[1] = R[2]
8: RDO[ R[5] ] = R[ R[1] ]
```

4.6.3 Analog I/O instructions

Analog input (AI) and analog output (AO) signals indicate levels as a value on a continuum. Thus, the magnitude of a signal represents a temperature, voltage, or other data.

R[i] = AI[i]

The R[i] = AI[i] instruction stores the value of an analog input signal in a register.

Figure 4–46. R[i] = AI[i] Instruction

R [i] = AI [i]

Register number (1 to 32)

Analog input signal number

Example
```
1: R[1] = AI[2]
2: R[ R[3] ] = AI[ R[4] ]
```

AO[i] = (value)

The AO[i] = (value) instruction outputs a value as a specified analog output signal.

Figure 4–47. AO[i] = (value) Instruction

AO [i] = (value)

Analog output signal number

Value of analog output signal

Example
```
3: AO[1] = 0
4: AO[ R[3] ] = 3276
```

AO[i] = R[i]

The AO[i] = R[i] instruction outputs a register value as an analog output signal.

Figure 4–48. AO[i] = R[i] Instruction

AO [i] = R [i]

Analog output signal number

Register number (1 to 32)

Example
```
5: AO[1] = R[2]
6: AO[ R[5] ] = R[ R[1] ]
```

4.6.4 Group I/O instruction

R [i] = GI [i]

The signal of the group input(GI) and the group output(GO) is that some digital input/output signals are grouped and this is controlled by one instruction.

The instruction,R[i]=GI[i],converts the binary value of the specified group input signal to the decimal value and inputs it to the specified register.

Figure 4–49. Instruction R [i] = GI [i]

R [i] = GI [i]

Register number (1 to 32) — Group input signal number

Example
```
7: R[1] = GI[1]
8: R[ R[3] ] = GI[ R[4] ]
```

GO [i] = (value)

The GO[i]=(VALUE) instruction sends the binary equivalent of a value on the specified group output lines.

Figure 4–50. Instruction GO [i] = (value)

GO [i] = (value)

Group output signal number — Group output signal value

Example
```
3: GO[1] = 0
4: GO[ R[3] ] = 32767
```

GO [i] = R [i]

The GO[i]=R[i] instruction sends the binary equivalent of the contents of specified register on the specified group output lines.

Figure 4–51. Instruction GO [i] = R [i]

GO [i] = R [i]

Group output signal number — Register number (1 to 32)

Example
```
5: GO[1] = R[2]
6: GO[ R[5] ] = R[ R[1] ]
```

4.7 Branch Instructions

A branch instruction causes a branch from one line of a program to another. Four types of branch instructions are supported.

- Label instruction
- Program end instruction
- Unconditional branch instruction
- Conditional branch instruction

4.7.1 Label instruction

LABEL[i]

The label instruction (LBL[i]) is used to specify a program execution branch destination. A label is defined with a label definition instruction.

```
Instruction                          JOINT 30%
 1 Registers           5 JMP/LBL
 2 I/O                 6 CALL
 3 IF/SELECT           7 Palletizing
 4 WAIT                8 ---next page---
PROGRAM
```

A comment can be added to explain a label. Once a label is defined, it can be used for either an unconditional branch or conditional branch. It is not possible to specify the label number as the indirect addressing. To add a comment,move the cursor to the label number and press the ENTER key.

Figure 4–52. LBL[i] Instruction

LBL [i : Comment]

Label (1 to 32766)

A comment can consist of up to 16 characters including alphanumeric characters, asterisks (*), underlines (_), and at marks (@ etc...).

Example

```
1: LBL[1]
2: LBL[ R[3] ]
```

4.7.2 Program end instruction

END

The program end instruction indicates the end of a program. The execution of a program is terminated by this instruction. If a program is called from another main program, control is returned to the main program.

```
Instruction                          JOINT 30%
 1 Registers           5 JMP/LBL
 2 I/O                 6 CALL
 3 IF/SELECT           7 Palletizing
 4 WAIT                8 ---next page---
PROGRAM
```

Figure 4–53. Program End Instruction

END

4.7.3 Unconditional branch instructions

An unconditional branch instruction invariably causes a branch from one line to another in the same program. Two types of unconditional branch instructions are supported.

- Jump instruction: Causes a branch to a specified label or program.
- Program call instruction: Causes a branch to another program.

Jump instruction JMP LBL[i]

The JMP LBL[i] instruction transfers program control to a specified label.

Figure 4–54. JMP LBL[i] Instruction

```
JMP LBL [ i ]
          └── Label (1 to 32767)
```

Example

```
3: JMP LBL[2:hand open]
4: JMP LBL[ R[4] ]
```

Program call instruction CALL (program)

The CALL (program) instruction transfers program control to the first line of another program (subprogram) in order to execute it. When a program end instruction (END) in a called program is executed, control is returned to the instruction immediately after the program call instruction in the calling program (main program). To enter the calling program name,select it with the sub–menu automatically displayed or press F5,STRINGS to enter characters directly.

Figure 4–55. CALL (program) Instruction

```
CALL ( Program )
          └── Name of a program to be called
```

Example

```
5: CALL SUB1
6: CALL PROGRAM2
```

*) It is possible to set an argument for the program call instruction and use its value in a subprogram. See Section 4.7.5, ”Arguments” for details.

4.7.4 Conditional branch instructions

A conditional branch instruction causes a branch from one location in a program to another when some condition is satisfied. Two types of conditional branch instructions are available.

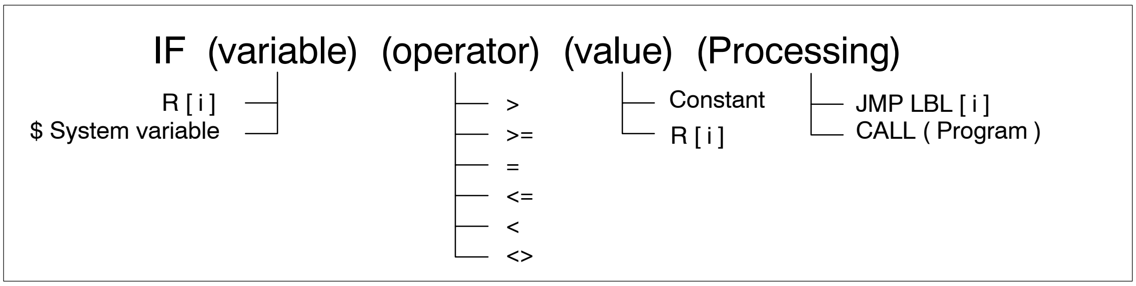
```
Instruction                      JOINT 30%
 1 Registers          5 JMP/LBL
 2 I/O                6 CALL
 3 IF/SELECT          7 Palletizing
 4 WAIT               8 ---next page---
PROGRAM
```

- Conditional compare instruction: Causes a branch to a specified label or program when some condition is satisfied. The register conditional compare instruction and I/O conditional compare instruction are available.
- Conditional select instruction: Causes a branch to a specified jump instruction or subprogram call instruction according to the value of a register.

Register conditional compare instruction IF R[i] (operator) (value) (processing)

A register conditional compare instruction compares the value stored in a register with another value and, when the compare condition is satisfied, executes processing.

Figure 4–56. Register Conditional Compare Instruction

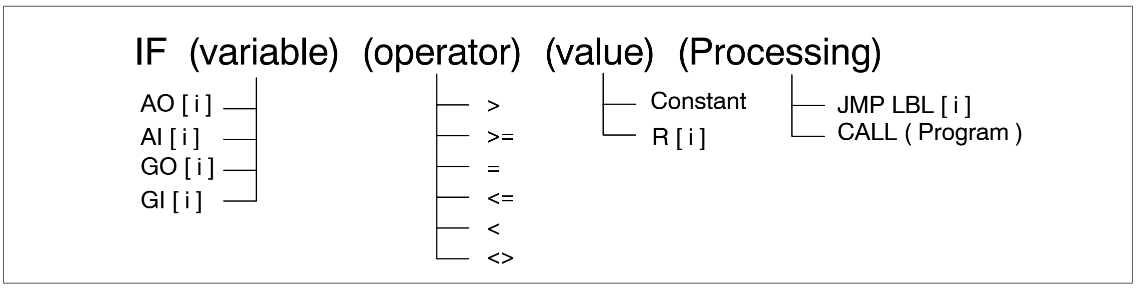


> **CAUTION**
> When the contents of a register is compared with the real value using the operator "=",the contents does not always correspond to the real value because of the rounding–off error of the contents. To compare with the real value,use the operator without on equal sign.

I/O conditional compare instruction IF (I/O) (operator) (value) (processing)

The I/O conditional compare instruction compares the value of an input/output signal with another value. When the comparison condition is satisfied, specified processing is executed.

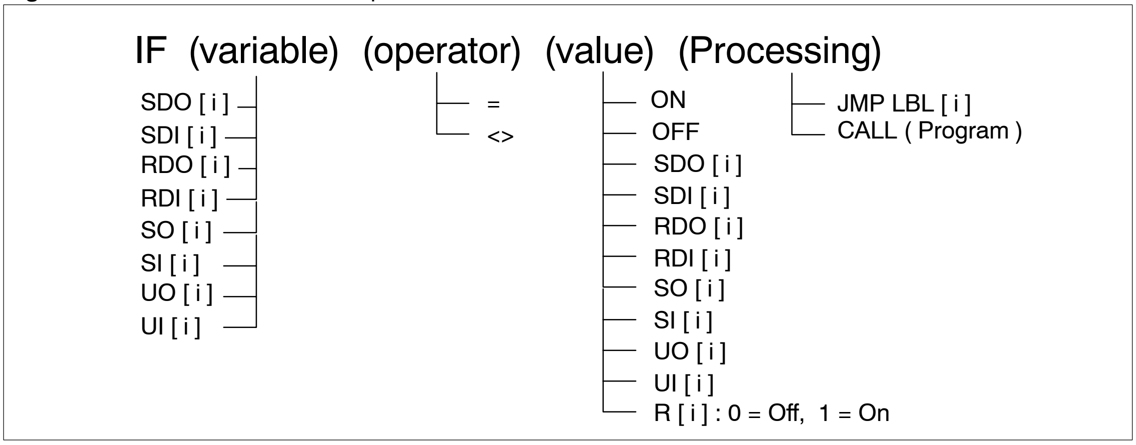
Figure 4–57. I/O Conditional Compare Instruction 1



Example

```
7: IF R[1] = R[2], JMP LBL[1]
8: IF AO[2] >= 3000, CALL SUBPRO1
9: IF GI[ R[2] ] = 100, CALL SUBPRO2
```

Figure 4–58. I/O Conditional Compare Instruction 2



Example
```
10: IF RO[2] <> OFF, JMP LBL[1]
11: IF DI[3] = ON, CALL SUB1
```

In a conditional branch instruction, multiple conditions can be specified on a single line in the condition statement, using the logical operators (”and” and ”or”). This simplifies the program structure, allowing the conditions to be evaluated efficiently.

Instruction format

- Logical product (and)
  IF <condition 1> and <condition 2> and <condition 3>, JMP LBL [3]
- Logical sum (or)
  F <condition 1> or <condition 2>, JMP LBL [3]

If the ”and” (logical product) and ”or” (logical sum) operators are used together, the logic becomes complex, impairing the readability of the program and case of editing. For this reason, this function prohibits the use of the logical operators ”and” and ”or” in combination.

If multiple ”and” (logical product) or ”or” (logical sum) operators are specified for an instruction on a single line, and one of the operators is changed from ”and” to ”or” or from ”or” to ”and,” all other ”and” or ”or” operators are changed accordingly, and the following message appears:

TPIF–062 AND operator was replaced to OR

TPIF–063 OR operator was replaced to AND

Up to five conditions can be combined with ”and” or ”or” operators on a single line.

Example IF <condition 1> and <condition 2> and <condition 3> and <condition 4> and <condition 5>, JMP LBL [3]

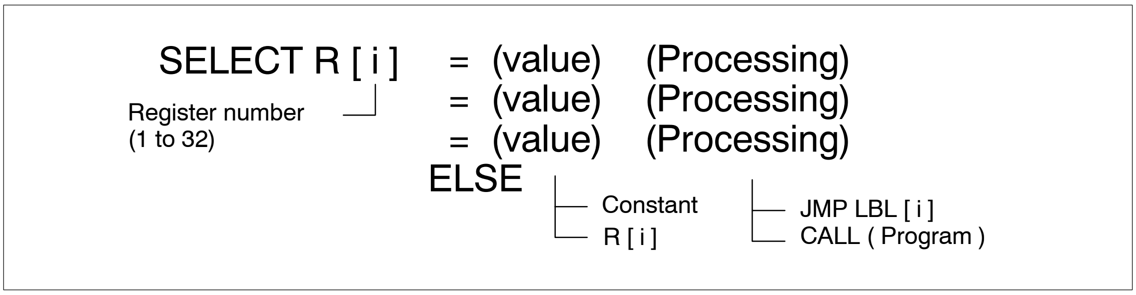
Conditional select instruction

SELECT R[i] = (value) (processing) = (value) (processing) = (value) (processing) ELSE (processing)

The conditional select instruction consists of several register compare instructions. The conditional select instruction compares the value of a register with one or more values, then selects a statement that satisfies the comparison condition.

- If the value of a specified register matches one value, the jump instruction or subprogram call instruction corresponding to the value is executed.
- If the value of a specified register does not match any of the values, the jump instruction or subprogram call instruction corresponding to ELSE is executed.

Figure 4–59. Conditional Select Instruction



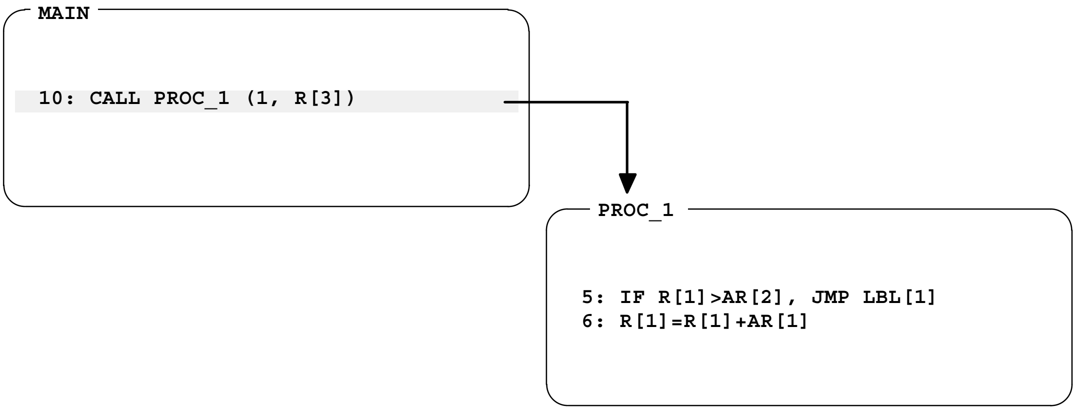
Example

```
11: SELECT R[1] = 1, JMP LBL[1]
12:     = 2, JMP LBL[2]
13:     = 3, JMP LBL[2]
14:     = 4, JMP LBL[2]
15:     ELSE, CALL SUB2
```

4.7.5 Arguments

By using ”arguments” and ”argument registers,” it is possible to transfer data between two programs only.

Example In this example, the main program MAIN calls the subprogram PROC_1 with two arguments. PROC_1 can use the values of the arguments with the argument registers. The first argument corresponds to AR[1] while the second argument corresponds to AR[2].



Arguments can be used in macro instructions in the same way.

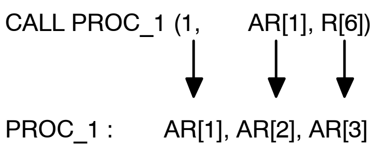
Argument types

The following arguments are supported.

Table 4–2. Argument types

Argument type	Example
Constant	1, 3.5
Character string	’Perch’
Argument register	AR[3]
Register	R [6]

*1 Available in KAREL programs only.
*2 Used as argument registers in subprograms.



Instructions for which arguments can be set

Table 4–3. Instructions for Which Arguments Can be Set

Instruction	Example
Program call instruction	CALL SUBPRG (1, R[3], AR[1])
Macro instruction	Vacuum hand open (2.5)

NOTE A program call used for branching with an instruction such as a conditional branch instruction cannot use arguments. This problem can be solved by programming as follows:

```
(Arguments cannot be set)   (Arguments can be set)
IF R[1]=3, CALL PROC_5 →   IF R[1]<>3, JMP LBL[1]
                            CALL PROC_5 (1, R[2])
                            LBL[1]
```

Instructions that can use argument registers

Table 4–4. Instructions That Can Use Argument Registers

Instruction	Example
Right side of an instruction and conditional expression having a register on the left side	R[1]=AR+R[2]+AR[4] IF R[1]=AR[1], JMP LBL[1]
Right side of the analog output (AO[]) and group output (GO[]) instructions	AO[1]=AR[2] GO[1]=AR[2]
Right side of a conditional expression having analog input/output (AI[]/AO[]) or group input/output (GI[]/GO[]) on the left side	IF AO[1]=AR[1], JMP LBL[1] WAIT GI[1]<>AR[2], TIMEOUT, LBL[1]
Right side of the user coordinate system selection instruction and the tool coordinate system selection instruction	UTOOL_NUM=AR[4]
Indirect index specification	R[AR[1]]=R[AR[2]] SDO[AR[1]]=ON
Argument of a program call instruction	CALL SUBPRG1 (AR[5])
Argument of a macro instruction	hand 3 open (AR[1])

Restrictions on arguments

The following restrictions are imposed on arguments:

- Up to 10 arguments can be set.
- An argument of character string type can be one to sixteen characters in length. (An argument with 0 characters is regarded as being uninitialized.)
- An indirect specification cannot be used for an already indirectly specified element of an index.
  - ○ R[AR[1]]
  - × R[R[AR[1]]]
- The value stored in an argument register cannot be changed in a subprogram.

Specifying arguments

When a program call instruction or macro instruction is specified, the cursor stops at the end of the line. If no arguments need be specified, press the ENTER key or ”→” or ”↓” key to move the cursor to the next line.

To display the argument selection submenu, press function key [CHOICE].

```
CALL PROC_1

        [CHOICE]
```

```
Parameter select                 JOINT 10%
  1 R[ ]                  5 <None>
  2 Constant              6 <Insert>
  3 String                7
  4 AR[ ]                 8
```

Specifying arguments of the constant type

To specify an argument of the constant type, press the [CHOICE] function key and select "1 Constant" from the submenu (see "Specifying arguments").

```
Parameter select
 1 R[ ]
 2 Constant
 3 String
 4 AR[]
```

1: CALL PROC_1 (Constant)

1: CALL PROC_1 (1)

Specifying arguments of character type

To specify an argument of character type, press function key [CHOICE] and select String from the submenu (see "Specifying arguments"). The character string type selection menu appears.

```
Parameter select
 1 R[ ]
 2 Constant
 3 String
 4 AR[ ]
```

```
String select                      JOINT 10%
  1 PARTS             5 DEV
  2 TOOL              6 PALT
  3 WORK              7 GRIP
  4 POS               8 --- next page ---
MAIN
                                         1/2
  1: CALL PROC_1 (1,   )
[End]

                         [CHOICE]  [STRING]
```

F4 F5

When a character string type is selected, the character string selection menu appears.

```
String select                      JOINT 10%
  1 Parts ITEM1       5 Parts ITEM5
  2 Parts ITEM2       6 Parts ITEM6
  3 Parts ITEM3       7 Parts ITEM7
  4 Parts ITEM4       8 --- next page ---
```

Select a character string from the menu. The character string is confirmed.

1: CALL PROC_1 ('Parts ITEM2') Select Parts ITEM2 from the menu

To enter a character string directly, press function key STRINGS from the character type selection menu or the character string selection menu.

```
                                   JOINT 10%
  1 Words
  2 Upper Cuse
  3 Lower Cuse
  4 Options
MAIN
                                         1/2
  1: CALL PROC_1 ('Tool12   ')
[End]

$  [      ]       [ PART ] [ POS ] [TOOL]
```

F1 F2 F3 F4 F5

Press the Enter key to confirm the character string.

1: CALL PROC_1 ('Tool 12')

```
                 JOINT 10%
CALL PROC_1 (1, 'TOOL12')
       [CHOICE]     CHANGE
```

F5

To change a character string, move the cursor to the character string and press the [CHANGE] function key. The character string type selection menu appears.

Specifying arguments of the argument register type

To set an argument of the argument register type, press the [CHOICE] function key and select AR[] from the submenu (see "Specifying arguments).

```
Parameter select
 1 R[ ]
 2 Constant
 3 String
 4 AR[ ]
```

1: CALL PROC_1 (AR[...])

Enter an index.

1: CALL PROC_1 (AR[1])

To toggle between direct and indirect index specifications, press the [INDIRECT] function key. The display changes as follows:

AR[R[...]] → AR[AR[...]] → AR[R[...]] → ···

Specifying arguments of the register type

To set an argument of register type, press the [CHOICE] function key and select "4 R[]" from the submenu (see "Specifying arguments).

```
Parameter select
 1 R[ ]
 2 Constant
 3 String
 4 AR[ ]
```

1: CALL PROC_1 (R[...])

Enter an index.

1: CALL PROC_1 (R[1])

To toggle between direct and indirect index specifications, press the [INDIRECT] function key. The display changes as follows:

R[R[...]] → R[AR[...]] → R[R[...]] → ···

Adding arguments

Move the cursor to ")" at the end of the line.

1: CALL PROC_1 (1)

Press function key CHOICE and select an argument type from the submenu (see "Specifying arguments). A new argument can be added to the cursor position.

1: CALL PROC_1 (1, Constant)

Select an argument type and set a value.

1: CALL PROC_1 (1, Constant)	Select the constant type
1: CALL PROC_1 (1, 2)	Set a value of "2"

Inserting arguments

Move the cursor to the argument for which an argument is to be inserted.

```
          JOINT 10%
    5 <None>
    6 <Insert>
```

1: CALL PROC_1 (1, 2)

Press function key [CHOICE] and select <Insert> from the submenu (see ”Specifying arguments). A new argument can be inserted at the cursor position.

1: CALL PROC_1 (1..., 2)

Select an argument type and set a value, index, and so on.

1: CALL PROC_1 (1, R[...],2)	Select the constant type
1: CALL PROC_1 (1, R[3], 2)	Set a value of ”3”

NOTE An argument cannot be inserted when no argument has been set, and at ”)” at the end of a line. The same submenu reappears; select the argument type.

Deleting arguments

Position the cursor to the argument to be deleted.

```
          JOINT 10%
    5 <None>
    6 <Insert>
```

1: CALL PROC_1 (1, 2 , 3)

Press function key [CHOICE] and select <None> from the submenu (see ”Specifying arguments”). The argument is deleted from the cursor position.

1: CALL PROC_1 (1, 3)

NOTE Selecting <None> when no argument has been set, and at ”)” at the end of a line, simply closes the submenu; no argument is deleted.

Specifying argument registers

The following explanation uses a register instruction as an example.

The selections for the right side of a register instruction are as follows:

```
           JOINT 10%
R[1]=...

         [CHOICE]
```

F4

```
REGISTER Statement               JOINT 10%
 1 R[ ]                  5 AI[]
 2 AR[ ]                 6 GO[]
 3 Constant              7 GI[]
 4 AO[ ]                 8--- next page---
```

To use an argument with the instruction, select AR[] from the menu.

1: R[1]=AR[...]

Specify the index.

1: R[1]=AR[1]

```
DIRECT INDIRECT [CHOICE]
```

F3

If function key F3 ”Indirect specification” is pressed twice at an element having an index, an argument register can be used for an indirect index specification.

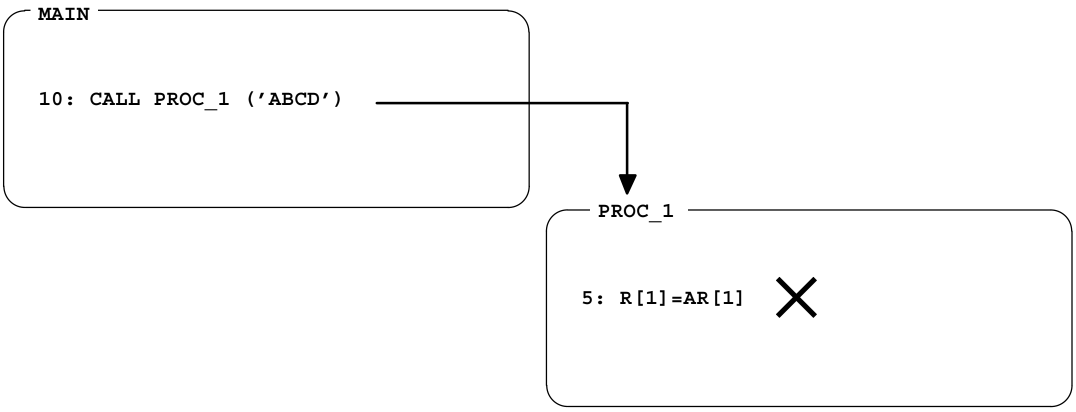
1: WAIT R[R ...]]	When F3 is pressed once
1: WAIT R[AR ...]]	When F3 is pressed again

Notes on using arguments

Note the following when specifying arguments:

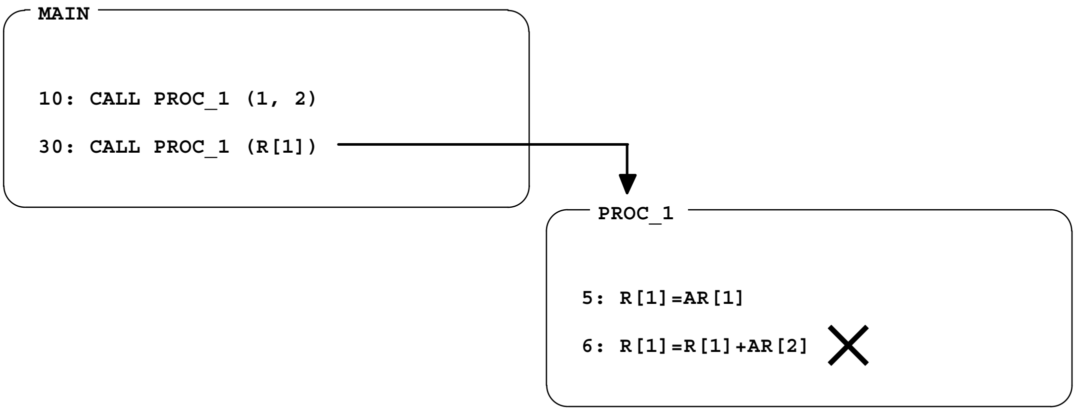
- The contents of an argument are not checked when the argument is specified. If the type of an argument does not match the type of the corresponding one in the subprogram, an error occurs during execution.

  Example In this example, although a value of AR[1] is assigned to the register in subprogram PROC_1, an argument of character string type is specified in the main program. An error occurs when line 5 of the subprogram is executed.



- The number of arguments is not checked when arguments are specified. Even if the number of arguments is not correct, no errors occur if the arguments specified in the main program are not used in a subprogram.

  Example In this example, only one argument is specified in the main program, but two arguments are used in subprogram PROC_1. An error occurs when line 6 of PROC_1 is executed.



Notes on specifying arguments for a program call instruction

- When the program name is changed, the arguments that have been set are kept intact.
- When the program call instruction itself is re–specified, not only the program name but all the arguments are deleted.

Notes on specifying arguments for a macro instruction

- When the macro name is changed, those arguments that have been set are kept intact.

Notes on execution

As described in "Notes on using arguments," the contents and number of arguments to be passed between the calling program and the called program are not checked when they are specified. If an argument is set or used incorrectly, an error occurs on a line where a conflict is detected during program execution.

- Check that the number of arguments specified in the main program is equal to that of the arguments used in the subprogram.
- If the arguments specified in the main program are not used in the subprogram, an error does not occur.
- Check that the contents of the arguments specified in the main program match the types of instructions in the subprogram that use those arguments.
- Check that the indexes and values of the specified arguments are set correctly.

1: CALL PROC_1 (Constant) An error occurs because the value is uninitialized
2: CALL PROC_1 (R[...]) The index is uninitialized

When lines containing these are executed, the error "INTP–201 Unspecified Statement" occurs.

System variables relating to arguments

The argument–attached program call/macro instruction function displays, as selections, the character strings set as system variables when an argument of the character string type is to be selected. These system variables are given below.

Table 4–5. System Variables Relating to Arguments

Item	System variable	Remarks
Single character string type	`$STRING_PRM=TRUE/FALSE` `Standard value=FALSE`	(Note)
Two–character string	`$ARG_STRING[i].$TITLE` `(i = 1 to 10)`	More than 1 and up to 16 characters (Note)
Three–character string	`$ARG_STRING[i].$ITEMJ` `(i = 1 to 10, j=1 to 20)`	Up to 16 characters (Note)
Four word at character entry	`$ARG_WORD [i]` `(i = 1 to 5)`	Up to 7 characters (Note)

NOTE Arguments of character string is able to use on KAREL program only.

4.8 Wait Instructions

A wait instruction is used to stop program execution for a specified period of time or until a condition is satisfied. Two types of wait instructions are available.

```
Instruction                       JOINT 30%
1 Registers          5 JMP/LBL
2 I/O                6 CALL
3 IF/SELECT          7 Palletizing
4 WAIT               8 ---next page---
PROGRAM
```

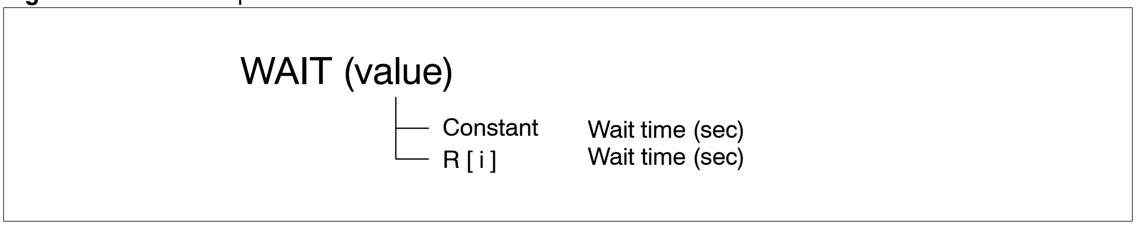
- Time–specified wait instruction: Waits program execution for a specified period of time.
- Conditional wait instruction: Waits program execution until a specified condition is satisfied or a specified period of time has elapsed.

4.8.1 Time–specified wait instruction

WAIT (TIME)

The time–specified wait instruction waits program execution for a specified period of time (in seconds).

Figure 4–60. Time–Specified Wait Instruction



Example

```
1: WAIT
2: WAIT 10.5sec
3: WAIT R[1]
```

4.8.2 Conditional wait instructions

WAIT (condition) (processing)

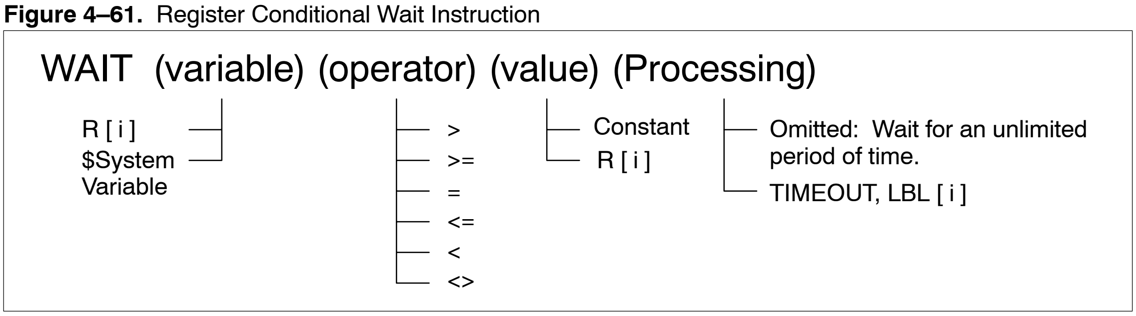
A conditional wait instruction waits program execution until a specified condition is satisfied or a specified period of time has elapsed. Two methods of specifying time–out processing are available:

- If no processing is specified, program execution is waits until a specified condition is satisfied.
- Timeout, LBL[i] is transferred to a specified label if the specified condition is not satisfied until the time specified in 14 WAIT timeout on the system configuration screen.

Register conditional wait instruction

The register conditional wait instruction compares the value of a register with another value, and waits until the comparison condition is satisfied,

Figure 4–61. Register Conditional Wait Instruction



Example

```
3: WAIT R[2] <> 1, TIMEOUT LBL[1]
4: WAIT R[ R[1] ]> = 200
```

I/O conditional wait instruction

The I/O conditional wait instruction compares the value of an input/output signal with another value, and waits until the comparison condition is satisfied.

Figure 4–62. I/O Conditional Wait Instruction 1

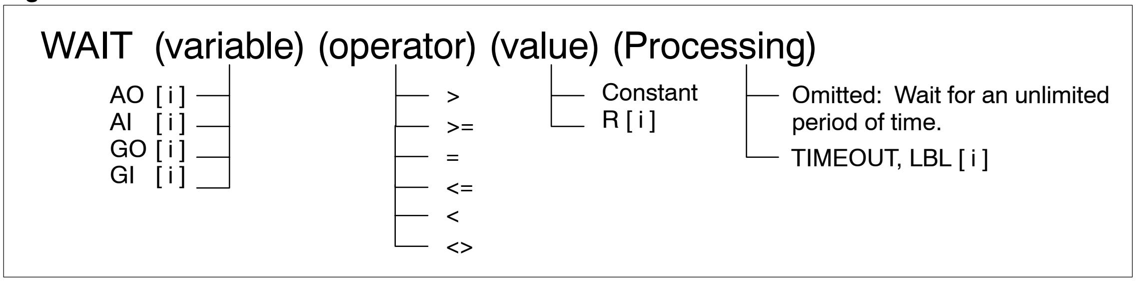
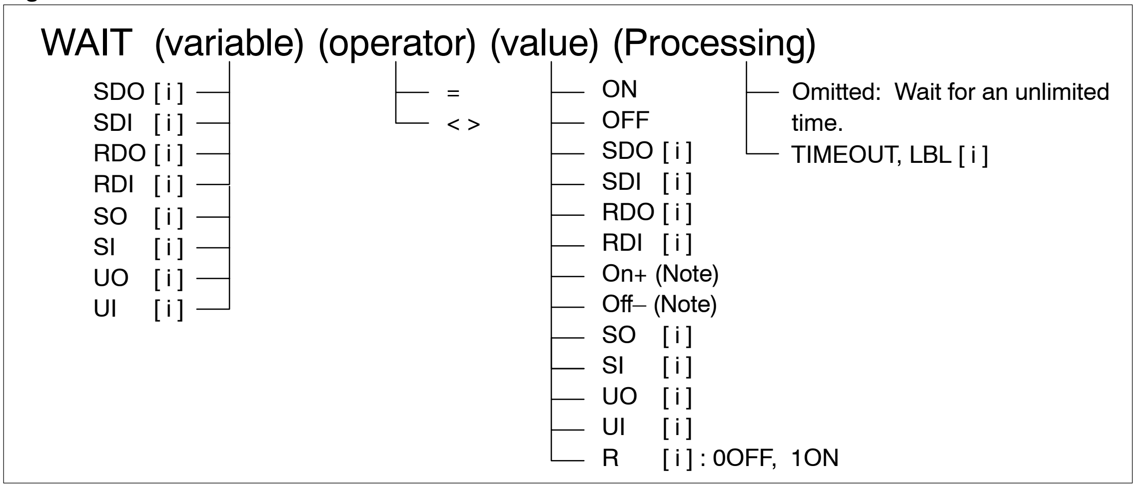


Figure 4–63. I/O Conditional Wait Instruction 2



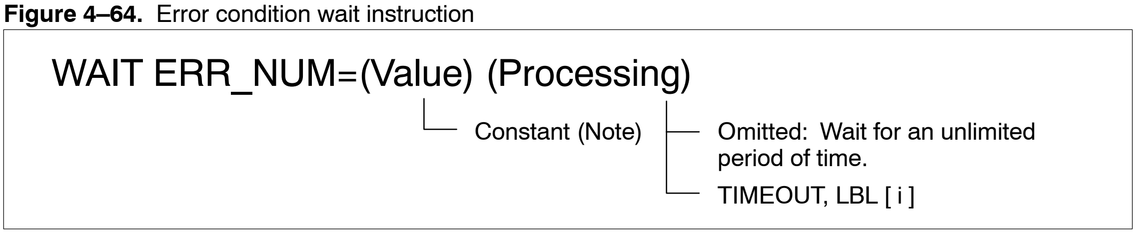
Example
```
5: WAIT SDI[2] <> OFF, TIMEOUT LBL[1]
6: WAIT RDI[ R[1] ] = R[1]
```

NOTE Off–: The falling edge of a signal is regarded as being a detection condition. The condition is not satisfied while the signal remains off. The detection condition is satisfied when the signal changes from the on state to the off state.
On+: The rising edge of a signal is regarded as being a detection condition. The condition is not satisfied while the signal remains on. The detection condition is satisfied when the signal changes from the off state to the on state.

Error condition wait instruction

The error condition wait instruction waits for the occurrence of an alarm having a specified error number.

Figure 4–64. Error condition wait instruction



NOTE An error number is specified with an alarm ID followed by an alarm number.
Error number = aabbb
where aa = alarm ID
bbb = alarm number
For an explanation of alarm IDs and numbers, refer to the alarm code table given in the operator's manual.

Example For SRVO–006 HAND broken, the servo alarm ID is 11, and the alarm number is 006. Thus,
Error number = 11006

In the condition wait instruction, multiple conditions can be specified on a single line in the condition statement, using the logical operators (”and” and ”or”). This simplifies the program structure, allowing the conditions to be evaluated efficiently.

Instruction format

- Logical product (and)
  WAIT <condition 1> and <condition 2> and <condition 3>
- Logical sum (or)
  WAIT <condition 1> or <condition 2> or <condition 3>

If the ”and” (logical product) and ”or” (logical sum) operators are used in combination, the logic becomes complex, impairing the readability of the program and the ease of editing. For this reason, this function prohibits the use of the logical operators ”and” and ”or” in combination.

If multiple ”and” (logical product) or ”or” (logical sum) operators are specified for an instruction on a single line, and one of the operators is changed from ”and” to ”or” or from ”or” to ”and,” all other ”and” or ”or” operators are changed accordingly, and the following message appears:

TRIF–062 AND operator was replaced to OR

TRIF–063 OR operator was replaced to AND

Up to five conditions can be combined with ”and” or ”or” operators on a single line.

Example WAIT <condition 1> and <condition 2> and <condition 3> and <condition 4> and <condition 5>

4.9 Skip Condition Instruction

The skip condition instruction specifies, in advance, a skip condition (condition for executing a skip instruction) used with a skip instruction. Before a skip instruction can be executed, a skip condition instruction must be executed. A skip condition once specified is valid until the execution of the program is completed, or the next skip condition instruction is executed. (For the skip instruction, see Section 4.3.5.)

```
Instruction
1 Miscellaneous      5 MACRO
2 Program control    6 SENSOR
3 Skip               7 Multiple control
4 Offset/Frames      8 ---next page---
```

A skip instruction causes a jump to a branch destination label if the skip condition is not satisfied. If the skip condition is satisfied, a skip instruction causes the robot to suspend the current motion toward a target point, instead executing the program instruction on the next line.
If the skip condition is currently not satisfied, a skip instruction causes a jump to a destination label upon the completion of the current motion.

Figure 4–65. Skip Condition Instruction (Register Condition)

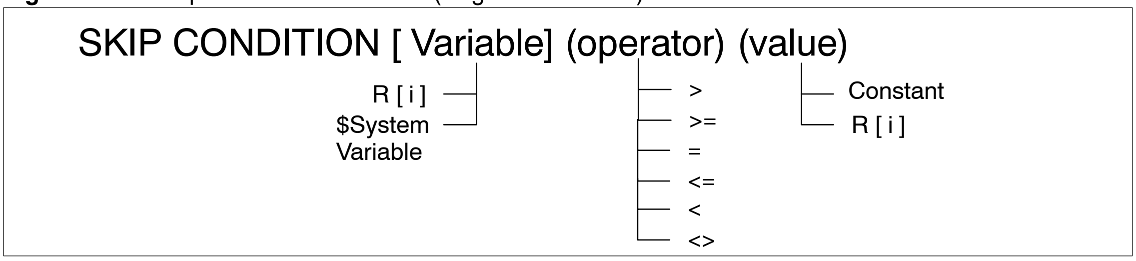


Figure 4–66. Skip Condition Instruction (I/O condition 1)

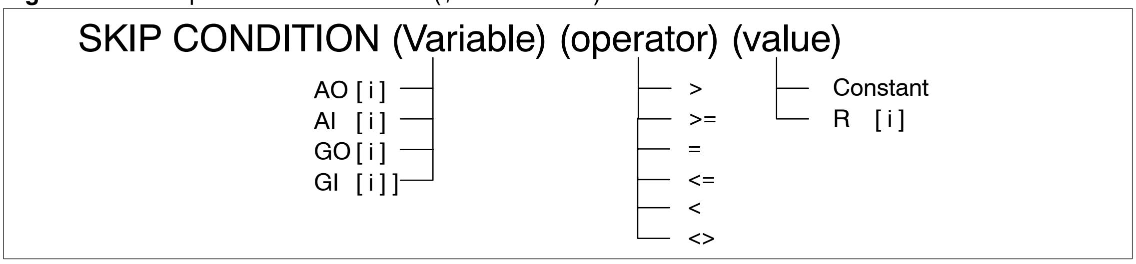
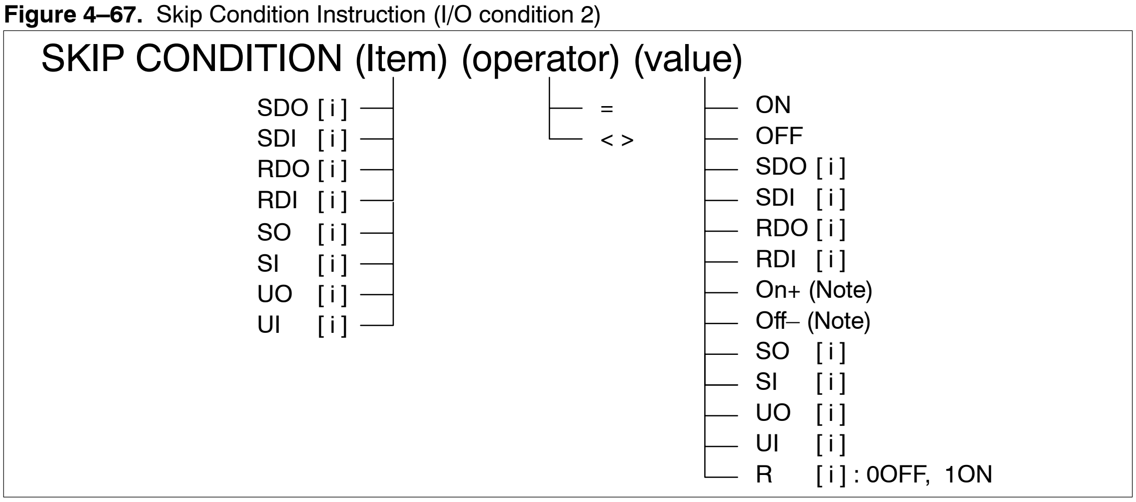


Figure 4–67. Skip Condition Instruction (I/O condition 2)



Example

```
1: SKIP CONDITION DI[ R[1] ] <> ON
2: J  P[1] 100% FINE
3: L  P[2] 1000mm/sec FINE Skip, LBL[1]
4: J  P[3] 50% FINE
5:    LBL[1]
6: J  P[4] 50% FINE
```

NOTE Off–: The falling edge of a signal is regarded as being a detection condition. The condition is not satisfied while the signal remains off. The detection condition is satisfied when the signal changes from the on state to the off state.
On+: The rising edge of a signal is regarded to be a detection condition. The condition is not satisfied while the signal remains on. The detection condition is satisfied when the signal changes from the off state to the on state.

Figure 4–68. Skip Condition Instruction (Error condition)

```
SKIP CONDITION ERR_NUM=(Value)
                          └─ Constant (Note)
```

NOTE An error number is specified with an alarm ID followed by an alarm number.
Error number = aabbb
where aa = alarm ID
bbb = alarm number
For an explanation of alarm IDs and numbers, refer to the alarm code table in the operator's manual.

Example For SRVO–006 Hand broken, the servo alarm ID is 11, and the alarm number is 006. Thus,
Error number = 11006

In the skip condition instruction, multiple conditions can be specified on a single line in the condition statement, using the logical operators ("and" and "or"). This simplifies the program structure, allowing the conditions to be evaluated efficiently.

Instruction format

- Logical product (and)
  SKIP CONDITION <condition 1> and <condition 2> and <condition 3>
- Logical sum (or)
  SKIP CONDITION <condition 1> or <condition 2> or <condition 3>

If the "and" (logical product) and "or" (logical sum) operators are used in combination, the logic becomes complex, impairing the readability of the program and case of editing. For this reason, this function prohibits the use of the logical operators "and" and "or" in combination.

If multiple "and" (logical product) or "or" (logical sum) operators are specified for an instruction on a single line, and one of the operators is changed from "and" to "or" or from "or" to "and," all other "and" or "or" operators are changed accordingly, and the following message appears:

TRIF–062 AND operator was replaced to OR

TRIF–063 OR operator was replaced to AND

Up to five conditions can be combined with "and" or "or" operators on a single line.

Example SKIP CONDITION <condition 1> and <condition 2> and <condition 3> and <condition 4> and <condition 5>

4.10 Offset Condition Instruction

The OFFSET CONDITION instruction specifies the offset condition used in the OFFSET CONDITION instruction,in advance. The OFFSET CONDITION is needed to be executed before the OFFSET instruction is executed. The specified offset condition is effective until the program execution finishes or the next OFFSET CONDITION instruction is executed.(For the offset instruction, see Section 4.3.5.)

```
Instruction                       JOINT 30%
1 Miscellaneous      5
2 Program control    6
3 SKIP               7
4 Offset/Coordinate  8 ---next page---
PROGRAM
```

- The position register specifies the shifting direction and the shift amount.
- When the positional information is expressed in the joint frame,the shift amount of each axis is applied.
- When the positional information is expressed in the Cartesian coordinate system,the number of user frame by which the offset condition is decided should be specified. When it is not specified,the user frame being selected now is used.

CAUTION

If teaching is made by joint coordinates, changing the user coordinate system does not affect the position variables and position registers. If teaching is performed in cartesian coordinates, and the user coordinate system input option is not used, the position variable is not influenced by the user coordinate system. In other cases, both the position variable and position register are influenced by the user coordinate system.

The OFFSET instruction shifts positional information programmed at the destination position by the offset amount specified by position register,and moves the robot to the shifted position.

The shifting condition is specified by the OFFSET CONDITION instruction.

Figure 4–69. Offset Conditional Instruction

OFFSET CONDITION PR [i] (UFRAME [j])

Position register number (1 to 10)

User frame number (1 to 5)

Example

```
1:  OFFSET CONDITION PR[ R[1] ]
2: J  P[1] 100% FINE
3: L  P[2] 500mm/sec FINE Offset
```

4.11 Tool Offset Condition Instructions

A tool offset condition instruction specifies the offset condition used in a tool offset instruction. Execute a tool offset condition instruction before executing the corresponding tool offset instruction. Once the tool offset conditions have been specified, they remain effective until the program terminates or the next tool offset condition instruction is executed. (For the tool offset instruction, see Section 4.3.5 ”Additional motion instructions”)

```
Instruction                        JOINT 30%
1 Miscellaneous      5 Tool_Offset
2 Program control    6 MACRO
3 SKIP               7
4 Offset/Coordinate  8 ---next page---
PROGRAM
```

- The position register specifies the direction in which the target position shifts, as well as the amount of shift.
- The tool coordinate system is used for specifying offset conditions.
- When the number of a tool coordinate system is omitted, the currently selected tool coordinate system is used.
- When the position data is given as coordinates, an alarm is issued and the program stops temporarily.

A tool offset instruction moves the robot to a position shifted from the target position, recorded in the position data, by the offset specified in the tool offset conditions. The condition when the offset is applied is specified by a tool offset condition instruction.

Figure 4–70. Tool Offset Condition Instruction

TOOL_OFFSET CONDITION PR[i] (UTOOL[j])

Position register number
(1 to 10)

Tool frame number
(1 to 9)

Example

```
1: TOOL_OFFSET PR[1]
2: J P[1] 100% FINE
3: L P[2] 500mm/sec FINE Tool_Offset
```

4.12 Frame Instructions

The FRAME instruction is used to change the setting of the Cartesian coordinate system by which the robot works. There are two kinds in the FRAME instruction.

```
Instruction                        JOINT  30 %
  1 Miscellaneous        5 MACRO
  2 Program control      6 SENSOR
  3 SKIP                 7 Multiple control
  4 Offset/Coordinate    8 ---next page---
PROGRAM
```

- Frame setup instruction – The definition of the specified frame is changed.
- Frame select instruction – The frame number being selected now is changed.

The frame setup instruction

The tool frame setup instruction changes the setting of the tool frame specified by the tool frame number in this instruction. The user frame setup instruction changes the setting of the user frame specified by the user frame number in this instruction.

Figure 4–71. Tool Frame Setup Instruction

UTOOL [i] = (value)

Tool frame number (1 to 9)

PR [i]

Figure 4–72. User Frame Setup Instruction

UFRAME [i] = (value)

User frame number (1 to 9)

PR [i]

Example

```
1: TOOL[1] = PR[1]
2: UFRAME[3] = PR[2]
```

Frame select instruction

The tool frame select instruction changes the current tool frame number.
The user frame select instruction changes the current user frame number.

Figure 4–73. Tool Frame Select Instruction

UTOOL_NUM = (Value)

R [i]
Constant   Tool frame number (0 to 9)

Figure 4–74. User Frame Select Instruction

UFRAME_NUM = (Value)

R [i]
Constant   User frame number (0 to 9)

Example

```
1:  UFRAME_NUM = 1
2: J P[1] 100% FINE
3: L P[2] 500mm/sec FINE
4:  UFRAME_NUM = 2
5: L P[3] 500mm/sec FINE
6: L P[4] 500mm/sec FINE
```

4.13 Program Control Instructions

The program control instructions control program execution.

```
Instruction                        JOINT 30%
 1 Miscellaneous      5 MACRO
 2 Program control    6 SENSOR
 3 SKIP               7 Multiple control
 4 Offset/Coordinate  8 ---next page---
PROGRAM
```

- Halt instruction
- Abort instruction

4.13.1 Halt instruction

PAUSE

The halt instruction stops program execution, causing the robot in motion to decelerate and stop:

- If an operation instruction is being executed, the program stops before the operation is completed.
- The cursor moves to the next line. When restarted, the program is executed from this line.
- If the program timer is active, it is stopped. When the program is restarted, the program timer is activated.
- If a pulse output instruction is being executed, the program stops after that instruction has been executed.
- If an instruction other than a program call instruction is being executed, the program stops after that instruction has been executed. A program call instruction is executed when the program is restarted.

Figure 4–75. Halt Instruction

PAUSE

4.13.2 Abort instruction

ABORT

The abort instruction aborts program execution in the following way, causing the robot in motion to decelerate and stop:

- If an operation instruction is being executed, the program stops before the operation is completed.
- The cursor stops on the current line.
- When the abort instruction is executed, the execution of the program cannot be continued. Information held by a program call instruction about the main program is lost.

Figure 4–76. Abort Instruction

ABORT

4.14 Other Instructions

The following miscellaneous instructions are available:

```
Instruction                          JOINT 30%
1 Miscellaneous      5 MACRO
2 Program control    6 SENSOR
3 SKIP               7 Multiple control
4 Offset             8 ---next page---
PROGRAM
```

- RSR instruction
- User alarm instruction
- Timer instruction
- Override instruction
- Comment instruction
- Message instruction
- Parameter instruction
- Maximum speed instruction

4.14.1 RSR instruction

RSR [i] = (value)

The RSR instruction alternately enables and disables the RSR function having a specified RSR number.

Figure 4–77. RSR instruction

RSR [i] (value)

(1 to 4)

ENABLE: Enable RSR function
DISABLE: Disable RSR function

Example `RSR[2:Workproc.2.]=ENABLE`

4.14.2 User alarm instruction

UALM[i]

The user alarm instruction displays the alarm message corresponding to an already set user alarm number on the alarm display line. The user alarm instruction pauses the program which is on progress. A user alarm is specified on the user alarm setting screen(See Section 3.11) and this setting is registered in the system variable $UALM_MSG . The total number of user alarms can be changed at a controlled start (See Section B.1, "Start mode").

Figure 4–78. User Alarm Instruction

UALM [i]

Alarm number

Example `1: UALM[1] ($UALRM_MSG[1] = WORK NOT FOUND`

4.14.3 Timer instruction

timer [i] = (state)

The timer instruction starts/stops the program timer. The operating state of the program timer can be viewed on program timer screen STATUS PRGTIMER (option).

Figure 4–79. Timer instruction

Timer [i] (processing)

Timer Number

- SATRT: Start timer
- STOP: Stop timer
- RESET: Reset timer

Example

```
1: TIMER [1]=START
   TIMER [1]=STOP
   TIMER [1]=RESET
```

The value of the timer can be referenced in a program, using a register instruction. It is possible to determine whether the timer has overflowed by using a register instruction. The program timer overflows if it exceeds 2147483.647 seconds.

R[1]=TIMER [1]
R[2]=TIMER_OVER FLOW[1]
0: Not over flow
1: Over flow

4.14.4 Override instruction

OVERRIDE = (value)%

The override instruction changes a feedrate override.

Figure 4–80. Override Instruction

OVERRIDE = (value) %

+ R[i]
+ Const
+ AR[]

(value) : Feedrate override (1 to 100)

Example

```
1: OVERRIDE = 100%
```

4.14.5 Comment instruction

!(Remark)

The comment instruction adds a comment in a program. A comment has no effect on program execution. A comment specified in a comment instruction can consist of up to 32 characters including alphanumeric characters, asterisk s (*) underlines (_), and at marks (@). To add a comment, press the ENTER key.

Figure 4–81. Comment Instruction

! (Remark)

A comment can consist of up to 32 characters including alphanumeric characters, asterisks (*), underlines (_), and at marks (@ etc...).

Example

```
1: !APPROACH POSITION
```

4.14.6 Message instruction

MESSAGE[message statement]

The message instruction displays a specified message on the user screen. (For the user screen, see Section 7.2.) A message can consist of up to 24 characters including alphanumeric characters, asterisks (*), underlines (_) , and at marks (@). To add a comment, press the ENTER key.

Figure 4–82. Message Instruction

Message [message statement]

A message statement can consist of up to 24 characters including alphanumeric characters, asterisks (*), underlines (_), and at marks (@).

Example `1: MESSAGE[ DI[1] NOT INPUT ]`

4.14.7 Parameter instruction

$(SYSTEM VARIABLE NAME) = (value)

The parameter instruction changes the value of a system variable. This instruction can be used only for a system variable containing a numeric value (constant). You can enter the parameter name after pressing the ENTER key. It is possible to enter the parameter name up to 30 characters or less without the first character,"$".

There are two types of system variables, variable type and position type. A system variable of variable type can be assigned to a register. A system variable of position type can be assigned to a position register.

System variables of position data type are divided into three data types, cartesian (XYZWPR type), joint type (J1–J6 type), and matrix type (AONL type).

When a system variable of position data type is assigned to a position register, the data type of the position register is converted to the data type of the system variable.

If a system variable of position type is assigned to a register, or if a system variable of variable type is assigned to a position register, the following alarm is generated during execution.

INTP–240 Incomputible datatype

Figure 4–83. Parameter Instruction (Writing)

$ (SYSTEM VARIABLE name) = (value)

System variable name

System variable value (numeric value)
R [X]
PR [X]

Example `1: $SHELL_CONFIG.$JOB_BASE = 100`

Figure 4–84. Parameter Instruction (Reading)

(value) = $ (SYSTEM VARIABLE name)

R [X]
PR [X]

System variable name

Example `1: R[1] = $SHELL_CONFIG.$JOB_BASE`

WARNING

The operation of the robot and control unit is controlled with system variables. Only a person who is aware of how changes to the system variables will affect the system should set system variables. If a person without detailed knowledge attempts to set the system variables, the robot and control unit would malfunction, causing injury to personnel or damage to equipment.

Procedure 4–3 Specifying parameter instructions

Step

1 On the program edit screen, press function key [INST]. Select item Miscellaneous from the menu. Then, select item Parameter name from the menu.

```
Miscellaneous stat                    JOINT 10%
 1 $...=...          5
 2 ...=$...          6
 3                   7
 4                   8
PNS0001
                                          1/1
[End]


Select item
                          [CHOICE]
```

2 Select item 2 ”... =$ ...”

```
Miscellaneous stat                    JOINT 10%
 1 R[ ]              5
 2 PR[ ]             6
 3                   7
 4                   8
PNS0001

1: ...=$...
[End]

Select item
                          [CHOICE]
```

3 Select item 1 ”R[ ]” and enter the desired register number.

```
PNS0001                               JOINT 10%
                                          1/2
1: R[1]=$...
[End]





Press ENTER

                          [CHOICE]
```

4 To display the system variable menu, press the [CHOICE] function key.
To enter a character string, press the Enter key.

When function key CHOICE is pressed

```
 Parameter menu                   JOINT 10%
  1 DEFPULSE        5
  2 WAITTMOUT       6
  3 RCVTMOUT        7
  4                 8 --- next page ---
PNS0001
                                       1/2
 1: R[1]=$...
 [End]

 Select item
                      [CHOICE]
```

5 Select item 1 ”DEFPULSE.”

```
 PNS0001                          JOINT 10%
                                       1/2






 1: R[1]=$DEFPULSE
 [End]

[INST]                              [EDCMD]
```

6 When ENTER is pressed

```
                                  JOINT 10%
  1 Words
  2 Upper Case
  3 Lower Case
  4 Options            -- Insert --
 PNS0001

 1: R[1]=$...
 [End]


        $     [      ]    .
```

7 Enter the desired system variable name.

4.14.8 Maximum speed instructions

A maximum speed instruction specifies the maximum operating speed of a program. There are two maximum speed instructions, the instruction for specifying the joint operation speed and that for specifying the path control operating speed. If a speed exceeding the speed specified with a maximum speed instruction is specified, the speed specified with the maximum speed instruction is assumed.

JOINT_MAX_SPEED[i]=(value)

Figure 4–85.

```
JOINT_MAX_SPEED [ i ] = (value)
                            ├── Comstant (deg/sec)
                            └── R [i] (deg/sec)
```

Example `JOINT_MAX_SPEED[3] = R[3]`

LINEAR_MAX_SPEED= (value)

Figure 4–86.

```
LINEAR_MAX_SPEED = (value)
                       ├── Comstant (deg/sec)
                       └── R [i] (deg/sec)
```

Example `LINEAR_MAX_SPEED = 100`

4.15 Multiaxis Control Instructions

Multiaxis control instructions control the execution of a multitask program.

These instructions can be specified and executed only when the multitask option is supported.

```
Instruction
1 Miscellaneous       5 MACRO
2 Program control     6 SENSOR
3 Skip                7 Multiple control
4 Offset/Frames       8 --- next page ---
```

- Semaphore instruction
- Semaphore wait instruction
- Program execution instruction

4.15.1 Semaphore instruction

The semaphore instruction switches a semaphore, specified with a number, between on and off. A semaphore is a kind of switch used to synchronize the execution of tasks.

- For example, if the execution of a certain step and subsequent steps of a program is to wait until a certain condition is satisfied in another program being executed simultaneously, first turn off the semaphore having the specified number and make the program wait until the semaphore is turned on, using the semaphore wait instruction (–>Section 4.15.2, "Semaphore wait instruction"). When the semaphore is turned on by the other program that is being executed simultaneously, the program is released from the wait state. This allows the execution of multiple programs at prescribed timings.
- Each semaphore remembers how many times it has been turned on. When a wait statement for waiting for a semaphore whose "on" count is other than zero is executed, its execution ends immediately and the "on" count is decremented by 1.
- When a program uses a wait statement to wait for a semaphore whose "on" count is zero, the program enters the wait state until the semaphore is turned on by another program.
- When a semaphore is turned off, the "on" count is cleared to zero.
- Semaphores numbered 1 through 32 can be used.

Figure 4–87. Semaphore instruction

SEMAPHORE [i] = (value)

Semaphore number (1 to 32)

(value): ON / OFF

4.15.2 Semaphore wait instruction

The semaphore instruction causes the program to wait until a semaphore having a specified number is turned on.

- The program is placed in the wait state until the specified semaphore is turned on by another program.
- Either unlimited waiting (waiting time not limited) or specified time waiting can be selected. The waiting time is specified for system variable $WAITTMOUT, in the same way as in normal wait statements.

Figure 4–88. Semaphore wait instruction

WAIT SEMAPHORE [i] = (Processing)

Semaphore number (1 to 32)

(Processing): Omitted: Forever / TIMEOUT, LBL [i]

4.15.3 Program execution instruction

During the execution of a program, the program execution instruction starts the execution of another program.

- The difference from the program call instruction is that, with the program call instruction, those lines following the call instruction are executed after the called program has been executed, whereas with the program execution instruction, the program that starts the execution of another program continues concurrently.
- To synchronize programs that are being executed simultaneously, use the semaphore instruction and the semaphore wait instruction.
- If an attempt is made to execute a program for which the same motion group is specified, an alarm is generated. If this occurs, specify a different motion group.

Figure 4–89. Program execution instruction

RUN (Program name)

Example

PROG1	PROG2
1: SEMAPHORE[1]=OFF	1: J P[3] 100% FINE
2: RUN PROG1	2: J P[4] 100% FINE
3: J P[1] 100% FINE	3: J P[5] 100% FINE
4: J P[2] 100% FINE	4: J P[6] 100% FINE
5: WAIT SEMAPHORE[1]	5: SEMAPHORE[1]=ON
MOTION GROUP[1,*,*,*,*]	MOTION GROUP[*,1,*,*,*]

4.16 Operation Group Instructions

The operation group instructions enable the following in single–line operation instructions in a program having multiple operation groups:

- Specification of the operation format for each operation group (excluding the arc)
- Specification of the feedrate for each operation group
- Specification of the positioning format for each operation group

This allows each operation group to operate asynchronously.

These instructions can be specified and executed only when the multitask option is supported.

```
Instruction
1 Miscellaneous      5 MACRO
2 Program control    6 Independent GP
3 Skip               7 Simultaneous GP
4 Offset/Frames
```

- Asynchronous operation group instruction
- Synchronous operation group instruction

With ordinary operation instructions for which these operation group instructions are not specified, all operation groups are executed with the same operation format, feedrate, and positioning format, and are synchronized with the operation add instructions. The operation group having the longest travel time is that with which the other operation groups are synchronized.

4.16.1 Asynchronous operation group instruction

The asynchronous operation group instruction controls operation groups asynchronously, with the operation formats, feedrates, and positioning formats specified separately for the individual operation groups.

Figure 4–90. Asynchronous operation group instruction

Independent GP
GPi (Operation statement of operation group i)
GPj (Operation statement of operation group j)

Operation group number (operation group of the program)

Operation statement for the operation group

4.16.2 Synchronous operation group instruction

The synchronous operation group instruction controls operation groups synchronously, with the operation formats specified separately for the individual operation groups.

- As with ordinary operation instructions, the operation group having the longest travel time is that with which the other operation groups are synchronized. Thus, the feedrate is not always the same as that specified in the program.
- The positioning format for an operation group with the smallest CNT value (closest to FINE) is also applied to the other operation groups.

Figure 4–91. Synchronous operation group instruction

Simultaneous GP
GPi (Operation statement of operation group i)
GPj (Operation statement of operation group j)

Operation group number (operation group of the program)

Operation statement for the operation group

5. PROGRAMMING

This chapter describes how to create and change a program for moving the robot.

☐ Contents of this chapter

Various program instructions are issued with the robot and peripherals to specify robot and hand motions. When these instructions are combined together, they create what is called a hand application program. A hand application program for instance, can:

- Move the robot to desired positions in the operating area along the specified path
- Handle workpiece
- Send output signals to the peripherals
- Receive input signals from the peripherals

Before programming, design the outline of a program. In the design, incorporate the most effective method for the robot to do the target work. This enables efficient programming and ensures that only the instruction s appropriate for the purpose are used.

Instructions must be selected from menus displayed on the teach pendant during programming. To teach a target position to the robot, the robot must be moved to the target position by jog feed.

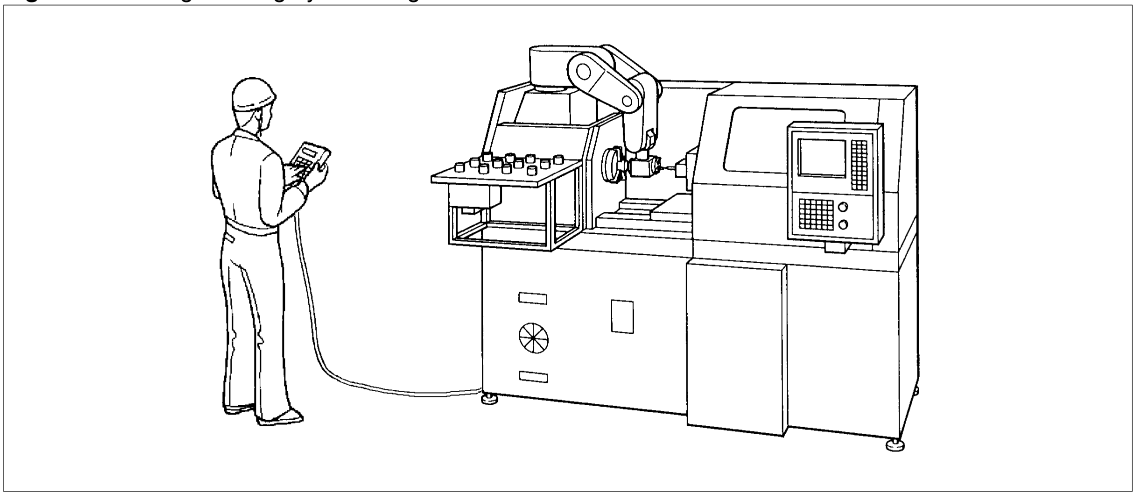
After you have finished creating the program, change the program if necessary. To change, add, delete, copy, find, or replace an instruction, select the desired item from the menu displayed on the teach pendant.

This chapter describes the following:

- Tips on effective programming
- Turning on the power and jog feed
- Creating a program
- Changing a program

See Chapter 4 for the configuration of a program and the program instructions.

Figure 5–1. Programming by Teaching



5.1 Tips on Effective Programming

This section describes tips on effective programming. The following items are explained:

- Motion instructions
- Fixed positions

NOTE This section describes tips on programming, but does not describe tips on jog feed.

5.1.1 Motion instructions

Refer to the following instructions when teaching motions to the robot.

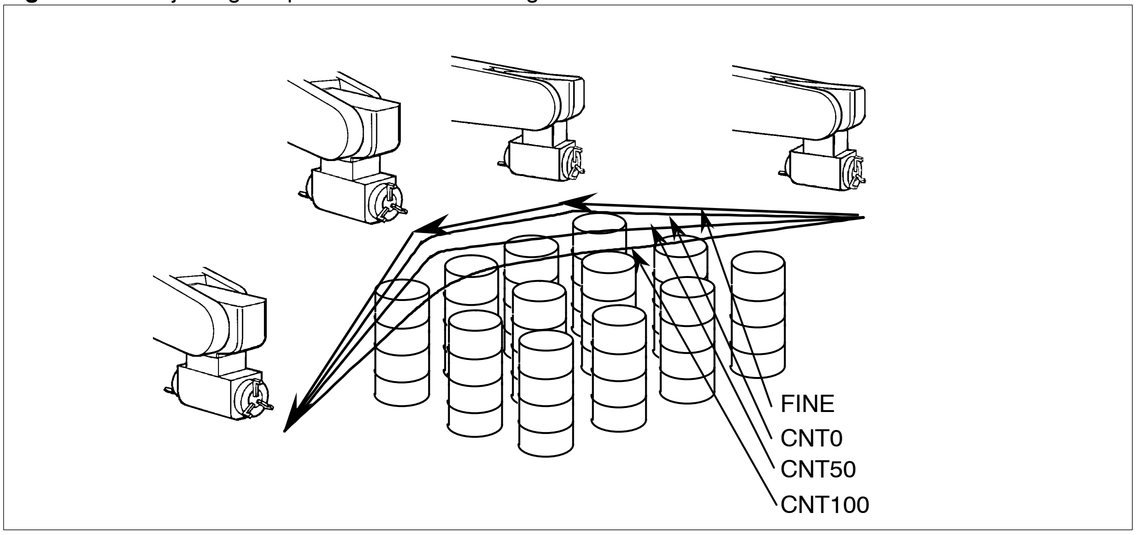
Workpiece hold position = FINE positioning

Use FINE positioning for all workpiece hold positions. The robot stops exactly at the workpiece hold position. When CNT positioning is used (explained next), the robot does not stop at taught points.

Moving around workpieces = CNT positioning

Use CNT positioning for moving around workpieces. The robot continuously moves to the next target point without stopping at taught points. If the robot moves near the workpieces, adjust the path of CNT positioning.

Figure 5–2. Adjusting the path of CNT Positioning

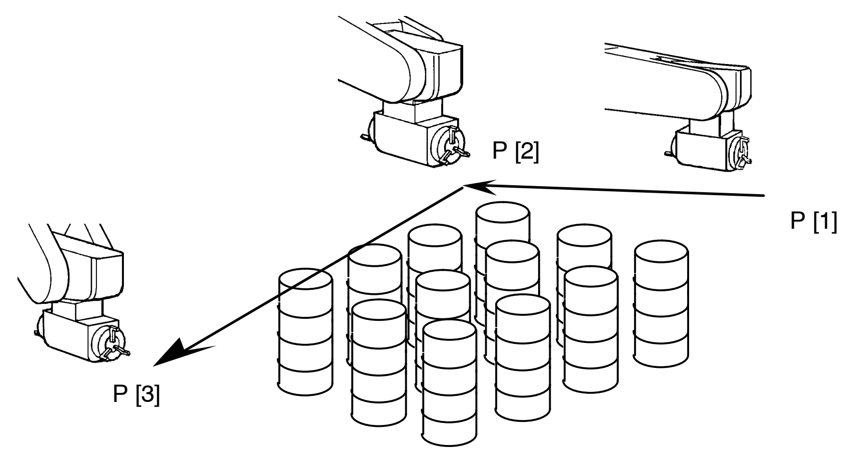


Fixing the attitude of the tool

Cycle time is wasted when the robot motion considerably changes the attitude of the tool. The robot moves much faster when the attitude of the tool is changed smoothly and gradually. Teach positions so that the attitude of the tool changes as gradually as possible with respect to the robot.

When the attitude of the tool must be changed considerably, teach one large motion by dividing it into several small motions. Namely, teach positions so that the attitude of the tool changes gradually.

Figure 5–3. Teaching Positions According to the Tool Attitude



To change the attitude of the hand as smoothly as possible:

1 Teach the first position of the work so that the robot has a normal attitude.
2 Move the robot to the last position of the work by jog feed. Then check that the robot has a normal attitude.
3 Teach the last position.
4 In accordance with the work, teach a position between the first and last positions.
5 Select a Cartesian coordinate system (World, user or jog coordinate system) and move the robot to the first position by jog feed.
6 Select the Cartesian coordinate system, move the robot toward the last position by jog feed, then stop the robot at the next position to be taught.
7 Correct the taught position so that the robot has a normal attitude.

WARNING

If the J5 axis passes singular points (near 0 degrees) when the robot is operated by setting the move type to linear, the additional move instruction with no attitude must be used for these points, or the move type must be changed from linear to axial.

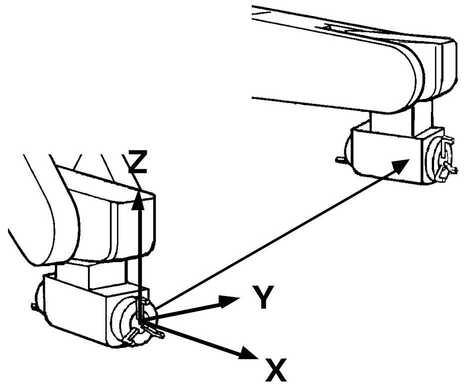
8 Repeat steps 6 and 7 for all the remaining positions to be taught between the first and last positions.

Special motions of the LR Mate 100

Since the LR Mate 100 is a five–axis robot, it has some attitudes which are impossible. The following structural limits are therefore set for linear and circular motions under attitude control. In such cases, use joint motions or wrist joint motions (WRIST JOINT) to move the robot.

1 During linear jog feed, if the LR Mate is programmed to move at an impossible attitude, it instead moves at the possible attitude which most closely approximates the impossible one. Therefore, care must be taken because this motion sequentially changes the attitude of the tool. Circular jog feed is impossible (W, P, and R keys are disabled).

2 The attitude control is enabled in the LR Mate 100 only when the surface of the flange faces up or down. Otherwise, care must be taken because the LR Mate 100 moves at the possible attitude which most closely approxinates the impossible attitude. Moreover, if the surface of the flange faces up or down during joint jog,the waring message, "Horizontal fixture position", is displayed and the jog motion pauses.



3 When the surface of the flange faces up or down, the attitude of the wrist is changed (NOFLIP when the surface of the flange faces the body of the robot or FLIP when the surface of the flange faces away from the body of the robot). The robot cannot move under attitude control if the attitude of the wrist at the start point is different from that at the end point. In this case, use joint motions or wrist joint motions (WRIST JOINT) to move the robot.

4 Care must be taken when the control point, namely the intersection of the J4– and J5–axes, passes near the J1–axis. This is because the robot may move around the J1– or J5–axis at high speed.

5 A stroke limit error occurs when the robot moves 180° or more per stroke around the J5–axis. To prevent this, the angle per stroke for linear or circular motion must always be less than 180°.

6 The accuracy of path control or palletizing tends to worsen near the boundary of the operating area (when the J2– and J3–axis arms are arranged in a straight line). Avoid teaching positions near the boundary if possible . When teaching positions cannot be avoided, add as many positions to be taught as possible.

5.1.2 Predefined position

The predefined position is the position that is referenced many times in a program. The predefined positions that are used often are the pounce position and the home(perch) position. You should define these positions to program efficiently or delete cycle time.

Pounce position

The pounce position is the reference position for all work. This is the safe position away from the motion area of the machine tool and peripheral device.

Home(perch) position

The home position,or perch position,is a safety position away from the machine tool and the workpiece transfer area. The reference position digital output signal is turned on when the robot is at this position.(See Section 3.9, ”Setting a Reference Position)

NOTE HOME is a peripheral device I/O input signal, and does not represent a home position. A reference position is one of the home positions, but there is no utility used to move the robot to the reference position.

Other predefined position

The pounce position, reference position, or any other position can be defined as a predetermined position. Specify those positions that are frequently used in a program as predetermined positions.

When using the fixed position, use position registers (See Section 7.4) and macro instructions (See Section 9.1).

CAUTION

If the position variable and position register are taught according to joint type, they are not affected when the user coordinate system is changed. If the position variable is taught according to cartesian coordinates, and the user coordinate system input option is not used, the position variable is not affected by the user coordinate system. In other cases, both of the position variable and position register are affected when the user coordinate system is changed.

NOTE To move the robot to the same spatial position when the position register is shared by two programs, the two programs must have the same tool and user coordinate system.

5.2 Turning on the Power and Jog Feed

5.2.1 Turning on the power and turning off the power

Turning on the power starts up the robot system. Turning on the power normally executes internal processing called a cold start or hot start, then the system is started up. The special operation is necessary to perform processing with a control or initial start.(See Section B.1, ”START MODE”)

> **CAUTION**
>
> Some systems require inspection before the robot is turned on. For the sake of safety, the system should be checked before the robot is turned on.

Hot start

You can select hot start if the hot start is setup when you start the robot system. The hot start is the function that save the condition of the system just before power off and revives it after the next power on.(See Section 3.14, ”System config menu”)

- If the hot start is set to disable($SEMIPOWERFL=FALSE), the system starts up with the cold start. In cold start mode,the system software of the controller is initialized during starting. When you change the setting of the system such as I/O configuration,you should start up in cold start mode.
- If the hot start is set to be effective($SEMIPOWERFL=TRUE), the system starts up in hot start mode. In hot start mode,the system software of the controller is not initialized during starting up.

HOT START done signal

You can set that the digital output signal(DO) is turned on when the hot start is finished. This function is set with the system configuration screen [6 SYSTEM.Config].(See Section 3.14, ”System Config Menu”)

Automatic start program

An automatic start program can be specified. The program is automatically started when the power is turned on. If override and parameter instructions are specified in the program to be started, the system can be customized when the power is turned on.

- In Autoexec program for Cold start of the system setting menu, register a program to be automatically started when power interrupt handling is disabled. Such a program, if not defined, is not started.

The automatic start program cannot operate the robot. The automatic start program is used to set up the system or initialize the state of I/O,etc.(See Section 3.14, ”System config menu”)

Program selection after power on

The condition of the program selection after the power on is the following:

- When hot start is disable,it depends on the setting of the system variable,$DEFPROG_ENB. You can set $DEFPROG_ENG with the system config menu.
  - TRUE : The program which had been selected at the power off is selected as it is.
  - FALSE : No program is selected.
- When hot start is effective,a program which had been selected at power off is selected as it is.

System condition

The table below lists settings in different start modes.

Table 5–1. System Statuses in Different Start Modes

	Hot start	
	Effective	**Disable(default setting)**
Contents of register	○	○
Override	○	× [10%]
Selection program	○	△ (NOTE 3)
Execution line	○	× [First line]
Condition of I/O	○ (NOTE 1)	× [All off]
TP screen	△ (NOTE 2)	× [Hint screen]

○ : All values that are current at power–down are saved and restored at power–up.

△ : Only some of the values that are current at power–down are saved.

× : The values that are current at power–down are not saved. At power–up, the default values are set.

NOTE 1 Generally, the status existing at power–down is restored, but digital output (SDO), being performed by a pulse instruction at power–down, is turned off. To restore the I/O status, specify the desired restoration status in [6 SYSTEM Config] (see Section 3.14).
Even if power interrupt handling is enabled, none of the output signals are resumed, but all output signals are turned off in the following cases:

- The I/O allocation was changed before power–off.
- The fuse of the I/O device blew, or the power to the I/O device was turned off.
- The I/O device configuration was changed.

NOTE 2 The screen type selected at power–down is restored, but the page, cursor, and other screen statuses are not restored. Instead, the screen is restored using the same page, cursor, and other screen statuses assumed immediately after a cold start.

NOTE 3 The name of the main program that calls the subprogram is stored.

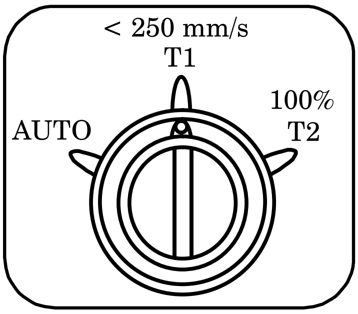
CAUTION

Before the power is turned on, system statuses in the corresponding start mode described above should be checked.

5.2.2 Three–Mode Switch

The three–mode switch is a key operation switch installed on the operator's panel or operation box. This switch is used to select an optimum robot operation mode according to the robot operation conditions and use status. There are operation modes AUTO, T1, and T2. See Figure 5–4.

Figure 5–4. Three–mode switch



When the three–mode switch is used to switch between operation modes, a message appears on the screen of the teach pendant, and the robot halts. When the key is removed from the switch, the switch setting position can be fixed. (For the CE and RIA specifications, when the switch is set to T2 mode, the key cannot be removed to fix the switch setting position.)

> **CAUTION**
> For the RIA specification, if switching between T1 or T2 mode and AUTO mode is made with the deadman's switch kept holded, a system error occurs. In this case, selected mode is not set until the deadman's switch is released. Release the deadman's switch, then hold the deadman's switch again.

Connection: Connect the *FENCE signal to the protective fence. Make a connection in such a way that, when the protective fence is open, the signal entered to the robot is off and, when it is closed, the signal is on.
The *SFSPD signal can be used in accordance with the design of your system.

The following explains the operation modes that can be selected using the three–mode switch:

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

This mode is intended for use to teach the position of operation to the robot. It can also be used to check the robot path at low speed and the program sequence.

Program execution: A program can be executed only from the teach pendant.

Robot speed at jogging

- The speeds at the tool tip and flange are both limited not to exceed 250 mm/sec.

Robot speed at executing program

- The override value can be increased to up to 100%, but the speeds at the tool tip and flange surface are limited to 250 mm/sec or slower. For example, if the taught speed is 300 mm/sec, the speeds at the tool tip and flange surface are limited to 250 mm/sec. If the taught speed is 200 mm/sec, they are not limited. Even when the taught speed is 250 mm/sec or below, the speed on the flange surface may exceed 250 mm/sec in a portion (for example, a corner) where the posture of the tool changes. In this case, the actual operation speed is limited.
  The warning message MOTN – 231 T1 speed limit (G:i) appears only if the operation speed is limited and the taught speed is 250 mm/sec or below.
- Speed limitation is performed based on the taught speed with an override value of 100%.
  Therefore, if the taught speed is, for example, 2000 mm/sec, the operation speed is limited to 250 mm/sec for an override value of 100%. However, the operation speed can be decreased further, for example, to 125 mm/sec by lowering the override value to 50%.

Protective fence: If you want to work with the protective fence kept open, it is necessary to set the three–mode switch to T1 or T2 before starting operating the robot.

- It is possible to operate the robot only when the teach pendant is enabled and the deadman switch is pressed (gripped).
- Disabling the teach pendant puts the robot in an emergency stop alarm condition, so the robot cannot run.
- When the teach pendant is enabled, but the deadman switch is not pressed, the robot is in an emergency stop alarm condition, so it cannot run.

> **CAUTION**
> When checking the program you created, be sure to follow the safety manual.

Fixing operation mode: When the switch is set in the T1 mode position, the operation mode can be fixed to T1 mode by removing the key.

Troubleshooting

- When the switch is set in the T1 mode position, turning off the teach pendant enable switch stops the robot and causes an error message to appear. To release the error, set the teach pendant enable switch to on, then press the RESET key.

T2 (100%): Test mode 2

The T2 mode is intended for use to make a final check of the program you created. In the T1 mode, it is impossible to verify the robot's actual tool path and cycle time because the operation speed is limited. In the T2 mode, it is possible to verify them by running the robot at the production speed because there is basically no speed limitation(*).

* Using a safety speed based on the *SFSPD signal can limit the operation speed of the robot even in the T2 mode by lowering the override value.

(Reference) If *SFSPD is off, the override value is limited to within a value specified by $SCR.$SFRUNOVLIM (default: 30%).

Program execution: A program can be executed only from the teach pendant.

Robot speed at jogging

- The speeds at the tool tip and flange are both limited not to exceed 250 mm/sec.

Robot speed at executing program

- The override value can be increased to up to 100%. There is no special speed limitation.

Protective fence: If you want to work with the protective fence kept open, it is necessary to set the three–mode switch to T1 or T2 before starting operating the robot.

- It is possible to operate the robot only when the teach pendant is enabled and the deadman switch is pressed (gripped).
- Disabling the teach pendant puts the robot in an emergency stop alarm condition, so the robot cannot run.
- When the teach pendant is enabled, but the deadman switch is not pressed, the robot is in an emergency stop alarm condition, so it cannot run.

> **CAUTION**
> When checking the program you created, be sure to follow the safety manual.

Fixing operation mode: When the switch is set in the T2 mode position, the operation mode can be fixed to T2 mode by removing the key. (For the CE and RIA specifications, however, the key cannot be removed.)

Troubleshooting

- When the switch is set in the T2 mode position, turning off the teach pendant enable switch stops the robot and causes an error message to appear. To release the error, set the teach pendant enable switch to on, then press the RESET key.

AUTO: Auto mode

The AUTO mode is intended for use at production.

Program execution: There is no restrictions on program execution. A program can be executed from external devices, operator's panel, and teach pendant. Only when the RIA specification is used, however, program execution from the teach pendant is impossible if the switch is set in the AUTO mode position.

Robot speed at jogging: The robot can be operated at a maximum speed.

Only for the CE and standard specifications, jogging is possible. For the RIA specification, jogging is not possible.

- The speeds at the tool tip and flange are both limited not to exceed 250 mm/sec.

Robot speed at executing program

- The robot can be operated at a maximum speed.

Safety devices: Close the protective fence. When the protective fence is opened during program execution, the robot responds as follows:

<CE and RIA specifications>

- The robot decelerates and stops. After a certain time, the robot enters the emergency stop state. If the protective fence is opened while the robot is operating at a high speed, the robot may enter the emergency stop state during deceleration. In this case, the robot stops immediately at this point.

<Standard specification>

- The robot stops immediately in the same manner as when another emergency stop signal is applied.

Fixing operation mode: When the switch is set in the AUTO mode position, the operation mode can be fixed to AUTO mode by removing the key.

Troubleshooting

<RIA specification>

When the switch is set in the AUTO mode position, turning on the teach pendant enable switch stops the robot and causes an error message to appear. To release the error, set the teach pendant enable switch to off, then press the RESET key.

Three–mode switch and program operation

The following table lists the relationships among the three–mode switch setting, protective fence status (*FENCE signal), teach pendant (TP) enabled/disabled, deadman switch setting, *SFSPD signal status, and program–specified robot operation speed.

Relationships between three–mode switch settings and program operations [standard (domestic) specification]

Three–mode switch	Protective fence(*1)	*SFSPD	TP enabled/ disabled	TP deadman	Robot status	Units that can be started	Program–specified operation speed
AUTO	Open	ON	Enabled	Gripped	Emergency stop (fence open)		
				Released	Emergency stop (deadman, fence open)		
			Disabled	Gripped	Emergency stop (fence open)		
				Released	Emergency stop (fence open)		
	Closed	ON	Enabled	Gripped	Operable	TP only	Programmed speed
				Released	Emergency stop (deadman)		
			Disabled	Gripped	Operable	External start(*2)	Programmed speed
				Released	Operable	External start(*2)	Programmed speed
T1	Open	ON	Enabled	Gripped	Operable	TP only	T1 speed
				Released	Emergency stop (deadman)		
			Disabled	Gripped	Alarm and stop (T1/T2 and TP disabled)		
				Released	Emergency stop (T1/T2 and TP disabled)		
	Closed	ON	Enabled	Gripped	Operable	TP only	T1 speed
				Released	Emergency stop (deadman)		
			Disabled	Gripped	Alarm and stop (T1/T2 and TP disabled)		
				Released	Emergency stop (T1/T2 and TP disabled)		
T2	Open	ON(*4)	Enabled	Gripped	Operable	TP only	Programmed speed(*3)
				Released	Emergency stop (deadman)		
			Disabled	Gripped	Alarm and stop (T1/T2 and TP disabled)		
				Released	Emergency stop (T1/T2 and TP disabled)		
	Closed	ON	Enabled	Gripped	Operable	TP only	Programmed speed
				Released	Emergency stop (deadman)		
			Disabled	Gripped	Alarm and stop (T1/T2 and TP disabled)		
				Released	Emergency stop (T1/T2 and TP disabled)		

*1 Protective fence status
Open: *FENCE is off.
Closed: *FENCE is on.

*2 External speed
Remote mode: Program start on the line control panel
Local mode: Start button on the robot operation panel

*3 When the three–mode switch is in the T1 position and the fence is open, if you want to clamp a program–specified speed with the SFSPD override value, configure the system in such a way that the *SFSPD mentioned at *4 becomes off.

NOTE SFSPD override: When the program is executed with *SFSPD turned off, the override value is limited to within a value specified in $SCR.$SFRUNOVLIM (default value: 30%).

Relationships between three–mode switch settings and program operations [CE specification]

Three–mode switch	Protective fence(*1)	*SFSPD	TP enabled/ disabled	TP deadman	Robot status	Units that can be started	Program–specified operation speed
AUTO	Open	ON	Enabled	Gripped	Emergency stop (fence open)		
				Released	Emergency stop (deadman, fence open)		
			Disabled	Gripped	Emergency stop (fence open)		
				Released	Emergency stop (fence open)		
	Closed	ON	Enabled	Gripped	Operable	TP only	Programmed speed
				Released	Alarm and stop (deadman)		
			Disabled	Gripped	Operable	External start(*2)	Programmed speed
				Released	Operable	External start(*2)	Programmed speed
T1	Open	ON	Enabled	Gripped	Operable	TP only	T1 speed
				Released	Emergency stop (deadman)		
			Disabled	Gripped	Emergency stop (T1/T2 and TP disabled)		
				Released	Emergency stop (T1/T2 and TP disabled)		
	Closed	ON	Enabled	Gripped	Operable	TP only	T1 speed
				Released	Emergency stop (deadman)		
			Disabled	Gripped	Emergency stop (T1/T2 and TP disabled)		
				Released	Emergency stop (T1/T2 and TP disabled)		
T2	Open	ON(*4)	Enabled	Gripped	Operable	TP only	Programmed speed(*3)
				Released	Emergency stop (deadman)		
			Disabled	Gripped	Emergency stop (T1/T2 and TP disabled)		
				Released	Emergency stop (T1/T2 and TP disabled)		
	Closed	ON	Enabled	Gripped	Operable	TP only	Programmed speed
				Released	Emergency stop (deadman)		
			Disabled	Gripped	Emergency stop (T1/T2 and TP disabled)		
				Released	Emergency stop (T1/T2 and TP disabled)		

*1 Protective fence status
Open: *FENCE is off.
Closed: *FENCE is on.

*2 External speed
Remote mode: Program start on the line control panel
Local mode: Start button on the robot operation panel

*3 When the three–mode switch is in the T1 position and the fence is open, if you want to clamp a program–specified speed with the SFSPD override value, configure the system in such a way that the *SFSPD mentioned at *4 becomes off.

NOTE SFSPD override: When the program is executed with *SFSPD turned off, the override value is limited to within a value specified in $SCR.$SFRUNOVLIM (default value: 30%).

Relationships between three–mode switch settings and program operations [RIA specification]

Three–mode switch	Protective fence(*1)	*SFSPD	TP enabled/ disabled	TP deadman	Robot status	Units that can be started	Program–specified operation speed
AUTO	Open	ON	Enabled	Gripped	Emergency stop (fence open)		
AUTO	Open	ON	Enabled	Released	Emergency stop (deadman, fence open)		
AUTO	Open	ON	Disabled	Gripped	Emergency stop (fence open)		
AUTO	Open	ON	Disabled	Released	Emergency stop (fence open)		
AUTO	Closed	ON	Enabled	Gripped	Alarm and stop (AUTO and TP enable)		
AUTO	Closed	ON	Enabled	Released	Alarm and stop (deadman)		
AUTO	Closed	ON	Disabled	Gripped	Operable	External start(*2)	Programmed speed
AUTO	Closed	ON	Disabled	Released	Operable	External start(*2)	Programmed speed
T1	Open	ON	Enabled	Gripped	Operable	TP only	T1 speed
T1	Open	ON	Enabled	Released	Emergency stop (deadman)		
T1	Open	ON	Disabled	Gripped	Emergency stop (T1/T2 and TP disabled)		
T1	Open	ON	Disabled	Released	Emergency stop (T1/T2 and TP disabled)		
T1	Closed	ON	Enabled	Gripped	Operable	TP only	T1 speed
T1	Closed	ON	Enabled	Released	Emergency stop (deadman)		
T1	Closed	ON	Disabled	Gripped	Emergency stop (T1/T2 and TP disabled)		
T1	Closed	ON	Disabled	Released	Emergency stop (T1/T2 and TP disabled)		
T2	Open	ON(*4)	Enabled	Gripped	Operable	TP only	Programmed speed(*3)
T2	Open	ON(*4)	Enabled	Released	Emergency stop (deadman)		
T2	Open	ON(*4)	Disabled	Gripped	Emergency stop (T1/T2 and TP disabled)		
T2	Open	ON(*4)	Disabled	Released	Emergency stop (T1/T2 and TP disabled)		
T2	Closed	ON	Enabled	Gripped	Operable	TP only	Programmed speed
T2	Closed	ON	Enabled	Released	Emergency stop (deadman)		
T2	Closed	ON	Disabled	Gripped	Emergency stop (T1/T2 and TP disabled)		
T2	Closed	ON	Disabled	Released	Emergency stop (T1/T2 and TP disabled)		

*1 Protective fence status
Open: *FENCE is off.
Closed: *FENCE is on.

*2 External speed
Remote mode: Program start on the line control panel
Local mode: Start button on the robot operation panel

*3 When the three–mode switch is in the T1 position and the fence is open, if you want to clamp a program–specified speed with the SFSPD override value, configure the system in such a way that the *SFSPD mentioned at *4 becomes off.

NOTE SFSPD override: When the program is executed with *SFSPD turned off, the override value is limited to within a value specified in $SCR.$SFRUNOVLIM (default value: 30%).

5.2.3 Moving the robot by jog feed

The robot moves by jog feed when the jog keys on the teach pendant are pressed. The robot must be moved to a target position when motion instructions are specified in the program.

Jog feed depends on the following two factors:

- Feedrate override: Robot motion speed (jog feedrate)
- Manual–feed coordinate system: Coordinate system for robot motion (jog feed type)

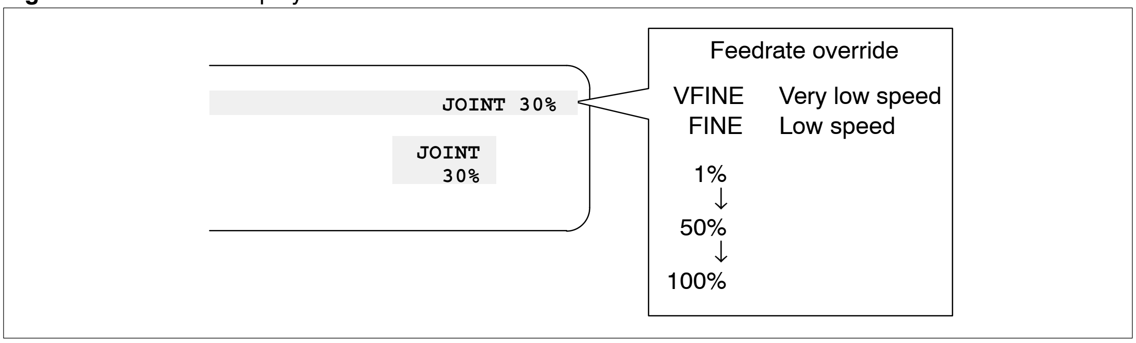
Feedrate override

A feedrate override is one of the two factors on which jog feed depends. The feedrate override is represented in percentage (%).

The current feedrate override is displayed at the upper right corner of the screen of the teach pendant.

Pressing the feedrate override key displays a pop up window in reverse video at the upper right of the screen to call the user's attention. The popup window in reverse video automatically disappears after a few seconds or when another key is pressed.

Figure 5–5. Screen Display for Feedrate Override



Feedrate override 100% means that the robot moves at the maximum feedrate. The step feed–rate of FINE is specified by a system variable, $JOG_GROUP.$FINE_DIST in linear jog.(Standard : 0.1mm). In standard setting,each axis rotates at 0.001deg per step. The step width of VFINE is one–tenth of that of FINE.

NOTE If VFINE or FINE is used as the current speed override, the robot makes a motion of a single step at a time. To resume the robot motion, release and press the jog key.

Table 5–2 shows the change in feedrate override when the override key is pressed.

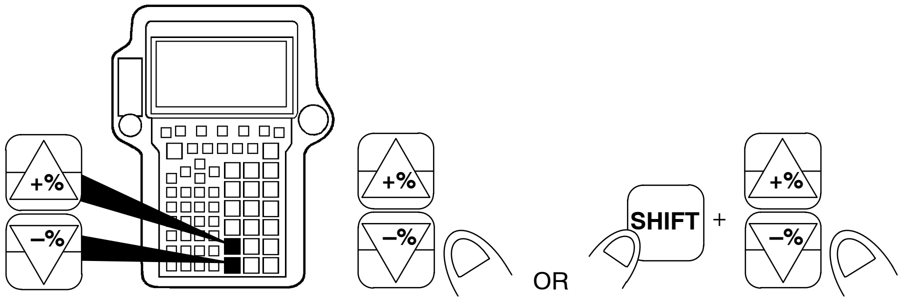
Table 5–2. Feedrate Override

When the override key is pressed	VFINE → FINE → 1% → 5% → 50% → 100% In 1% increments (1% → 5%), In 5% increments (5% → 50%)
When the override key is pressed while pressing the SHIFT key(*1)	VFINE → FINE → 5% → 50% → 100%

*1 Enabled only when $SHFTOV_ENB is 1

To change the feedrate override, press the override key. Whenever the override key is pressed while the shift key is pressed, the feedrate changes sequentially in the order: FINE, VFINE, 5%, 50%, and 100%. However, the feedrate is changed in this way only when system variable $S HFTOV_ENB = 1.

Figure 5–6. Override Keys



A feedrate override must be determined according to the condition of the machining cell, type of the robot motion, or the skill of the operator. Therefore, an inexperienced robot operator should use a low feedrate override.

NOTE When the override key is pressed, a window indicating the manual feed coordinate system and speed override appears on the screen in reverse video. Pressing the override key again enables you to change the override value.
If the override key is not pressed, the window closes automatically.

This window is automatically closed if the override keys are not pressed for a while.

When the safe speed signal (*SFSPD input) (→ Section 3.1) is turned off, the speed override is reduced to the value of $SCR.$FENCEOVRD. In this state, the speed override cannot be increased beyond the upper limit specified by $SCR.$SFJOGOVLIM (→ Section 3.1).

A function is available which restores the speed override when the safety fence is closed (→ Section 3.14).

Jog feedrate

A jog feedrate is a speed at which the robot moves during jog feed. The jog feedrate is obtained by the following expression: If the following value exceeds the speed limit 250 mm/sec for the T1 or T2 mode described above, the operation speed is clamped at the one described earlier.

Figure 5–7. Jog Feedrate

Jog feedrate (joint feed) =

$$\text{Maximum joint feedrate} \times \frac{\text{Each axis jog override}}{100} \times \frac{\text{Feedrate override}}{100}$$

Jog feedrate (linear feed) (mm/sec) =

$$\text{Maximum linear feedrate} \times \frac{\text{Jog override}}{100} \times \frac{\text{Feedrate override}}{100}$$

Jog feedrate (Circular feed) (mm/sec) =

$$\text{Maximum circular feedrate} \times \frac{\text{Jog override}}{100} \times \frac{\text{Feedrate override}}{100}$$

Each axis jog override	$SCR_GRP.$JOGLIM_JNT[i] (%)
Jog override	$SCR.$JOGLIM (%)
Maximum joint feedrate	$PARAM_GROUP.$JNTVELLIM (deg/sec)
Maximum linear feedrate	$PARAM_GROUP.$SPEEDLIM (mm/sec)
Maximum circular feedrate	$PARAM_GROUP.$ROTSPEEDLIM (deg/sec)

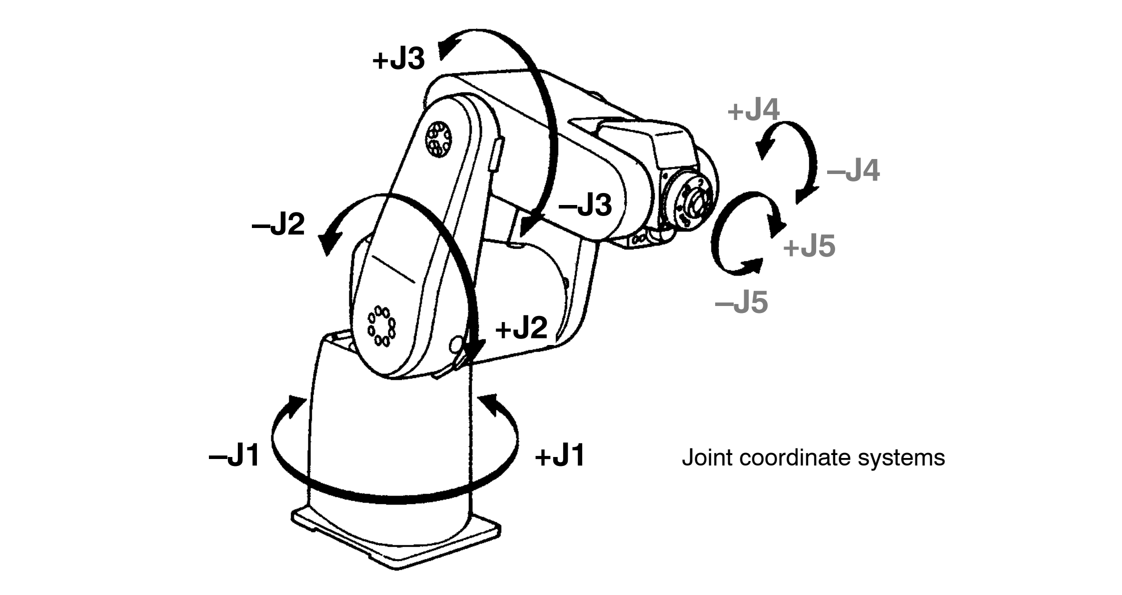
Manual–feed coordinate systems (Jog type)

Manual–feed coordinate systems determine how the robot moves during jog feed. The manual–feed coordinate systems are classified into three types:

Joint jog (JOINT)

During joint jog, the robot moves independently around each axis according to each joint coordinate system. See Section 3.8 for the joint coordinate systems.

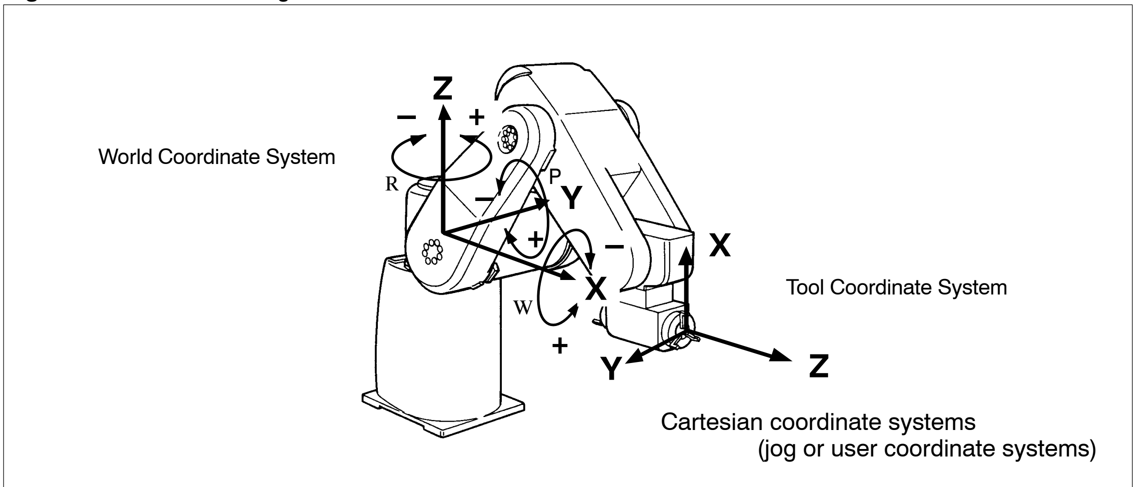
Figure 5–8. Joint Jog



Cartesian jog (XYZ)

During Cartesian jog, the tool center point of the robot moves along the X–, Y–, and Z–axes of the user or jog coordinate systems. You can not cause the robot to rotate the tool around x–,y–,and z–axis of the user frame or jog frame.(See Section 3.8.2, "Setting a user coordinate system", and Section 3.8.3, "Setting a jog coordinate system")

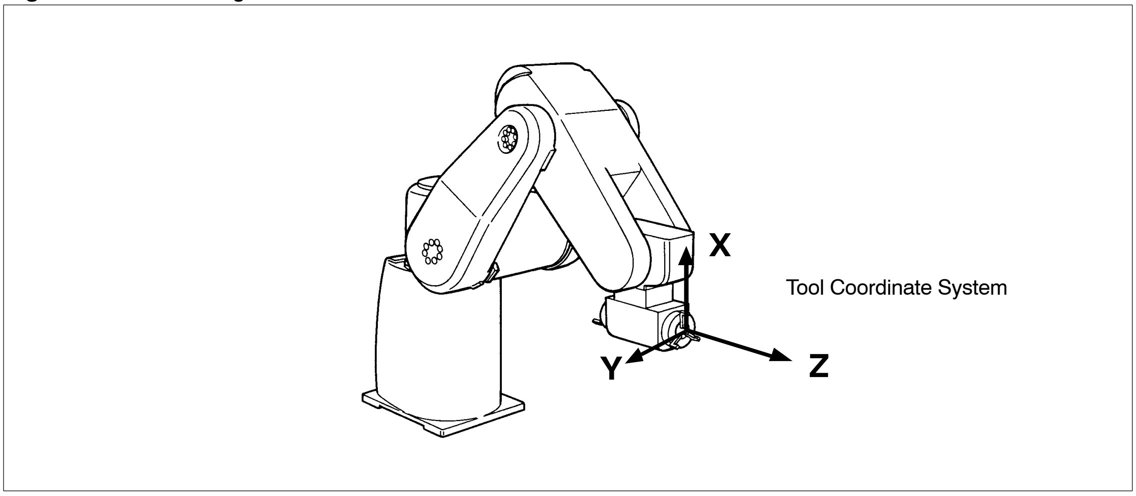
Figure 5–9. Cartesian Jog



Tool jog (TOOL)

During tool jog, the tool center point (TCP) moves along the X–, Y–, and Z–axes of the tool coordinate system defined for the wrist of the robot. You can not cause the robot to rotate the tool around x–,y–,and z–axis of the tool frame.(See Section 3.8.1, "Setting a tool coordinate")

Figure 5–10. Tool Jog

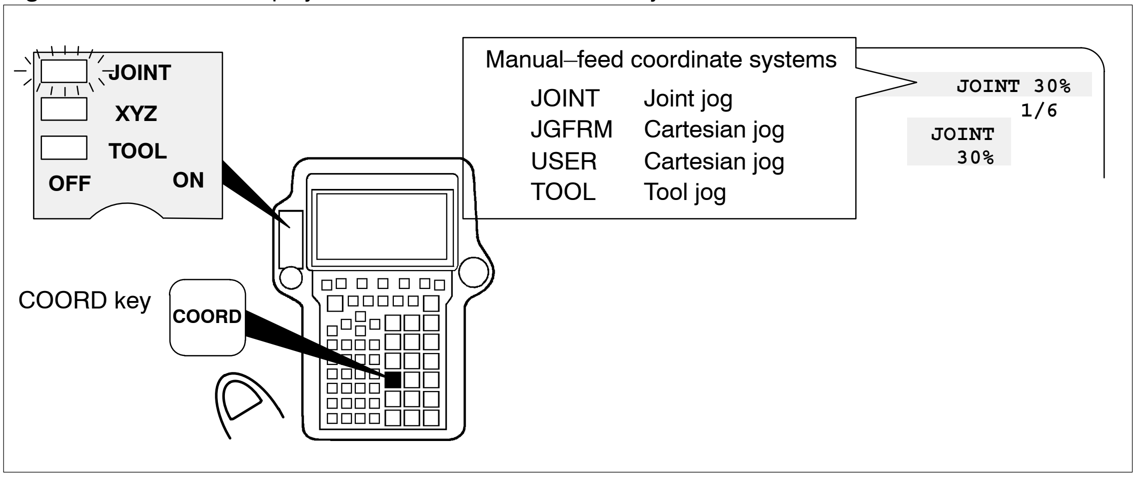


Selecting a manual–feed coordinate system

The current manual–feed coordinate system is displayed at the upper right corner of the screen of the teach pendant.

Pressing the COORD key displays a popup menu in reverse video at the upper right of the screen to call the user's attention. The popup menu in reverse video automatically disappear after a few seconds or when another key is pressed.

Figure 5–11. Screen Display for Manual–Feed Coordinate Systems



Whenever the COORD key on the teach pendant is pressed, the selected manual–feed coordinate system change cyclically. When a manual–feed coordinate system changes sequentially in the order shown in Table 5.2. is selected, its corresponding LED lights.

Table 5–3. Jog type Selection Sequence

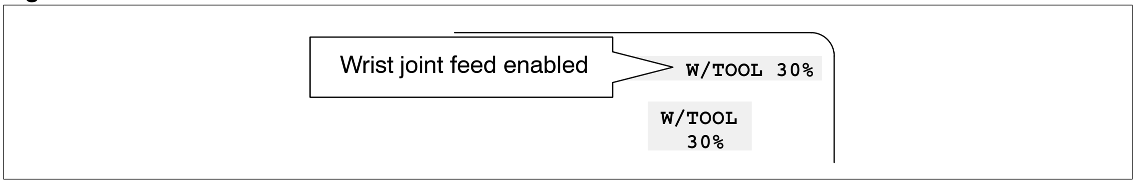
Screen display	JOINT → JGFRM → TOOL → USER → JOINT
LED state	JOINT LED on → XYZ LED on → TOOL LED on → XYZ LED on → JOINT LED on

Enabling a wrist joint feed

In wrist joint feed, the attitude of the tool is not held during linear feed (Cartesian jog feed) or circular feed (tool jog feed).

- When wrist joint feed is disabled, the attitude of the tool is held during jog feed. (Standard setting)
- When wrist joint feed is enabled, the attitude of the tool is not held during jog feed. In this case, [W/] is displayed on the screen.
  - In linear feed (linear motion along the axes of the Cartesian coordinate system), the tool center point moves linearly while the wrist joint is fixed.

Figure 5–12. Indication that Wrist Joint Feed Is Enabled



NOTE When the motion instruction for linear or circular motion under path control is executed, wrist joint feed has the same function as the wrist joint motion additional instruction (WRIST JOINT).

Switching to additional axes

In addition to the standard robot axes (usually 5 axis) in one operation group, up to three additional axes can be controlled as a subgroup.

NOTE The user can switch to a subgroup by using the auxiliary menu or jog menu described below.

Jog menu

With the jog menu function, the following data related to jog operation can be displayed or updated easily:

- Tool, jog, or user coordinate system number currently selected
- Group number currently selected
- Subgroup selection state (robot or additional axes)

To display the jog menu, press the manual feed coordinate system key while holding down SHIFT key.

```
TEST
UTILITIES Hints                  TOOL 100%
                           Tool (.=10) 2
                           Jog         3
                           User        1
                           Group       2
                           Robot/Ext
```

Table 5–4. Operation Procedure Using the Jog Menu

Operation	Procedure
Opening the menu	Press the manual feed COORD key while holding down SHIFT key.
Closing the menu	● Press the manual feed COORD key while holding down SHIFT key. ● Press PREV key. ● Value modification using numerical key (See the descriptions of coordinate system number change and group switching.)
Moving the cursor	cursor key
Changing the coordinate system number	● Tool coordinate system, jog coordinate system 1 to 10 (Put the ”.” key to select 10.) ● User coordinate system 0 to 9
Group switching (for a multi–group system only)	Numeric key (valid for existing group numbers only)
Subgroup switching (for a system with a subgroup)	After moving the cursor to the line containing Robot/Ext, switch between Robot and Ext by using the left/right cursor key. (The position of reverse video switches.)

WARNING

Be sure to remember the current coordinate system number/group number. Otherwise, in such a case, a robot may move in an unexpected direction at jog time, or a robot of an unexpected group may move, thus leading to a fatal accident.

WARNING

After coordinate system number/group number switching, be sure to close the jog menu. If the jog menu is left open, the operator may change the coordinate system number or group number by touching a numeric key of the teach pendant unconsciously. In such a case, a robot may move in an unexpected direction at jog time, or a robot of an unexpected group may move, thus leading to a fatal accident.

Procedure 5–1 Moving the robot by jog feed

Condition ■ Do not enter the operating area. Do not put any obstacles within the work area.

> **CAUTION**
> Before you jog the robot be sure that all safety requirements for the work area are satisfied. Otherwise, injury or property damage could occur.

Step

1 Press the COORD key to display a desired manual–feed coordinate system on the teach pendant.

NOTE The feedrate override is automatically set to 10%.

2 Press the override key to adjust the jog feedrate displayed on the teach pendant.

3 Hold the teach pendant and press the deadman switch on the back of the teach pendant. Continue pressing the deadman switch during jog feed.

4 Turn on the teach pendant enable switch.

NOTE If the deadman switch is released when the teach pendant enable switch is on, an alarm occurs. To reset the alarm, press and hold down the deadman switch again, then press the RESET key on the teach pendant.

NOTE If the operator is not accustomed to the operation of the robot or is not sure about the robot motions, low feedrate overrides should be set.

> **CAUTION**
> The robot starts its motion in the next step. If the jog feed of the robot needs to be stopped in an emergency in order to avoid danger, the operator should release the deadman's switch or press the emergency stop button.

5 To move the robot by jog feed, press the jog key corresponding to the desired robot motion direction while pressing the SHIFT key. When the jog key is released, the robot stops.

NOTE When the override is FINE or VFINE,press the jog key and release it every time for each motion.

Switch to wrist joint feed

6 Press the FCTN key. The function menu is displayed.

7 Select 5,TOGGLE WRIST JOG. The mark,[W/],is displayed to show the wrist joint jog mode. To release this mode,select 5,TOGGLE WRIST JOG again.

```
3
4 TOGGLE WRIST JOG
5 SAVE
```

FCTN

```
SAMPLE1                        W/TOOL   30 %
                                         1/6
```

Switch to a extended axis

8 Press the FCTN key. The function menu is displayed.

9 Select 4,TOGGLE SUB GROUP. The jog control is switched from the robot standard axes to an extended axis. The control will be returned when it is done.

```
3 CHANGE GROUP
4 TOGGLE SUB GROUP
5 TOGGLE WRIST JOG
```

FCTN

```
SAMPLE1                  S     W/TOOL   30 %
                                         1/6
```

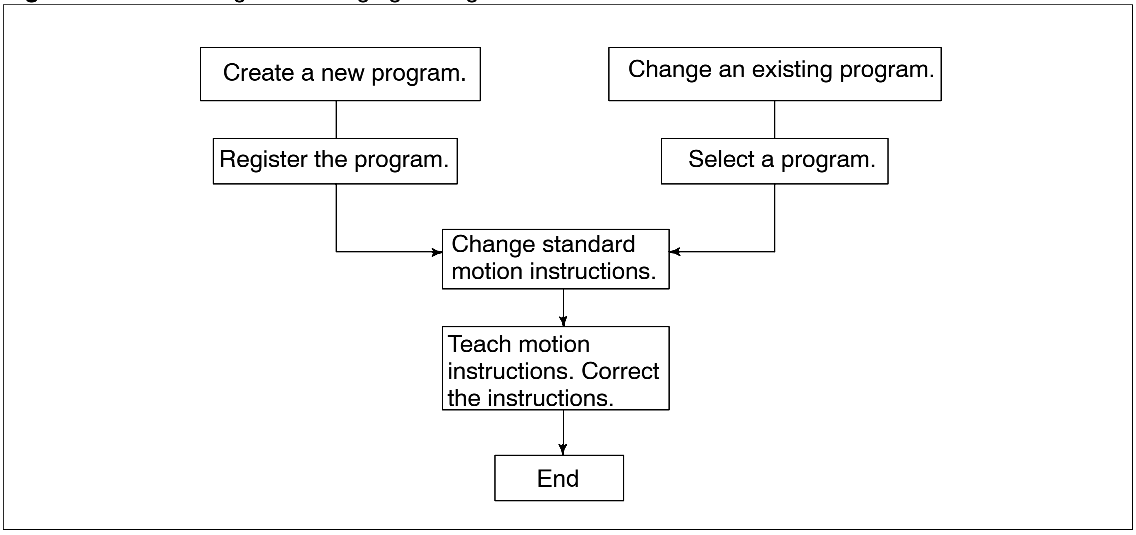
10 To terminate jog feed, turn off the teach pendant enable switch and release the deadman switch.

5.3 Creating a Program

To create a program, use the following procedure:

- Register a program and specify program information
- Modify standard instruction (standard motion instructions)
- Teach motion instructions
- Teach various control instructions including a palletizing instruction

Figure 5–13. Creating and Changing a Program



Registering a program

Create a null program with a new name.

Specifying program information

Specify the attributes of the program.

Changing standard motion instructions

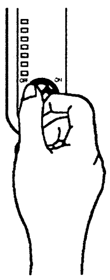
Respecify the standard instructions to be used when teaching motion instructions.

Teaching motion instructions

Teach a motion instruction and an supplementary motion instruction.

Teaching control instructions

Teach control instructions including a palletizing instruction.



Use the teach pendant to create a new program and correct an existing program. To do this, the teach pendant must be enabled beforehand.
To enable the teach pendant, satisfy the following condition:

■ The teach pendant enable switch must be turned on.

To prevent the program from being started by mistake,prohibit starting a program with a teach pendant while teaching.(See Figure 2–13, ”Function menu”)

5.3.1 Registering a program

Enter a program name and register the program. A program name consists of up to eight alphanumeric characters including symbols to discriminate program names from one another. For the program name, see Section 4.1.1.

Register a program on the program registration screen.

> **CAUTION**
> When a new program is made, the current program is halted.

Entering a program name

There are three methods for entering a program name:

- Words: Up to five words consisting of up to seven characters can be used as program names. Enter these reserved words, such as PRG, MAIN, SUB, and TEST, in $PGINP_WORD[1 TO 5] in advance (See Sectioin 3.14, ”System config menu”)
- Uppercase or lowercase alphabetic characters: Any letter of the alphabet can be specified for a program name. The alphabetic characters combined with any numeric characters and/or any symbols are used as the characters of a program name.

> **CAUTION**
> Asterisks (*) and at marks (@) should not be used in a program name.

Options

During optional settings, an overwrite or insert mode can be specified for character entry, or character string deletion.

- In the overwrite mode, entered characters are written over existing characters.
- In the insert mode, entered characters are inserted before the character pointed to by the cursor. In this case, all the characters to the right of the entered character(s) are shifted to the right. INSERT or OVRWPT is displayed on the screen.
- All the characters in the field where the cursor is positioned are deleted.

NOTE The program name should not begin with a numeral.

Setting program information

Set the following program information items on the program information screen. See Section 4.1.

- Program name
- Subtype
- Comments: Comments can be written in a program. Up to 16 alphanumeric characters and symbols, which can be used for a program name. In some cases, comments may not be entered.
- Group mask: Specifies a motion group to be controlled in a program.
- Write protection: Prevents a program from being changed.
- Interruption disable: Causes the program having no motion no to be paused by an alarm with a severity of WARN,PAUSE,STOP,and SERVO, the emergency stop, and HOLD. However, this setting is not applied to the alarm that is generated by the program. In this case the program is stopped.

Procedure 5–2 Registering a program

Condition ■ The teach pendant must be enabled.

Step

1 Press the MENUS key to display the screen menu.

2 Select SELECT.

Alternatively, the following program selection screen can also be displayed by pressing the SELECT key.

```
SELECT                                    JOINT 30%
                  61276 bytes free
 1  SAMPLE1           SAMPLE PROGRAM1
 2  SAMPLE2           SAMPLE PROGRAM2
 3  PROG001           PROGRAM001
 4  PROG002           PROGRAM002



[TYPE]  CREATE  DELETE  MONITOR  [ATTR]  >
 COPY   DETAIL  LOAD    SAVE      PRINT  >
```

3 Press the F2 [CREATE] key. The program registration screen is displayed.

```
                                       JOINT 30%
 1  Words
 2  Upper Case
 3  Lower Case
 4  Options            ---Insert---
SELECT

   ---Create Teach Pendant Program---
 Program Name      [            ]
 Sub type          [            ]
                               ---End---
Enter program name
PRG     MAIN    SUB    TEST
```

4 Select a method for entering a program name (words or alphabetic characters) using the cursor keys.

```
                                       JOINT 30%
 1  Words
 2  Upper Case
 3  Lower Case
 4  Options            ---Insert---
SELECT

   ---Create Teach Pendant Program---
 Program Name      [            ]

                               ---End---
Enter program name
abcdef  ghijkl  mnopqr  stuvwx  yz_@*.  >
```

5 Enter a program name by pressing the function keys corresponding to the characters in the program name. The function key menu displayed depends on the method selected in step 4.
With alphabetic character entry, for instance, press the function key corresponding to a desired character repeatedly until the character is displayed in the program name field; that is, if you want to enter P, press the F4 function key four times. Press the NEXT key to move the cursor to the right one character. Repeat this procedure until the program name is completely entered.

```
abcdef ghijkl mnopqr st
```

F4

```
Select                               JOINT 30%
                                          1/3
   ---Create Teach Pendant Program---
 Program name:      [S               ]



abcdef  ghijkl  mnopqr  stuvwx  yz_@*.  >
```

NOTE When creating a program using RSR or PNS for automatic operation, follow the rule below. Otherwise, the program does not run.

- A RSR program must be written as RSRnnnn, where nnnn represents a four–digit number. An example is RSR0001.
- A PNS program must be written as PNSnnnn, where nnnn represents a four–digit number. An example is PNS0001.

6 After entering a program name, press the ENTER key.

```
SELECT
  ---Create Teach Pendant
 Program Name: [SAMPLE3
```

ENTER

```
Select                               JOINT 30%
 1  Jobs             5
 2  Processes        6
 3  Macro            7
 4                   8
Select
   ---Create Teach Pendant Program---
 Program Name:      [SAMPLE3         ]
 Sub type           [                ]
                           ---End---
Select Sub type
```

7 To edit the registered program, press the F3 (EDIT) or ENTER key. The program edit screen for the registered program is displayed.

8 To enter program information, press the F2 (DETAIL) key (or the ENTER key). The program information screen is displayed.

```
Select function
      DETAIL   EDIT
```

F2

```
Program detail                    JOINT  30 %
                                        1/6
  Creation Date:               10-MAR-1998
  Modification Date:           11-MAR-1998
  Copy Source:          [****************]
  Positions: FALSE  Size:      312 Byte
   1  Program name:              [SAMPLE3 ]
   2  Sub Type:         [          Process]
   3  Comment:          [SAMPLE PROGRAM 3]
   4  Group Mask:              [1,*,*,*,*]
   5  Write protect:            [     OFF]
   6  Ignore pause:             [     OFF]

     END  PREV  NEXT
```

9 Specify the following program information items:

- To change a program name, move the cursor to the setting field, change the program name, then press the ENTER key.
- To change a subtype (see Section 4.1.3), press the F4 [CHOICE] key to display a subtype menu. Then, select None, Job, Process, or Macro. JOB or PROCESS can be selected only when system variable $JOBPROC_ENB is set to 1.

- To enter comments, move the cursor to the setting field, enter the comments, then press the ENTER key (see Subsection 4.1.2).
- To specify a group mask, move the cursor to the setting field and select 1, *. The specified motion group is controlled (see Section 4.1.4). For safety, specify (*, *, *, *, *) for programs which do not contain any motion instructions.

> **CAUTION**
> You cannot change the motion group of a program that contain operation instructions cannot be changed.

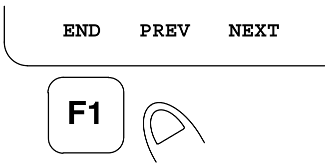
NOTE If the system used does not have the multi–group setting, only either of the following settings is allowed: The first group is set as 1; An asterisk (*) indicating no group is set.

- To specify write protection, move the cursor to the setting field and select ON or OFF (see Subsection 4.1.5).
- To specify interruption disable, move the cursor to the setting field and press the function key (ON or OFF) (see Section 4.1.6). Select ON for programs not to be halted when an alarm occurs such as macro instructions or automatic start programs.

NOTE To return to the list screen, press the PREV key repeatedly until the list screen is displayed.

```
Program detail                JOINT  30 %
                                     1/6
 Creation Date:           10-MAR-1998
 Modification Date:       11-MAR-1998
 Copy Source:         [****************]
 Positions: FALSE  Size:      312 Byte
  1  Program name:         [SAMPLE3 ]
  2  Sub Type:        [          Process]
  3  Comment:         [SAMPLE PROGRAM 3]
  4  Group Mask:           [1,*,*,*,*]
  5  Write protect:         [      OFF]
  6  Ignore pause:          [      OFF]

   END  PREV  NEXT
```

10 After entering the program information items, press the F1 (END) key. The program edit screen for the registered program is displayed.




```
SAMPLE3                          JOINT 30%
                                      1/1
 [End]



POINT                            TOUCHUP >
```

5.3.2 Changing a standard motion instruction

For specification of a move statement, many items including move type, move speed, and positioning type need to be set. For convenience, the user can register frequently used move instructions as standard move statements.

To modify a standard operation statement, first press the F1 key. A list of standard operation statements appears. Press the F key again. The screen for editing the standard operation statements appears.

```
POINT   SINGLE   DUAL   BACKUP   TOUCHUP >
```

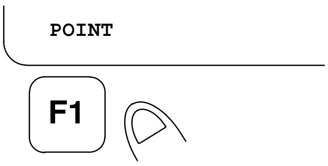
- Press F1 POINT to list the standard operation statements.

Procedure 5–3 Changing a standard motion instruction

Condition
- The program edit screen must be selected.
- The teach pendant must be enabled.

```
SAMPLE3                                JOINT 30%
                                            1/1
 [End]




POINT                                  TOUCHUP >
```

Step 1 Press the F1 POINT key. The standard motion instruction menu is displayed.




```
Joint default menu                     JOINT 30%
1  J  P[ ] 100% FINE
2  J  P[ ] 100% CNT100
3  L  P[ ] 1000cm/min FINE
4  L  P[ ] 1000cm/min CNT100
SAMPLE3
                                            1/1
[End]

ED_DEF                                 TOUCHUP >
```

NOTE If the instructions listed on the submenu are necessary, they need not be changed.

2 To change a standard motion instruction, press the F1 ED_DEF.


```
Default Motion                         JOINT 30%
                                            1/4
  1  J  P[ ] 100% FINE
  2  J  P[ ] 100% FINE
  3  L  P[ ] 1000cm/min CNT50
  4  L  P[ ] 1000cm/min CNT50



                                          DONE >
```

3 Move the cursor to the instruction item to be changed (motion type, feedrate, positioning type, or supplementary motion instruction) using the cursor keys.

```
Default Motion                     JOINT 30%
                                        2/4
  1  J  P[ ] 100% FINE
  2  J  P[ ] 100% FINE
  3  L  P[ ] 1000cm/min CNT50
  4  L  P[ ] 1000cm/min CNT50


Enter value
                        [CHOICE]    DONE >
```

4 Select numeric keys and function keys to correct the instruction item.

To change the feedrate, for instance, move the cursor to feedrate. Enter a new value with numeric keys, then press the ENTER key.

7 0

```
Old Value: 100
```

ENTER

```
Default Motion                     JOINT 30%
                                        2/4
  1  J  P[ ] 100% FINE
  2  J  P[ ] 70% FINE
  3  L  P[ ] 1000cm/min CNT50
  4  L  P[ ] 1000cm/min CNT50


Enter value
                                    DONE >
```

5 When [CHOICE] is displayed in the F4 key name field, press the F4 key. Then, an option of another instruction item can be selected from the submenu.

```
[CHOICE]
```

F4

```
Motion Modify                      JOINT 30%
1 Fine                    5
2 Cnt                     6
3                         7
4                         8
Default Motion
                                        2/4
  1  J  P[ ] 100% FINE
  2  J  P[ ] 70% FINE
  3  L  P[ ] 1000cm/min CNT50
  4  L  P[ ] 1000cm/min CNT50

Select item
                        [CHOICE]    DONE >
```

```
Motion Modify
  1 FINE
  2 CNT
  3
  4
Default Motion
```

ENTER

5 0 ENTER

```
Default Motion                     JOINT 30%
                                        4/4
  1  J  P[ ] 100% FINE
  2  J  P[ ] 70% CNT50
  3  L  P[ ] 1000cm/min CNT50
  4  L  P[ ] 1000cm/min CNT50


Enter value
                        [CHOICE]    DONE >
```

6 Repeat steps 3 to 5 for each instruction to be changed.

7 After teaching is completed, press the F5 (DONE) key.

```
DONE  >
```

F5

5.3.3 Teaching a motion instruction

A motion instruction moves the robot to the specified position in the work area at the specified feedrate using the specified movement method. When the motion instruction is taught, the instruction items of the motion instruction and position data are simultaneously taught.

The instruction items of a motion instruction are as follows (see Section 4.3 for the motion instruction):

- Motion type: Controls a path to the specified position. (joint, linear, circular)
- Position variable: Stores data on positions to which the robot moves.
- Feedrate: Specifies the speed of the robot when it moves.
- Positioning type: Specifies whether positioning is performed at the specified position.
- Supplementary motion instruction: Specifies the instruction which executes with the loader robot.

Teaching a motion instruction is selected after a standard motion instruction is created. In this case, the current position (position data) is stored in the position variable.

- Press the F1, F2, F3, or F4 key to list the stored standard statements. Choose a desired statement from the list, and then program that statement.
- To program a single standard statement repeatedly, hold down the shift key and press the F1, F2, F3, or F4 key.

```
POINT   SINGLE   DUAL   BACKUP   TOUCHUP >
```

  - Press F1 POINT to list the standard operation statements.
- Check whether the position to be programmed is one of the robot's singular points (for singular points, see Position data in 4.3.2). The user can program the position by using the axial method, if so desired. (see Singular point check functions in 5.7)

Procedure 5–4 Teaching a motion instruction

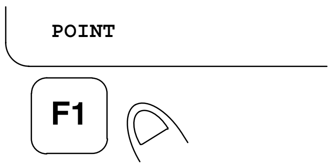
Step

1 Move the robot to the desired position in the work area by jog feed.

2 Move the cursor to END.

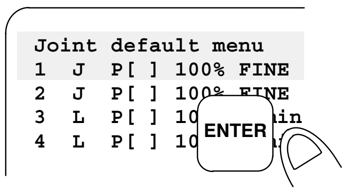
```
SAMPLE1                              JOINT 30%
                                           1/1
[End]



POINT                                TOUCHUP >
```

3 Press the F1 [POINT] key to display the standard motion instruction menu.




```
Joint default menu                   JOINT 30%
1  J  P[ ] 100% FINE
2  J  P[ ] 100% FINE
3  L  P[ ] 1000cm/min CNT50
4  L  P[ ] 1000cm/min CNT50
SAMPLE3
                                           1/1
[End]

ED_DEF                               TOUCHUP >
```

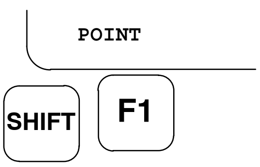
4 Select the standard motion instruction to be taught, press the ENTER key, and specify the desired motion instruction. At the same time the position is taught.




```
SAMPLE1                              JOINT 30%
                                           2/2
1: J  P[1] 100% FINE
[End]


Position has been recorded to P[1].

POINT                                TOUCHUP >
```

5 Repeat steps 2 to 4 for each motion instruction to be specified in the program.

6 To specify the same standard motion instruction repeatedly, press the F1 [POINT] key while pressing the SHIFT key. This adds the previously specified motion instruction to the currently selected standard motion instruction.




```
SAMPLE1                              JOINT 30%
                                           3/3
1: J  P[1] 100% FINE
2: J  P[2] 100% FINE
[End]

Position has been recorded to P[2].
POINT                                TOUCHUP >
```

5.3.4 Teaching an supplementary motion instruction

The supplementary motion instruction makes the robot do special work while it is moving according to the motion instruction. Some of the following supplementary motion instructions are provided (see Section 4.3.5 for the supplementary motion instructions):

- Wrist joint motion instruction
- Acceleration/deceleration override instruction
- Skip instruction
- Position compensation instruction
- Direct position compensation instruction
- Tool offset instruction
- Direct tool offset instruction
- Incremental instruction
- Path instruction
- Soft float
- Asynchronous additional speed
- Synchronous additional speed
- Pre–execution
- Post–execution

To teach an supplementary motion instruction, place the cursor behind the motion instruction and press the F4 [CHOICE] key to display the supplementary motion instruction menu. Select an supplementary motion instruction from the menu. (See Appendix A.3 for the program instruction menu.)

```
                JOINT 30%
                     4/5
 500mm/sec CNT10

           [CHOICE]
```

F4

```
Motion modify                    JOINT  30 %
  1 No option        5 Offset
  2 Wrist Joint      6 Offset,PR[  ]
  3 ACC              7 Incremental
  4 Skip,LBL[]       8 ---next page---
PROGRAM1
```

NOTE The available supplementary motion instructions vary according to your software configuration.

Procedure 5–5 Teaching the supplementary motion instruction

Step

1 Place the cursor immediately behind the motion instruction.

```
PROGRAM1                                JOINT 30%
                                              4/5
4: L  P[3] 500mm/sec CNT10
[End]

                              [CHOICE]
```

2 Press the F4 [CHOICE] key. The supplementary motion instruction menu is displayed.

```
              JOINT 30%
                   4/5
500mm/sec CNT10

         [CHOICE]
```


```
Motion modify                      JOINT  30 %
  1 No option           5 Offset
  2 Wrist Joint         6 Offset,PR[  ]
  3 ACC                 7 Incremental
  4 Skip,LBL[]          8 ---next page---

                                          4/5
   4:J P[3] 100% FINE
[End]
                              [CHOICE]
```

3 Select a desired item.

For example, the following screen teaches a acceleration override instruction.

```
Motion Modify
  1 No option        5
  2 Wrist Joint      6
  3 ACC              7
  4 Skip,LBL[ ]      8
PROGRAM1
```

```
PROGRAM1                                JOINT 30%
                                              4/5
 4: L  P[3] 500mm/sec CNT10
  : ACC 150
[End]

                              [CHOICE]
```

For details of the instructions, see Chapter 4.

Procedure 5–6 Teaching the incremental instruction

Step

1 Move the cursor to the space at the end of the motion instruction. The teaching incremental instruction is shown as follow.

```
SAMPLE1                              JOINT  30 %
                                           4/5
   4:J P[3] 100% FINE
[End]
                          [CHOICE]
```

```
                JOINT 30%
                     4/5
500mm/sec CNT10

          [CHOICE]
```

F4

```
Motion modify                        JOINT  30 %
 1 No option           5 Offset
 2 Wrist Joint         6 Offset,PR[  ]
 3 ACC                 7 Incremental
 4 Skip,LBL[]          8 ---next page---
PROGRAM1
                                          4/5
   4:J P[3] 100% FINE
```

```
SAMPLE1                              JOINT  30 %
                                           4/5
   4:J P[3] 100% FINE INC
[End]
                          [CHOICE]
```

CAUTION

Teaching the incremental instruction makes the position data have no position information.
Enter the incremental amount to the position data manually.

2 Enter the incremental amount directly to the position data.

```
SAMPLE1                              JOINT  30 %
                                           4/5
   4:J P[3] 100% FINE INC
[End]
                          [CHOICE] POSITION
```

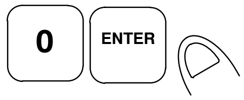
```
      [CHOICE] POSITION
```

F5

```
Position Detail
 P[3]  GP:1 UF:0 UT:1    CONF:N 00
 X    *******  mm  W    *******  deg
 Y    *******  mm  P    *******  deg
 Z    *******  mm  R    *******  deg
SAMPLE1
                                          4/5
   4:J P[3] 100% FINE INC
[End]
Enter value
          PAGE   CONFIG   DONE   [REPRE]
```

3 Enter the incremental amount directly.

```
P       0.000  deg
R     *******  deg
```




```
  Position Detail
   P[3]   GP:1 UF:0 UT:1   CONF:N 00
   X     500.000   mm  W       0.000  deg
   Y     100.000   mm  P       0.000  deg
   Z     100.000   mm  R       0.000  deg
  SAMPLE1
                                          4/5
     4:J P[3] 100% FINE INC
  [End]
            PAGE    CONFIG    DONE  [REPRE]
```

4 When you are fished to enter the position data,press F4,DONE.

```
CONFIG   DONE  [REPRE]
```


```
  SAMPLE1                       JOINT  30 %
                                       4/5
     4:J P[3] 100% FINE INC
  [End]
  Enter value or press ENTER
                          [CHOICE] POSITION
```

5.3.5 Teaching a control instruction

A control instruction is a program instruction for the R–J3 controller that is not a motion instruction. The control instructions are as follows:

- Palletizing instruction
- Register instruction
- Position register instruction
- Soft float instruction
- I/O (input/output) instruction
- Branch instruction
- Wait instruction
- Macro instruction
- Program end instruction
- Comment instruction
- Supplementary motion instruction
- Other instructions

To teach a control instruction, first press the F1 (INST) key to display the submenu. Then, select a desired control instruction item from the menu (see Appendix A.3 for the program instructions menu).

```
Instruction                      JOINT 30%
1 Registers          5 JMP/LBL
2 I/O                6 CALL
3 IF/SELECT          7 Palletizing
4 WAIT               8 ---next page---
PROGRAM
```

```
Instruction                      JOINT  30 %
 1 Miscellaneous     5 MACRO
 2 Program control   6
 3 Skip              7
 4 Offset            8 ---next page---
PROGRAM
```

NOTE The program instructions vary according to you software configuration.

Procedure 5–7 Teaching a register instruction

Condition
- ■ The teach pendant must be enabled.
- ■ The program edit screen must be selected.

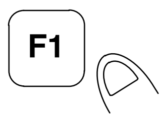
```
PROGRAM1                         JOINT 30%
                                      2/2
 1: J  P[1] 100% FINE
[End]



[INST]                           [EDCMD] >
```

Step

1 Move the cursor to END.

2 Press the F1 (INST) key. Then, the control instruction menu is displayed.

[INST]




```
Instruction                      JOINT 30%
1 Registers          5 JMP/LBL
2 I/O                6 CALL
3 IF/SELECT          7 Palletizing
4 WAIT               8 ---next page---
PROGRAM
```

3 To teach a register instruction, select REGISTERS.
The following screens indicate that the value of register [1] is increased by one.

```
Instruction
1 Registers       5
2 I/O             6
3 IF/SELECT       7
4 WAIT            8
ENTER
```

```
REGISTER statement              JOINT 30%
1 ...=...           5 ...=.../...
2 ...=...+...       6 ...=...DIV...
3 ...=...-...       7 ...=...MOV...
4 ...=...*...       8
PROGRAM1
```

```
REGISTER statement              JOINT 30%
1 R[ ]              5
2 PL[ ]             6
3 PR[ ]             7
4 PR[i,j]           8
PROGRAM1
                                      2/3
2: ...=...+...
[End]
```

```
REGISTER statement              JOINT  30 %
1 R[ ]              5 RO[  ]
2 Constant          6 RI[  ]
3 DO[  ]            7 GO[  ]
4 DI[  ]            8 ---next page---
PRG1
                                      2/2
   2:  R[1]=...+...
[End]
```

```
REGISTER statement              JOINT 30%
1 R[ ]              5 RO[  ]
2 Constant          6 RI[  ]
3 DO[  ]            7 GO[  ]
4 DI[  ]            8 ---next page---
PROGRAM1
                                      2/3
2: R[1]=R[1]+...
[End]
```

```
PROGRAM11                       JOINT 30%
                                      3/3
 1: J  P[1] 100% FINE
 2:    R[1]=R[1]+1
[End]


[INST]                          [EDCMD] >
```

For the details of the register instruction, see Chapter 4.

Procedure 5–8 Teaching the position register instruction

Step

1 Move the cursor to END.

2 Press the F1, [INST] key. Then, the control instruction menu is displayed.

```
Instruction                          JOINT 30%
1 Registers          5 JMP/LBL
2 I/O                6 CALL
3 IF/SELECT          7 Palletizing
4 WAIT               8 ---next page---
PROGRAM1
```

3 Select REGISTERS.

```
REGISTER statement                   JOINT 30%
1 ...=...            5 ...=.../...
2 ...=...+...        6 ...=...DIV...
3 ...=...-...        7 ...=...MOV...
4 ...=...*...        8
PROGRAM1
```

4 Select PR[ ].
Teach the instruction assigning the Cartesian coordinates of the current position to the position register on the following screens.

```
REGISTER statement                 JOINT  30 %
 1 R[ ]                 5
 2 PR[ ]                6
 3 PR[i,j]              7
 4                      8
PRG1
```

```
REGISTER statement                 JOINT  30 %
 1 Lpos                 5 UTOOL[  ]
 2 Jpos                 6 PR[ ]
 3 P[ ]                 7
 4 UFRAME[  ]           8
PRG1
                                         2/3
   2:  PR[1]=...
[End]
Select item
                       [CHOICE]
```

```
PROGRAM1                             JOINT 30%
                                          3/3
  2:   PR[1]=LPOS
 [End]


 [INST]                             [EDCMD] >
```

For details of the instruction, see Chapter 4.

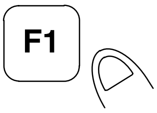
Procedure 5–9 Teaching an I/O instruction

Step

1 Move the cursor to END.

2 Press the F1, [INST] key. Then, the control instruction menu is displayed.

[INST]




```
Instruction                     JOINT 30%
1 Registers           5 JMP/LBL
2 I/O                 6 CALL
3 IF/SELECT           7 Palletizing
4 WAIT                8 ---next page---
PROGRAM
```

3 Select I/O.
Teach the instruction that turns on RO[1] on the following screens.

```
Instruction
1 Registers      5
2 I/O            6
3 IF/SELECT      7
4 WAIT           8
```

```
I/O statement                  JOINT  30 %
 1 DO[  ]=...         5 GO[  ]=...
 2 R[  ]=DI[  ]       6 R[  ]=GI[  ]
 3 RO[  ]=...         7 WO[  ]=...
 4 R[  ]=RI[  ]       8 ---next page---
PRG1
```

```
I/O statement                   JOINT 30%
1 On                  5
2 Off                 6
3 Pulse (,width)      7
4 R[ ]                8
PROGRAM1
                                      2/3
  2:  RO[1]=...
 [End]
```

```
PRG1                           JOINT  30 %
                                     3/3
   2:  RO[1]=ON
[End]



[ INST ]                          [EDCMD]>
```

For details of the instruction, see Chapter 4.

Procedure 5–10 Teaching move group instructions

Step

1 Move the cursor to the line number of a desired move statement (other than for circular movement).

```
 PROGRAM1                              JOINT 30%

 1: L P[1] 1000mm/sec CNT100
[End]



POINT                                  TOUCHUP>
```

2 Press F1, [INST]. Then, a list of control instructions is displayed.

```
 Instruction
 1 Register            5 JMP/LBL
 2 I/O                 6 Independent GP
 3 IF/SELECT           7 Simultaneous GP
 4 WAIT                8 --- next page ---
PROGRAM1
```

3 Select Independent GP or Simultaneous GP.
The contents of group 1 are moved to another group.
Note that in this case, position data remains unchanged.

```
 PROGRAM1                              JOINT 30%

 1: Independent GP
  : GP1 L P[1] 1000mm/sec CNT100
  : GP2 L P[1] 1000mm/sec CNT100


[INST]                                [EDCMD]>
```

4 For a move statement within the move group instructions, edit the move type, move speed, and positioning type in the same way as for an ordinary move statement.
Note that the following operations cannot be performed:

- Changing the move type to circular
- Specification of position data type (R[], PR[])
- Position number change
- Teaching of additional move instructions (Deletion is allowed.)
- Deletion/creation of move groups
- Position modification by SHIFT + TOUCHUP

For details of instructions, see Chapter 4.

5.3.6 TP start prohibition

The R–J3*i* MODEL B controller can execute the program immediately while editing it. To prevent the program from being executed by mistake,you can prohibit starting the program while teaching with this function.

When you select Disable FWD/BWD in the function menu,starting a program with a teach pendant is prohibited. At this time,"FBD" is reversely displayed in the upper right hand corner of the teach pendant screen to inform that TP FWD/BWD key is disabled.

This "FBD"means "Forward,Backward Disabled". To release the prohibition mode,press Disable FWD/BWD in the function menu again. At this time, the indicator of "FBD" disappears and the override is decreased to the setting value specified in the system variable,$SCR.$FWDENBLOVRD, if it is larger than the setting value.(Standard value : 10%)

Though the indicator,"FBD", displayed in upper right hand corner of the screen disappears when the teach pendant is disabled,"FBD" is displayed again when the teach pendant is enabled again.

Press and hold the SHIFT key,and press FWD or BWD in prohibition mode. At this time,a warning message,"Teach pendant is disabled",is displayed at the first line of the screen.

Jog feed during TP start prohibition

A system variable can be set to enable jog feed only in the TP start prohibition state.

To make this setting, system variable $SCR.$TPMOTNENABL is used.

To enable this function (to enable jog feed only in the TP start prohibition state), change the value of system variable $SCR.$TPMOTNENABL from 0 to 1 (or from 2 to 3) on the system variable screen.

The table below indicates the relationship between the value of system variable $SCR.$TPMOTNENABL and whether TP start and jog feed are enabled.

Table 5–5. Setting for Jog feed during TP start prohibition

$SCR.$TPMOTNENABL	TP start	Jog feed
0	Enabled	Enabled
1	Enabled	Disabled
2	Disabled	Enabled
3	Disabled	Enabled

With the standard setting, this function is disabled (jog feed is enabled irrespective of whether the teach pendant can start a program).

Procedure 5–11 Prohibiting Starting with Teach Pendant

Step

1 Press the FCTN key. The function menu is displayed.

2 Select 2 Disable FWD/BWD. ”FBD” is reversely displayed in the uppermost right hand line of the screen.

```
1 ABORT (ALL)
2 Disable FWD/BWD
3 CHANGE GROUP
```

FCTN

```
                                             FBD
SAMPLE              LINE 0
SAMPLE                          JOINT  30 %
                                       1/1
[End]

[ INST ]                          [EDCMD]>
```

3 To release the prohibition mode,select ”2 Disable FWD/BWD” in the function menu again. ”FBD” disappears and the override is reduced to a setting of $SCR.$FWDENBLOVRD.

```
1 ABORT (ALL)
2 Disable FWD/BWD
3 CHANGE GROUP
```

FCTN

```
SAMPLE              LINE 0
SAMPLE                          JOINT  30 %
                                       1/1
[End]

[ INST ]                          [EDCMD]>
```

Procedure 5–12 When effective/disable of teach pendant is switched

Condition

- TP is in prohibition mode.
- The teach pendant is disabled.

Step

1 The following program edit screen is displayed.
”FBD” is not displayed in TP prohibition state because a teach pendant is disabled.

```
SAMPLE              LINE 0
SAMPLE                          JOINT  30 %
                                       1/1
[End]

[ INST ]                          [EDCMD]>
```

2 Enable the teach pendant.
”FBD” is displayed at uppermost right hand corner of the screen and the override is reduced to the setting of $SCR.$FWDENBLOVRD.

```
                                             FBD
SAMPLE              LINE 0
SAMPLE                          JOINT  30 %
                                       1/1
[End]

[ INST ]                          [EDCMD]>
```

5.4 Changing a Program

The method of changing the contents of an existing program is described in this section.

- Selecting a program
- Modifying a standard motion instruction
- Changing a motion instruction
- Changing a control instruction
- Editing a program instruction
  - Inserting a blank line
  - Deleting a program instruction
  - Copying a program instruction
  - Finding a program instruction item
  - Replacing a program instruction item
  - Renumbering program lines

Selecting a program

Select a program from the menu of existing programs.

Changing a motion instruction

Change a motion instruction item. An example is position data, which is an instruction item that must be frequently changed.

Changing other instructions

Change other instructions.

5.4.1 Selecting a program

When selecting a program, call the registered program to display the program edit screen for editing, changing and executing a program.

Once a program is selected, the program is effective until another program is selected. While another screen is displayed such as the current position screen, the currently selected program is started by the start switch.

- When the teach pendant is enabled (The current or halted program is forcibly terminated when a program is selected.)
- When the teach pendant is disabled

Another program cannot be selected while a program is being executed or halted.

Select a program on the program selection screen.

Procedure 5–13 Selecting a program

Step

1 Press the MENUS key.

2 Select SELECT.

Alternatively, press the SELECT key to enable a program to be selected. In this case, the program selection screen is displayed.

```
Select                                  JOINT 30%
                 61092 bytes free          3/5
  1   SAMPLE1         JB[SAMPLE PROGRAM1    ]
  2   SAMPLE2         JB[SAMPLE PROGRAM2    ]
  3   SAMPLE3         JB[SAMPLE PROGRAM3    ]
  4   PROG001         PR[PROGRAM001         ]
  5   PROG002         PR[PROGRAM002         ]


 [TYPE]  CREATE  DELETE  MONITOR  [ATTR]  >
```

3 Move the cursor to the name of a program to be corrected using the cursor keys (↑ and ↓) press the ENTER key.The selected program edit screen is displayed.

```
SAMPLE3                                 JOINT 30%
                                           1/6
  1  J   P[1]  100% FINE
  2  J   P[2]  70% CNT50
  3  L   P[3]  1000cm/min CNT30
  4  L   P[4]  500mm/sec FINE
  5  J   P[1]  100% FINE
 [End]

POINT                                  TOUCHUP  >
```

5.4.2 Changing a motion instruction

When changing a motion instruction, change the instruction items of the motion instruction or change taught position data. For the motion instructions, see Section 4.3.

Changing position data

To change position data, assign new position data to the position variable by pressing the F5 (TOUCHUP) key while pressing the SHIFT key.

Position data information

The coordinates and configuration for position data can be directly changed on the position data information screen.

```
PAGE   CONFIG   DONE   [REPRE]
```

- F2 (PAGE): Toggles between the standard axes and the extended axes
- F3 (CONFIG): Edits the configuration value.
- F4 (DONE): Terminates changing the position data information.
- F5 (REPRE): Toggles between Cartesian coordinates and joint coordinates.

Changing an instruction item

To change an instruction item, press the F4, [CHOICE] key to display the motion instruction item menu, then select an instruction item from the menu.

- Motion type: Controls a path to the end position (joint, linear, circular). When the motion type is changed, the feedrate unit is also automatically changed.
- Position variable: The variable storing position data and the variable number are changed.
- Feedrate: The speed of the robot when it moves (robot motion speed) and the feedrate unit are changed.
- Positioning type: Positioning at the specified position is changed.
- Supplementary motion instruction: An additional instruction to be executed when the robot is moving is changed.

CAUTION

If teaching is made by joint coordinates, changing the user coordinate system does not affect the position variables and position registers. If the position variable is taught according to rectangular type, and the user coordinate system input option is not used, the position variable is not affected by the user coordinate system. In other cases, both of the position variable and position register are affected by the user coordinate system.

Procedure 5–14 Changing position data

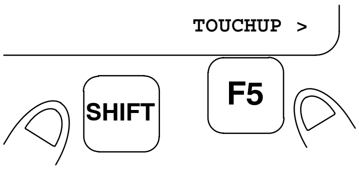
Condition
- The program to be changed must be selected.
- The teach pendant must be enabled.

Step

1 Move the cursor to the line number at which the motion instruction to be changed is displayed.

```
SAMPLE1                          JOINT 30%
                                      2/6
  1  J  P[1] 100% FINE
  2  J  P[2] 70% CNT50
  3  L  P[3] 1000cm/min CNT30
  4  L  P[4] 500mm/sec FINE
  5  J  P[1] 100% FINE
 [End]

POINT                            TOUCHUP >
```

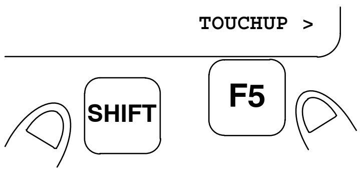
2 Move the robot to a new position and press the F5 [TOUCHUP] while pressing the SHIFT key. The new position is recorded.




```
SAMPLE1                          JOINT 30%
                                      2/6
  1  J  P[1] 100% FINE
  2  J  P[2] 70% CNT50
  3  L  P[3] 1000cm/min CNT30
  4  L  P[4] 500mm/sec FINE
  5  J  P[1] 100% FINE
 [End]
Position has been recorded to P[2].
POINT                            TOUCHUP >
```

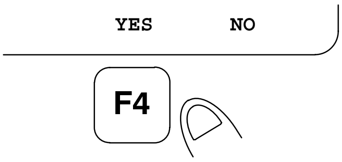
3 When the position data is taught to the motion instruction with a incremental option again,a incremental option is removed.

```
SAMPLE1                       JOINT  30 %
                                    4/5
   4:J P[3] 100% FINE INC
 [End]

 POINT                           TOUCHUP>
```



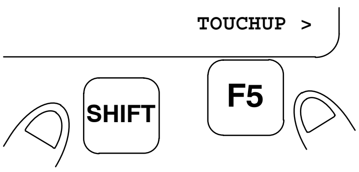

```
SAMPLE1                       JOINT  30 %
                                    4/5
   4:J P[3] 100% FINE INC
 [End]
Delete Inc option and record position ?
                          YES       NO
```

- YES : A incremental option is removed and position data is taught.
- NO : The position data is not taught.




```
SAMPLE1                       JOINT  30 %
                                    4/5
   4:J P[3] 100% FINE
 [End]
Position has been recorded to P[3].
 POINT                           TOUCHUP>
```

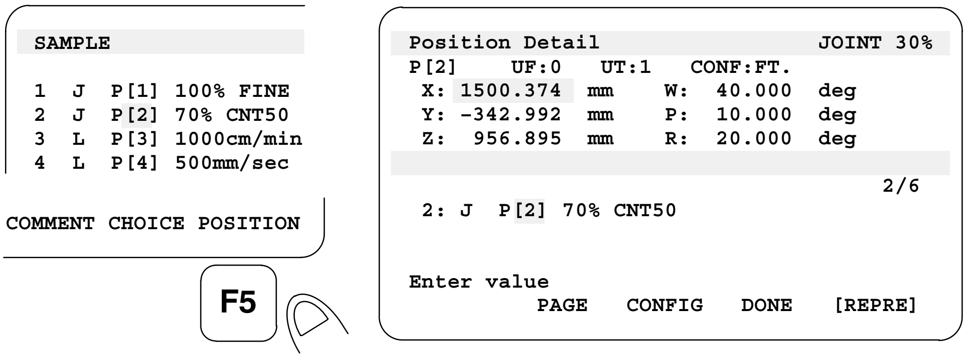
4 When position data is taught in the position register as a position variable,the position data in a register is changed by editing.



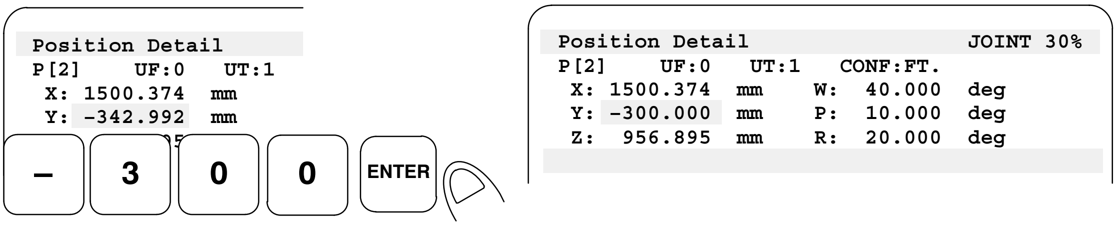

```
SAMPLE1                       JOINT  30 %
                                    5/6
   5:J PR[3] 100% FINE
 [End]
Position has been recorded to PR[3].
 POINT                           TOUCHUP>
```

Procedure 5–15 Changing position data information

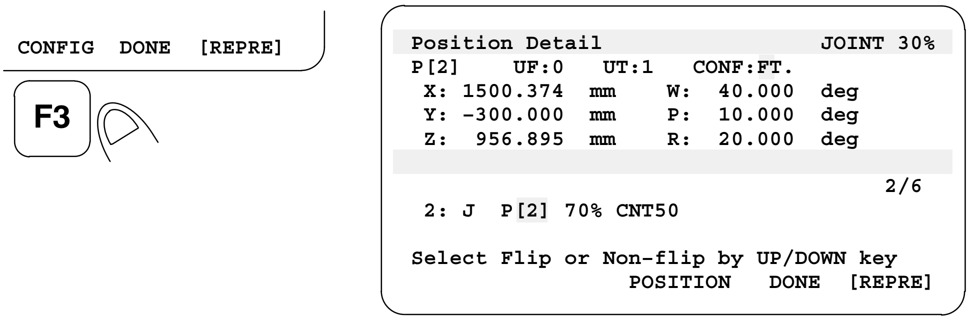
Step 1 To display position data information, move the cursor to the desired position variable, then press the F5, [POSITION] key. The position data information screen is displayed.



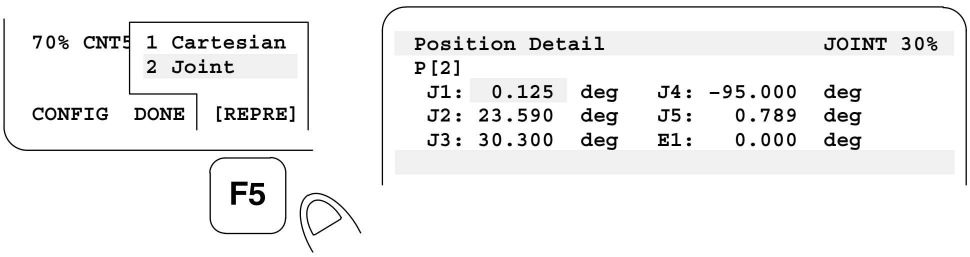
2 To change the position, move the cursor to the coordinates for each axis and enter new coordinates.



3 To change the configuration value, press the F3 [CONFIG] key, move the cursor to the configuration field, then enter a new configuration value with the cursor keys (↑ and ↓).

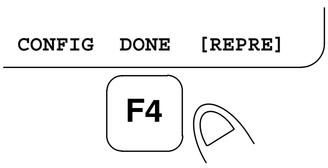


4 To change a coordinate system, press the F5 [REPRE] key and select the coordinate system to be changed.



NOTE JOINT display is valid when the robot is adjusted to the zero–degree position or when non–kinematic operation such as table operation control is executed.

5 After changing position data information, press the F4 [DONE] key.



Procedure 5–16 Changing a motion instruction

Step

1 Move the cursor to the instruction item of a motion instruction to be changed.

2 Press the F4 [CHOICE] key to display the submenu of the instruction items, then select the instruction item to be changed from the submenu.

The following screens show changing the motion type from linear motion to joint motion:

```
SAMPLE1

  5: L  P[5] 500cm/min
[End]

        [CHOICE]
```

F4

```
Motion Modify                    JOINT 30%
1  Joint
2  Linear
3  Circular
4
SAMPLE1
                                      5/6
5: L  P[5] 500cm/min CNT30
[End]
Select item
                       [CHOICE]
```

```
Motion Modify
 1 Joint
 2 Linear
 3 Circular
 4
```

ENTER

```
SAMPLE1                          JOINT 30%
                                      5/6
5: J  P[5] 100% CNT30
[End]


Enter value or press ENTER
              COMMENT  [CHOICE]  POSITION
```

3 The following screens show changing from the position variable to the position register.

```
SAMPLE1

  5: J  P[5] 100% CNT3
[End]

        [CHOICE]
```

F4

```
Motion Modify                    JOINT 30%
1  P[ ]
2  PR[ ]
3
4
SAMPLE1
                                      5/6
5: J  P[5] 100% CNT30
[End]
```

```
Motion Modify
 1 P[ ]
 2 PR[ ]
 3
 4
```

ENTER

```
SAMPLE1                          JOINT 30%
                                      5/6
5: J  PR[...] 100% CNT30
[End]


Enter Value
        DIRECT INDIRECT [CHOICE] POSITION
```

4 Change the feedrate.

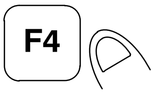
```
SAMPLE1

  2: J  P[2] 100% FINE
```

7 0 ENTER

```
SAMPLE1                          JOINT 30%
                                      2/6
2: J  P[2] 70% FINE
[End]


Enter Value
                       [CHOICE]
```

5 Change the feedrate unit.

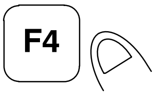
```
SAMPLE1

 4: L  P[2] 500cm/mm

     [CHOICE]
```




```
Motion Modify                    JOINT 30%
1  mm/sec             5  sec
2  cm/min             6  msec
3  inch/min           7
4  deg/sec            8
SAMPLE1
                                     4/6
4: L  P[4] 500cm/min CNT30
[End]
```

6 Change the positioning type.

```
              JOINT 30%
                  2/6
 70% FINE

        [CHOICE]
```




```
Motion Modify                    JOINT 30%
1  Fine
2  Cnt
3
4
SAMPLE1
                                     2/6
2: L  P[2] 70% FINE

Select item
                       [CHOICE]
```

Procedure 5–17 Changing a circular motion instruction

Step 1 Place the cursor at the motion type of the circular motion instruction to be changed.
The following screens show changing the circular motion instruction to the linear motion instruction.

```
SAMPLE1                                   JOINT 30%
                                                6/7
6: C  P[5]
 :    P[6] 500cm/min CNT30
[End]

                               [CHOICE]
```

```
SAMPLE1

 6: C  P[5]
  :    P[6] 500cm/min

      [CHOICE]
```

F4

```
Motion Modify                             JOINT 30%
1  Joint
2  Liner
3  Circular
4
SAMPLE1
                                                6/7
6: C  P[5]
 :    P[6] 500cm/min CNT30
```

```
Motion Modify
 1 Joint
 2 Linear
 3 Circular
 4
```

ENTER

```
SAMPLE1                                   JOINT 30%
                                                6/7
6: L  P[6] 500cm/min CNT30
[End]

                     COMMENT  [CHOICE]  POSITION
```

NOTE When a circular motion is changed to a joint or linear motion, two motion instructions are created as a result. One instruction moves the tool to the passing point of the circular motion, while the other moves the tool to the end point.

2 The following screens show changing the linear motion instruction to the circular motion instruction.

```
SAMPLE1

 6: L  P[6] 500cm/min
[End]

      [CHOICE]
```

F4

```
Motion Modify                             JOINT 30%
1  Joint
2  Liner
3  Circular
4
SAMPLE1
                                                6/7
6: L  P[6] 500cm/min CNT30
```

```
Motion Modify
 1 Joint
 2 Linear
 3 Circular
 4
```

ENTER

```
SAMPLE1                                   JOINT 30%
                                                6/7
6: C  P[6]
 :    P[...] 500cm/min CNT30
[End]
Enter value or press ENTER
                               [CHOICE]
```

NOTE When a joint or linear motion instruction is changed to a circular motion instruction, the taught data for the end point of the arc is canceled.

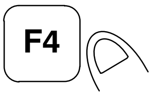
Procedure 5–18 Adding and deleting an additional motion instruction

Step

1 Position the cursor to an additional motion instruction. To add an offset condition instruction, for example, follow the procedure below:

```
SAMPLE1                               JOINT 30%
                                            7/8
7: L  P[2] 300mm/sec FINE
[End]


                          [CHOICE]
```

```
          JOINT 30%
               7/8
100% FINE

      [CHOICE]
```




```
Motion Modify                         JOINT 30%
 1  No Option            5  Incremetal
 2  Wrist Joint          6  Skip,LBL[ ]
 3  Offset               7
 4  Offset,PR[ ]         8
SAMPLE1
                                            7/8
 7: L  P[2] 300mm/sec FINE
[End]
```

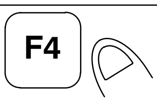
```
Motion Modify
   1  No Option
   2  Wrist Joint
   3  Offset
   4  Offset,PR[ ]
```


```
SAMPLE1                               JOINT 30%
                                            7/8
7: L  P[2] 300mm/sec FINE  offset
[End]


                          [CHOICE]
```

2 To delete an offset condition instruction, for example, follow the procedure below:

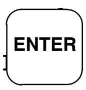
```
SAMPLE1                               JOINT 30%
                                            7/8
7: L  P[2] 300mm/sec FINE  offset
[End]


                          [CHOICE]
```

```
          JOINT 30%
               7/8
300mm/sec FINE  offset

      [CHOICE]
```




```
Motion Modify                         JOINT 30%
 1  No Option            5  Incremetal
 2  Wrist Joint          6  Skip,LBL[ ]
 3  Offset               7
 4  Offset,PR[ ]         8
SAMPLE1
                                            7/8
 7: L  P[2] 300mm/sec FINE  offset
[End]
```

```
Motion Modify
 1 No Option          5
 2 Wrist Joint
 3 Offset
 4 Offset,PR[ ]
```




```
SAMPLE1                               JOINT 30%
                                            7/8
7: L  P[2] 300mm/sec FINE
[End]


                          [CHOICE]
```

Procedure 5–19 Changing the move speed (between numeric specification and register specification)

```
 PNS0001                          JOINT 10%
                                        1/2
 1: J P[1] 100% FINE
[End]

Enter Value

REGISTER                          [CHOICE]
```

Step

1 To switch from numeric specification to register specification for the move speed of a move instruction, move the cursor to the speed value. Then, press the function key F1, [REGISTER].

```
 PNS0001                          JOINT 10%
                                        1/2
 1: J P[1] R[...]% FINE
[End]


Enter Value
SPEED     DIRECT     INDIRECT     [CHOICE]
```

2 Enter a desired register number (2 for example). For indirect specification, press F3 [INDIRECT]. (To return to direct specification mode, press F2 [DIRECT].)

```
 PNS0001                          JOINT 10%
                                        1/2
 1: J P[1] R[2]% FINE
[End]



                                  [CHOICE]
```

3 To switch from register specification to numeric specification for the move speed of a move instruction

```
 PNS0001                          JOINT 10%
                                        1/2
 1: J P[1] R[2]% FINE
[End]


Enter Value
SPEED     DIRECT     INDIRECT     [CHOICE]
```

4 Move the cursor to the speed value. Then, press the function key F1 (SPEED).

```
 PNS0001                          JOINT 10%
                                        1/2
 1: J P[1] ...% FINE
[End]


Enter Value
REGISTER                         [CHOICE]
```

5 Enter a desired speed value (20 for example).

```
 PNS0001                          JOINT 10%
                                        1/2
 1: J P[1] 20% FINE
[End]



                                 [CHOICE]
```

5.4.3 Changing a control instruction

You can change the syntax, item, or variable of a control instruction.

Step 1 Move the cursor to the instruction item to be changed.

```
PROGRAM1                                  JOINT 30%
                                              11/20
  10: J  P[5] 100% FINE
  11: WAIT RI[1]=ON
  12: RO[1]=ON

                                [CHOICE]
```

2 Press the F4 (CHOICE) key to display the instruction menu and select the instruction item to be changed. The following screens show changing the wait instruction.

```
[CHOICE]
F4
```

```
Wait statements                      JOINT  30 %
 1 R[ ]               5 DO[  ]
 2 Constant           6 DI[  ]
 3 On                 7 RO[  ]
 4 Off                8 ---next page---
PRG1
                                        11/20
  11:  WAIT RI[1]=ON
  12:  RO[1]=ON
Select item
                             [CHOICE]
```

```
Wait statements
 1 R[ ]
 2 Constant
 3 On
 4 Off
ENTER
```

```
PRG1                                 JOINT  30 %
                                        11/20
  11:  WAIT RI[1]=R[...]
Enter value
         DIRECT INDIRECT[CHOICE]  LIST
```

```
2  ENTER
```

```
PRG1                                      JOINT 30%
                                              11/20
  11: WAIT RI[1]=R[2]
  12: RO[1]=ON
         DIRECT INDIRECT[CHOICE]  LIST
```

```
[CHOICE]
F4
```

```
Wait statements
 1 <Forever>
 2 Timeout-LBL[  ]
 3
 4
ENTER
```

```
PRG1                                 JOINT  30 %
                                        11/20
  11:  WAIT RI[1]=R[2] TIMEOUT,LBL[...]
  12:  RO[1]=ON
Enter value
         DIRECT INDIRECT[CHOICE]
```

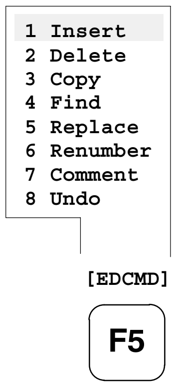
```
2  ENTER
```

```
PRG1                                 JOINT  30 %
                                        12/20
  11:  WAIT RI[1]=R[2] TIMEOUT,LBL[2]
  12:  RO[1]=ON
```

5.4.4 Program edit instructions

The program edit instructions are used to edit a program.

Press the F5 (EDCMDT) key to display the program edit instruction menu and select a desired program edit instruction from the menu.



Insert

Inserts blank lines, the number of which is specified, between the existing lines of a program. When blank lines are inserted, the program lines are renumbered.

Delete

Deletes a series of instructions from a program. After the instructions are deleted, the program lines are renumbered.

Copy

Copies a series of instructions and inserts the instruction range into another location in the program. When a series of instructions is copied, the instruction group is selected and recorded in memory. Once the series of instructions is copied, it can be inserted into other locations in the program repeatedly.

Find

A specified element of a program instruction is found. A specified element of a long program can be found quickly.

Replace

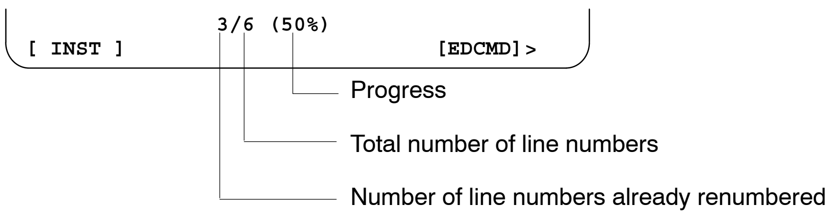
Replaces an item of the specified program instruction with another item. This program is used, for example, when setup data for the program is changed. (For example, when the I/O allocation is changed, and SDO[1] is to be changed to SDO[2] in the program)

Renumber

Renumbers the program lines by line number in ascending order. Whenever a motion instruction is taught, the line number is increased regardless of its location in the program. When insertions and deletions are repeated, the line numbers are not sequentially arranged in a program. Renumbering arranges them sequentially in the program.

The progress of renumbering is displayed as it is being performed.

CAUTION Do not perform power down before the progress indication reaches 100%. Otherwise, the program will be incomplete.



Comment

On the program editing screen, the user can choose whether to display or hide a comment for the instructions listed below. Note that no comment can be edited.

- SDI instruction, SDO instruction, RDI instruction, RDO instruction, GI instruction, GO instruction, AI instruction, AO instruction, UI instruction, UO instruction, SI instruction, SO instruction
- Register instructions
- Position register instructions (including position registers in the position data format for move instructions)
- Palletizing instructions
- Move instruction register speed specifications

NOTE The AI and AO instructions are analog I/O soft options.

The instructions listed below are always accompanied by a comment, and do not allow display switching but allow editing.

- Move instruction position variable
- Label instructions
- Power control instructions

NOTE The comment display area for an instruction item that is too long to be displayed on one line of the screen may be shortened.

NOTE No comment is displayed for a register indirect specification.
Position register [register [1]] = ...

Undo

Program edit operations such as an instruction modification, line insertion, and line deletion can be cancelled to return to the state present before those edit operations are performed. If an undo operation is performed during editing of a program line, all operations performed for that line are undone. For example, if a line is inserted or deleted, the state before the insertion or deletion operation is restored.

If an undo operation is immediately followed by another undo operation, the state present before the first undo operation is performed is restored.

NOTE If an undo operation is performed for a line during program editing, all operations performed for that line are undone. This means that if an instruction is taught in a blank line or the last line of a program, and an undo operation is performed for that line during editing, the taught instruction is deleted.

Procedure 5–20 Inserting blank lines

Step

1 Press the NEXT, > to display F5, EDCMD.

NEXT

```
SAMPLE1                              JOINT 30%
                                          4/6
  1: J  P[1] 100% FINE
  2: J  P[2] 70% CNT50
  3: L  P[3] 1000cm/min CNT30
  4: L  P[4] 500mm/sec FINE
  5: J  P[1] 100% FINE
 [End]

 [INST]                           [EDCMD]
```

2 Press the F5, [EDCMD] key. The edit instruction menu is displayed.

3 Select Insert.

```
100% FIN   1 Insert
70% CNT5   2 Delete
1000cm/m   3 Copy
500mm/se   4 Find
100% FIN   5 Replace
           6 Renumber

                [EDCMD]
```

F5 ENTER

```
SAMPLE1                              JOINT 30%
                                          4/6
  1: J  P[1] 100% FINE
  2: J  P[2] 70% CNT50
  3: L  P[3] 1000cm/min CNT30
  4: L  P[4] 500mm/sec FINE
  5: J  P[1] 100% FINE
 [End]
  How many line to insert ?:
```

In the example below, two blank lines are inserted between the 3rd and 4th lines.

4 Move the cursor to the line where instructions are to be inserted. In this example, move the cursor to the 4th line.

5 Enter the number of blank lines to be inserted (two) and press the ENTER key.

2 ENTER

```
SAMPLE1                              JOINT 30%
                                          4/8
  1: J  P[1] 100% FINE
  2: J  P[2] 70% CNT50
  3: L  P[3] 1000cm/min CNT30
  4:
  5:
  6: L  P[4] 500mm/sec FINE
  5: J  P[1] 100% FINE

 [INST]                           [EDCMD] >
```

The two blank lines are inserted into the program and all the lines in the program are renumbered.

Procedure 5–21 Deleting instructions

Step

1 Move the cursor to the top of the line in which the instruction to be deleted is positioned. (Specify the line to be deleted with the cursor.)

2 Press the NEXT, > to display F5, EDCMD.

NEXT

```
SAMPLE1                              JOINT 30%
                                          4/6
  1: J   P[1] 100% FINE
  2: J   P[2] 70% CNT50
  3: L   P[3] 1000cm/min CNT30
  4: L   P[4] 500mm/sec FINE
  5: J   P[1] 100% FINE
 [End]

 [INST]                                [EDCMD]
```

3 Press the F5, [EDCMD] key to display the editing instruction menu.

4 Select Delete

```
100% FIN   1 Insert
70% CNT5   2 Delete
1000cm/m   3 Copy
500mm/se   4 Find
100% FIN   5 Replace
           6 Renumber

              [EDCMD]
```

F5 ENTER

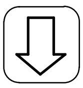
```
SAMPLE1                          JOINT  30 %
                                        4/6
    1:J P[1] 100% FINE
    2:J P[2] 70% CNT50
    3:L P[3] 1000cm/min CNT30
    4:L P[4] 500mm/sec FINE
    5:J P[1] 100% FINE
 [End]

Delete line(s) ?
                                 YES      NO
```

CAUTION

Once an instruction is deleted, the instruction is not restored. Be sure to confirm whether an instruction to be deleted should be done before doing it, or important data may be lost.

5 Specify the range of instruction lines to be deleted with the cursor keys (↑ and ↓).

```
  3: L   P[3] 1000cm/
  4: L   P[4] 500mm/s
  5: J   P[1] 100% FI
 [End]
```



6 To cancel deleting the selected line, press the F5 (NO) key. To delete the selected lines, press the F4 (YES) key.

```
        YES      NO
```

F4

```
SAMPLE1                              JOINT 30%
                                          4/4
  1: J   P[1] 100% FINE
  2: J   P[2] 70% CNT50
  3: L   P[3] 1000cm/min CNT30
 [End]


 [INST]                                [EDCMD] >
```

Procedure 5–22 Copying and pasting instructions

Step 1 Press the NEXT, > until F5, EDCMD.

NEXT

```
SAMPLE1                                JOINT 30%
                                            1/6
   1: J   P[1] 100% FINE
   2: J   P[2] 70% CNT50
   3: L   P[3] 1000cm/min CNT30
   4: L   P[4] 500mm/sec FINE
   5: J   P[1] 100% FINE
  [End]

  [INST]                                [EDCMD]
```

2 Press the F5, [EDCMD] key. The editing instruction menu is displayed.

3 Select 3 Copy.

The following screens show copying 2th to 4th lines to 5th to 7th lines.

```
 1 Insert
 2 Delete
 3 Copy
 4 Find
 5 Replace
 6 Renumber

     EDCMD
```

F5 ENTER

```
SAMPLE1                                JOINT 30%
                                            2/6
   1: J   P[1] 100% FINE
   2: J   P[2] 70% CNT50
   3: L   P[3] 1000cm/min CNT30
   4: L   P[4] 500mm/sec FINE
   5: J   P[1] 100% FINE
  [End]
   Select lines
           COPY                     PASTE
```

4 Select the range of lines to be copied.

```
   1: J   P[1]
   2: J   P[2]
   3: L   P[3]
   4: L   P[4]
   5: J   P[1]
  [End]
```

```
   1: J   P[1]
   2: J   P[2]
   3: L   P[3]
   4: L   P[4]
   5: J   P[1]
  [End]
   Move cursor to select range
           COPY
```

F2

```
SAMPLE1

   1: J   P[1] 100% FINE
   2: J   P[2] 70% CNT50
   3: L   P[3] 1000cm/min CNT30
   4: L   P[4] 500mm/sec FINE
   5: J   P[1] 100% FINE
  [End]
   Select lines
           COPY
```

As a result of above steps, the selected instructions (2nd to 4th lines in this example) were copied in memory.

5 Decide where you want to paste the sentences copied in the memory.

```
      PASTE
```

F5

```
SAMPLE1                             JOINT  30 %
                                          5/6
   1:J P[1] 100% FINE
   2:J P[2] 70% CNT50
   3:L P[3] 1000cm/min CNT30
   4:L P[4] 500mm/sec FINE
   5:J P[5] 100% FINE INC
  [End]

Paste before this line ?
       LOGIC    POS-ID   POSITION CANCEL>
     R-LOGIC R-POS-ID R-POSITION CANCEL>
```

6 Select the copying and pasting method (copying from the original).

```
POS-ID POSITION  CANCEL
```

F3

```
SAMPLE1                            JOINT 30%
                                        8/9
  1: J  P[1] 100% FINE
  2: J  P[2] 70% CNT50
  3: L  P[3] 1000cm/min CNT30
  4: L  P[4] 500mm/sec FINE
  5: J  P[2] 70% CNT50
  6: L  P[3] 1000cm/min CNT30
  7: L  P[4] 500mm/sec FINE
  8: J  P[1] 100% FINE
 [End]

  Select lines
                   COPY            PASTE >
```

7 By repeating the above steps 5 to 6, the same instruction group can be pasted at any number of locations in the program.

8 To terminate the pasting of instructions, press the PREV key

PREV

Pasting methods

The following copying and pasting methods are provided:

```
        LOGIC   POS-ID   POSITION CANCEL>
```

– F2 (LOGIC) : Copies and pastes motion instructions with no position data specified.
– F3 (POS–ID) : Copies and pastes motion instructions with the position numbers unchanged.
– F4 (POSITION) : Copies and pastes motion instructions with the position numbers updated.

Pressing the next page key (NEXT) displays the following function key menu:

NEXT

```
      R-LOGIC R-POS-ID R-POSITION CANCEL>
```

The selected instructions are copied in reverse order.

F3 and F5 have the following functions:

– F3 (RM–POS–ID) : Copies the move instructions at a copy source in reverse order without changing the position numbers of the move instructions. The move type, move speed, and so forth of each move instruction are changed so that a movement totally opposite to the movement of the copy source is made.

– F5 (RM–POS) : Copies the move instructions at a copy source in reverse order. then assigns new position numbers. The move type, move speed, and so forth of each move instruction are changed so that a movement totally opposite to the movement of the copy source is made.

NOTE The copy function for a reverse movement is not supported for the additional move instructions listed below. If the move instructions at a copy source include any of the move instructions below, RM–POS–ID or RM–POS generates a warning, and only a copy operation in reverse order is performed.

- Application instruction
- Skip and high–speed skip instructions
- Incremental instruction
- Continuous rotation instruction
- Pre–execution/post–execution instruction
- Multi–group operation

Example When the F4 (R–POSITION) is pressed

```
R-LOGIC R-POS-ID R-POSITION
```

F4

```
 1: J  P[1] 100% FINE
 2: J  P[2] 70% CNT50
 3: L  P[3] 1000cm/min CNT30
 4: L  P[4] 500mm/sec FINE
 5: J  P[7] 500mm/sec FINE
 6: L  P[6] 1000cm/min CNT30
 7: L  P[5] 70% CNT50
 8: J  P[1] 100% FINE
[End]

 Select lines
                  COPY            PASTE >
```

Procedure 5–23 Finding a program instruction item

Step 1 Press the NEXT, > until F5, EDCMD.

NEXT

```
SAMPLE3                        JOINT  30 %
                                    1/10
   1:J P[1] 100% FINE
   2:  R[1]=0
   3:  LBL[1]
   4:L P[2] 1000cm/min CNT30
   5:L P[3] 500mm/sec FINE
   6:  IF DI[1]=ON JMP LBL[2]
   7:  R[1]=R[1]+1
   8:  JMP LBL[1]
   9:  LBL[2]
[End]

[ INST ]                          [EDCMD]>
```

2 Press the F5, [EDCMD] key. The editing instruction menu is displayed.

3 Select Find.

4 Select a program instruction item to be found. The following screens show how to find instruction, WAIT.

```
100% FIN  1 Insert
70% CNT5  2 Delete
1000cm/m  3 Copy
500mm/se  4 Find
100% FIN  5 Replace
          6 Renumber

               [EDCMD]
```

F5 ENTER

```
Select Find menu              JOINT  30 %
 1 Registers        5 JMP/LBL
 2 CALL              6 Miscellaneous
 3 I/O               7 Program control
 4 IF/SELECT         8 ---next page---
SAMPLE3
```

```
Select Find item              JOINT  30 %
 1 JMP LBL[  ]       5
 2 LBL[  ]           6
 3                   7
 4                   8
SAMPLE3


Enter index value
```

5 When the item to be found is an index, enter the value. To find an item regardless of whether the item is an index, press the ENTER key without entering anything.

```
Enter index value
```

ENTER

```
 SAMPLE3                               JOINT  30 %
                                            1/10
    1:J P[1] 100% FINE
    2:  R[1]=0
    3:  LBL[1]
    4:L P[2] 1000cm/min CNT30
    5:L P[3] 500mm/sec FINE
    6:  IF DI[1]=ON JMP LBL[2]
    7:  R[1]=R[1]+1
    8:  JMP LBL[1]
    9:  LBL[2]
 [End]

                            NEXT   EXIT
```

If the specified instruction is found in the program, the cursor stops at the instruction.

6 To find the same instruction again, press the F4 (NEXT) key.

```
NEXT   EXIT
```

F4

```
 SAMPLE3                               JOINT  30 %
                                            1/10
    1:J P[1] 100% FINE
    2:  R[1]=0
    3:  LBL[1]
    4:L P[2] 1000cm/min CNT30
    5:L P[3] 500mm/sec FINE
    6:  IF DI[1]=ON JMP LBL[2]
    7:  R[1]=R[1]+1
    8:  JMP LBL[1]
    9:  LBL[2]
 [End]

                            NEXT   EXIT
```

7 To terminate finding an instruction, press the F5 (EXIT) key.

```
NEXT   EXIT
```

F5

NOTE The position of a track/offset instruction or touch sensor instruction cannot be found using the search instruction.

Procedure 5–24 Replacing a program instruction item

Step

1 Press the NEXT, > until F5, EDCMD.

```
SAMPLE3                              JOINT  30 %
                                           1/9
   1:J P[1] 100% FINE
   2:J P[2] 70% CNT50
   3:  LBL[1]
   4:L P[3] 1000cm/min CNT30
   5:L P[4] 500mm/sec FINE
    : SKIP LBL[2]
   6:  JMP LBL[1]
   7:  LBL[2]
   8:J P[5] 100% FINE
[End]

[ INST ]                               [EDCMD]>
```

2 Press the F5, [EDCMD] key. The changing instruction menu is displayed.

3 Select Replace.

4 Select a program instruction item to be replaced and press the ENTER key.
In the screen below the feedrate specified in the motion instruction is changed to another value.

```
 1 Insert
 2 Delete
 3 Copy
 4 Find
 5 Replace
 6 Renumber

      EDCMD
```

F5 ENTER

```
Select Replace menu           JOINT  30 %
 1 Registers          5 Motion modify
 2 CALL               6
 3 I/O                7
 4 JMP/LBL            8
SAMPLE3
```

```
Modify motion menu            JOINT  30 %
 1 Replace speed      5
 2 Replace term       6
 3 Insert option      7
 4 Remove option      8
SAMPLE3
```

The following replacement items are displayed:

– Replace speed: Changes the feedrate to another value.
– Replace term: Changes the positioning type to another value.
– Insert option: Inserts an supplementary motion instruction.
– Remove option: Deletes an supplementary motion instruction.

5 Select Replace speed.

```
Modify motion menu
 1 Replace speed
 2 Replace term
 3 Insert option
 4 Remove option
SAMPLE3
```

ENTER

```
Select interporate                JOINT 30%
 1 Unspecified type 5
 2 J                6
 3 L                7
 4 C                8
SAMPLE3
                                      1/10
```

– Unspecified type: Changes the feedrates in all motion instructions
– J: Changes the feedrates only in motion instructions for joint control.
– L: Changes the feedrates only in motion instructions for linear control.
– C: Changes the feedrates only in motion instructions for circular control.

6 Specify the target type of the operation instruction.

```
Select interpolate
 1 Unspecified type 5
 2 J                6
 3 L
 4 C
PNS0001
```

INPUT

```
Speed type menu                    JOINT  10 %
 1 All type          5
 2 Speed value       6
 3 R[ ]              7
 4 R[R[ ]]           8
PNS0001
```

- ALL type: No speed type is specified.
- Speed value: Operation statements that specify a speed with a numeric value are specified.
- R[ ]: Operation statements that specify a speed with a register are specified.
- R[R[ ]]: Operation statements that indirectly specify a speed value with registers are specified.

7 Specify a target speed format.

```
Speed type menu

 2 Speed value       6
 3 R[ ]              7
 4 R[R[ ]]           8
PNS0001
```

```
Select motion item                 JOINT  10 %
 1 %                 5 deg/sec
 2 mm/sec            6 sec
 3 cm/min            7
 4 inch/min          8
PNS0001
```

8 Specify a target speed unit.

```
Select motion item
 1 %                 5
 2 mm/sec            6
 3 cm/min
 4 inch/min
PNS0001
```

INPUT

```
Speed type menu                    JOINT  10 %
 1 Speed value       5
 2 R[ ]              6
 3 R[R[ ]]           7
 4                   8
PNS0001
```

- Speed value: The selected statement is changed to an operation statement which specifies a speed with a numeric value.
- R[ ]: The selected statement is changed to an operation statement which specifies a speed using a register.
- R[R[ ]]: The selected statement is changed to an operation statement which indirectly specifies a speed by using registers.

9 Specify the motion type of the motion instruction for which the feedrate is to be changed.

```
Select interporate
 1 Unspecified type  5
 2 J                 6
 3 L
 4 C
```

ENTER

```
Select motion item                 JOINT 30%
 1 %                 5 deg/sec
 2 mm/sec            6 sec
 3 cm/min            7
 4 inch/min          8
SAMPLE3
                                        1/10
```

10 Specify the unit of the feedrate to be changed.

```
Select motion item
 1 %                 5
 2 mm/sec            6
 3 cm/min            7
 4 inch/min          8
```

ENTER

```
Enter speed value:
```

11 Enter a desired feedrate.

```
Enter speed value:50
```

5 0 ENTER

```
SAMPLE3                              JOINT  30 %
                                           1/10
    1:J P[1] 100% FINE
    2:J P[2] 70% CNT50
 Modify OK ?

             ALL    YES       NEXT   EXIT
```

The kinds of replacing items are displayed.

– F2 (ALL): Replaces all the items in the current line and subsequent lines.
– F3 (YES): Replaces the item at the cursor and finds the next item.
– F4 (NEXT): Finds the next item.

12 Select a replacement method.

```
Modify OK ?
           ALL      YES
```

F2

```
SAMPLE3                              JOINT  30 %
                                            1/9
   1:J P[1] 50% FINE
   2:J P[2] 50% CNT50
   3:  LBL[1]
   4:L P[3] 1000cm/min CNT30
   5:L P[4] 500mm/sec FINE
     : SKIP LBL[2]
   6:  JMP LBL[1]
   7:  LBL[2]
   8:J P[5] 50% FINE
[End]

[ INST ]                              [EDCMD]>
```

13 To terminate item replacement, press the F5 (EXIT) key.

```
YES     NEXT     EXIT
```

F5

CAUTION

The replacement instruction allows no move instruction to be replaced with the track/offset instruction or touch sensor instruction. If an attempt for such replacement is made, a memory write alarm is issued. To replace a move instruction, first delete the move instruction, then insert the touch sensor instruction or track instruction.

Procedure 5–25 Renumbering the position number

Step

1 Press the NEXT, >, then press the F5, EDCMD.

```
SAMPLE1                                JOINT 30%
                                             1/8
  1: J  P[8] 100% FINE
  2: J  P[6] 70% CNT50
  3: L  P[3] 1000cm/min CNT30
  4: L  P[5] 500mm/sec FINE
  5: J  P[1] 100% FINE
  6: L  P[5] 500mm/sec FINE
  5: J  P[8] 100% FINE
 [End]

 [INST]                                 [EDCMD]
```

2 Press F5, EDCMD. The changing instruction menu is displayed.

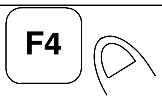
3 Select Renumber.

```
 1 Insert
 2 Delete
 3 Copy
 4 Find
 5 Replace
 6 Renumber

        EDCMD
```


```
Renumber OK?
                          YES     NO
```

4 To renumber the program lines, press the F4 (YES) key. To cancel renumbering the program lines, press the F5 (NO) key.

```
     YES      NO
```




```
SAMPLE1                                JOINT 30%
                                             1/8
  1: J  P[1] 100% FINE
  2: J  P[2] 70% CNT50
  3: L  P[3] 1000cm/min CNT30
  4: L  P[4] 500mm/sec FINE
  5: J  P[5] 100% FINE
  6: L  P[4] 500mm/sec FINE
  7: J  P[1] 100% FINE
  [INST]                                 [EDCMD]
```

Procedure 5–26 Comment display switching

Step

1 Press F→ to display F5 (EDCMD).

```
 PNS0001                              JOINT 10%
                                           1/9
   1:  R[1]=DI[2]
   2:  SDO[3]=ON
   3:  R[R[1]]=DI[R[2]]
   4:  PR[1]=P[3]
   5:  PR[1,2]=PR[R[3],R[4]]
   6:  PL[1]=PL[R[3]]
   7:  J PR[1] 100% FINE
   8:  J  P[1] 100% FINE
   9:  LBL[1]
 [End]

[INST]                              [EDCMD]>
```

2 Press F5 (EDCMD) to display the edit instruction menu.

```
 PNS0001                              JOINT 10%
                                           1/9
   1:  R[1]=DI[2]
   2:  DO[3]=ON                 1 Insert
   3:  R[R[1]]=DI[R[2]]         2 Delete
   4:  PR[1]=P[3]               3 Copy
   5:  PR[1,2]=PR[R[3],R[4]     4 Find
   6:  PL[1]=PL[R[3]]           5 Replace
   7:  J PR[1] 100% FINE        6 Remember
   8:  J  P[1] 100% FINE        7 Comment
   9:  LBL[1]                   8 Undo
 [End]

[INST]                               EDCMD
```

3 Select Item 7 Comment.

```
 PNS0001                              JOINT 10%
                                           1/9
   1:  R[1:Comment]=DI[2:Comment]
   2:  DO[3:Comment]=ON
   3:  R[R[1]]=DI[R[2]]
   4:  PR[1:Comment]=P[3:Comment]
   5:  PR[1,2:Comment]=PR[R[3],R[4]]
   6:  PL[1:Comment]=PL[R[3]]
   7:  J PR[1:Comment] 100% FINE
   8:  J  P[1:Comment] 100% FINE
   9:  LBL[1:Comment]
 [End]

[INST]                              [EDCMD]>
```

4 To disable comment display, select Comment of the function key F5 (EDCMD) again.

Procedure 5–27 Undoing edit operations

Step 1 Press F→ to display F5 (EDCMD).

```
 PNS0001                                JOINT 10%
                                             1/9
   1:   R[1]=DI[2]
   2:   SDO[3]=ON
   3:   R[R[1]]=DI[R[2]]
   4:   PR[1]=P[3]
   5:   PR[1,2]=PR[R[3],R[4]]
   6:   PL[1]=PL[R[3]]
   7:   J PR[1] 100% FINE
   8:   J  P[1] 100% FINE
   9:   LBL[1]
 [End]

[INST]                              [EDCMD]>
```

2 Press F5 (EDCMD) to display the edit instruction menu.

```
 PNS0001                                JOINT 10%
                                             1/9
   1:   R[1]=DI[2]
   2:   SDO[3]=ON               1 Insert
   3:   R[R[1]]=DI[R[2]]        2 Delete
   4:   PR[1]=P[3]              3 Copy
   5:   PR[1,2]=PR[R[3],R[4]    4 Find
   6:   PL[1]=PL[R[3]]          5 Replace
   7:   J PR[1] 100% FINE       6 Remember
   8:   J  P[1] 100% FINE       7 Comment
   9:   LBL[1]                  8 Undo
 [End]

[INST]                               EDCMD
```

3 Select Item 8 Undo.

```
 PNS0001                                JOINT 10%
                                             1/9
   1:   R[1]=DI[2]
   2:   DO[3]=ON
   3:   R[R[1]]=DI[R[2]]
   4:   PR[1]=P[3]
   5:   PR[1,2]=PR[R[3],R[4]]
   6:   PL[1]=PL[R[3]]
   7:   J PR[1] 100% FINE
   8:   J  P[1] 100% FINE
   9:   LBL[1]

 Undo? (Edit)
                                   YES      NO
```

4 To perform an undo operation, select F4, (YES). To cancel the undo operation, select F5, (NO). When F4, (YES) is selected, the edit operation is undone.

```
PNS0001                                JOINT 10%
                                            1/9
  1:  R[1]=DI[2]
  2:  SDO[3]=ON
  3:  R[R[1]]=DI[R[2]]
  4:  PR[1]=P[3]
  5:  PR[1,2]=PR[R[3],R[4]]
  6:  PL[1]=PL[R[3]]
  7:  J PR[1] 100% FINE
  8:  J  P[1] 100% FINE
  9:  LBL[1]
 [End]

 [INST]                              [EDCMD]>
```

5 When an additional undo operation is performed in succession, the first undo operation performed can be cancelled; this means the state present before the first undo operation is performed is restored.

NOTE If an edit operation is performed after an undo operation, the undo operation cannot be cancelled.

> **CAUTION**
>
> An undo operation automatically rewrites the program, so that the results may not be those expected by the operator. Before executing a program after an undo operation, carefully check the program.

- This function can undo the following operations:
  a) Instruction modifications
  b) Line insertion
  c) Line deletion
  d) Copying of program statements (reading)
  e) Copying of program statements (insertion)
  f) Program instruction replacement
  g) Reassignment of position numbers
- An undo operation cancels all edit operations performed on the line where the cursor is currently placed, and restores the state present before those edit operations are performed.
- The undo function is disabled when any of the following operations is performed:
  a) Power–off
  b) Selection of another program
- Undo operation cannot be performed in any of the following states:
  a) The teach pendant is disabled.
  b) The program is write–protected.
  c) Program memory is insufficient.
- The following edit operations cannot be undone:
  a) Teaching and editing of palletizing instructions
  b) Deletion of lines including palletizing instructions
  c) Copying of lines including palletizing instructions (reading)
  d) Copying of lines including palletizing instructions (insertion)
  e) Replacement in a program including palletizing instructions
  f) Number reassignment in a program including palletizing instructions
- If the power is turned off while an undo operation is being performed, the undo operation is stopped. Note that in this case, the program may become unusable.
- If any of the following instructions is performed after an edit operation, the undo function cannot be performed:
  a) Laser instruction
  b) Palletizing instruction
  c) Spot welding instruction
  d) Line tracking instruction
- If any of the following function is executed after an edit operation, the undo function cannot be performed:
  a) Online position modification
  b) Fine adjustment of welding speed

5.5 Program Operation

This section describes the following program operations:

- Changing program information
- Deleting a program
- Copying a program
- Reading a program
- Storing a program
- Printing a program
- Displaying the attribute of a program

5.5.1 Changing program information

The program header information is changed with a program detail screen(see Section 4.1). Setting without the motion group motion group can be done. The following items can be set:

- Program name: Name of program to be changed.
- Subtype: The subtype of a program to be changed.
- Comments: The comments in the program to be changed.
- Group mask: Specifies a motion group to be controlled in a program. You can also set so a program has no motion group.
- Write protection: Prevents the modification of a program.
- Interruption disable: Causes a program that has no motion group no to be paused by the alarm whose severity is SERVO or lower, the emergency stop, and the hold.

Display the following items on the program information screen:

- Creation Date:
- Modification Date:
- Name of the file to be copied
- Positions: FALSE/TRUE
- Memory area size of program

Deleting a program

The unnecessary program can be deleted.

Copying a program

The program with another name in the same content can be reproduced.

Display of a program attribute

The following program header informations can be displayed on the program selection screen:

- Comment – The comment in a header information is displayed.
- Protection – The settings of ”Write protect:” in a header information is displayed
- Last Modified – The settings of ”Modification Date:” in a header information is displayed.
- Size – The number of lines of program and memory size are displayed.
- Copy Source – The settings of ”Copy Source:” in a header information is displayed.

> **CAUTION**
> All of the free memory size displayed on the directory screen may not be usable to store a program. Even if the size of free memory is not 0, for example, no program may be creatable.

Procedure 5–28 Changing program information

Condition ■ The teach pendant must be enabled.

Step

1 Press the MENUS key to display the screen menu.

2 Select SELECT. The program selection screen is displayed.
Alternatively, press the SELECT key to display the program selection screen.

```
Select                              JOINT 30%
            61092 bytes free          3/5
 1 SAMPLE1          JB[SAMPLE PROGRAM1    ]
 2 SAMPLE2          JB[SAMPLE PROGRAM2    ]
 3 SAMPLE3          JB[SAMPLE PROGRAM3    ]
 4 PROG001          PR[PROGRAM001         ]
 5 PROG002          PR[PROGRAM002         ]

[TYPE]  CREATE  DELETE  MONITOR  [ATTR]  >
 COPY   DETAIL   LOAD    SAVE     PRINT  >
```

3 Press NEXT, > to display the next page, then press the F2, (DETAIL) key. The program information screen is displayed.

```
COPY   DETAIL  LOAD
        F2
```

```
Program detail                   JOINT  30 %
                                         1/6
 Creation Date:              10-MAR-1994
 Modification Date:          11-MAR-1994
 Copy Source:         [****************]
 Positions: FALSE  Size:        312 Byte
  1  Program name:              [SAMPLE3 ]
  2  Sub Type:        [None              ]
  3  Comment:         [SAMPLE PROGRAM 3]
     Group Mask:                [1,*,*,*,*]
  4  Write protect:             [OFF     ]
  5  Ignore pause:              [OFF     ]
   END   PREV   NEXT
```

4 Specify each item (see Section 4.1).
If the motion instruction is taught in the program, you can not set the 3 ”Group Mask:” of this program.

5 After specifying program information, press the F1, (END) key.

```
END   PREV   NEXT
F1
```

Procedure 5–29 Deleting a program

Step

1 Press the MENUS key to display the screen menu.

2 Select SELECT. The program selection screen is displayed.

The program selection screen can also be displayed by pressing the SELECT key, instead of executing steps 1 and 2 above.

```
Select                                      JOINT 30%
                 61092 bytes free             3/5
  1 SAMPLE1          JB[SAMPLE PROGRAM1        ]
  2 SAMPLE2          JB[SAMPLE PROGRAM2        ]
  3 SAMPLE3          JB[SAMPLE PROGRAM3        ]
  4 PROG001          PR[PROGRAM001             ]
  5 PROG002          PR[PROGRAM002             ]


 [TYPE]  CREATE  DELETE  MONITOR  [ATTR]  >
```

3 Move the cursor to the name of a program to be deleted, then press the F3 DELETE key.

```
[TYPE]  CREATE  DELETE
```

F3

```
Select                                      JOINT 30%
                 61092 bytes free             5/5
  3 SAMPLE3          JB[SAMPLE PROGRAM3        ]

Delete ?

                                   YES     NO
```

4 Press the F4 (YES) key.

5 The specified program is deleted.

```
YES     NO
```

F4

```
Select                                      JOINT 30%
                 61276 bytes free             2/4
  1 SAMPLE1          JB[SAMPLE PROGRAM1        ]
  2 SAMPLE2          JB[SAMPLE PROGRAM2        ]
  3 PROG001          PR[PROGRAM001             ]
  4 PROG002          PR[PROGRAM002             ]


 [TYPE]  CREATE  DELETE  MONITOR  [ATTR]  >
```

⚠ CAUTION

Once a program is deleted, the program cannot be restored. Make sure you delete only programs that you no longer want.

Procedure 5–30 Copying a program

Step

1 Press the MENUS key to display the screen menu.

2 Select SELECT. The program selection screen is displayed.

3 Press F1 (copy) on the next page and then a program copy screen is displyed.

```
Motion Modify
1  Words
2  Upper Case
3  Lower Case
4  Options                 ---Insert---
Select

    ---Copy Teach Pendant Program---
From: [SAMPLE3          ]
  TO: [                 ]

Press ENTER for next item
 PRG    MAIN    SUB    TEST
```

4 Enter the name of the program to be copied, then press the ENTER key.

```
 ---  Copy Teach Pendant Program  ---

   From : [SAMPLE3 ]
     To : [PRG1    ]

                          -- End  --
Copy OK ?
                          YES     NO
```

5 Press the F4 (YES) key.

6 The desired program is copied to the specified program, PROGRAM1.

```
YES    NO
```

F4

```
Select                              JOINT 30%
             48956 bytes free         6/6
 1 SAMPLE1        JB[SAMPLE PROGRAM1    ]
 2 SAMPLE2        JB[SAMPLE PROGRAM2    ]
 3 SAMPLE3        JB[SAMPLE PROGRAM3    ]
 4 PROG001        PR[PROGRAM001         ]
 5 PROG002        PR[PROGRAM002         ]
 6 PRG1           JB[PROGRAM003         ]

[TYPE]  CREATE  DELETE  MONITOR  [ATTR]  >
```

Procedure 5–31 Displaying the Attribute of the Program

Step

1 Press the MENUS key. The screen menu is displayed.

2 Select ”0 — next —”. ”1 SELECT” in the next page is displayed.
You can select a program selection screen by pressing the SELECT key instead of the above 1 to 2 procedure.

Program Selection Screen

```
 Select                            JOINT  30 %
               61276 bytes free      1/4
  No.  Program name        Comment
   1   SAMPLE1          [SAMPLE PROGRAM 1]
   2   SAMPLE2          [SAMPLE PROGRAM 2]
   3   PROG001          [PROGRAM001      ]
   4   PROG002          [PROGRAM002      ]

 [ TYPE ] CREATE DELETE  MONITOR [ATTR ]>
 COPY     DETAIL  LOAD    SAVE    PRINT  >
```

3 Press F5,[ATTR].

4 Select Size.

5 The number of lines and size of a program is displayed at the place that the comments are displayed.

```
1 Comment
2 Protection
3 Last Modified
4 Size
5 Copy Source
         ATTR
```

F5 ENTER

```
 Select                            JOINT  30 %
               61276 bytes free      1/2
  No.  Program name             Size
   1   SAMPLE1          [      10/     1000]
   2   SAMPLE2          [      10/     1000]
   3   PROG001          [      10/     1000]
   4   PROG002          [      10/     1000]

 [ TYPE ] CREATE DELETE  MONITOR [ATTR ]>
```

6 When you want to display the other item,select the desired item in the procedure 4.

5.6 Background Editing

While the robot is being operated, the background editing function allows another program to be edited in the background. With this function, another program can be modified and checked without stopping robot operation, thus increasing productivity and maintenance efficiency.

> **WARNING**
>
> **This function allow editing when the teach pendant is disabled. However, when the teach pendant is disabled, any edit operations performed by an operator near the robot are very dangerous. To ensure operator safety, be sure to perform edit operation outside the robot movement range.**

Outline of this function

This function is outlined below.

- Background editing is started by selecting a special program name for background editing when the teach pendant is disabled. The special program name is ”–BCKEDT–”.
- During background editing, the following data is displayed on the top of the edit screen of the teach pendant:
  - Program name selected in the background
  - <<BACKGROUND>> for indicating that background editing is in progress

```
AAA                                    RUNNING
BBB                                  JOINT 10%
              <<BACKGROUND>>
  1:  J P[1] 100% FINE
  2:


[INST]                              [EDCMD]>
```

a: Execution status of the program selected (status line)
b: Program name selected in the background
c: Indication that background editing state is set

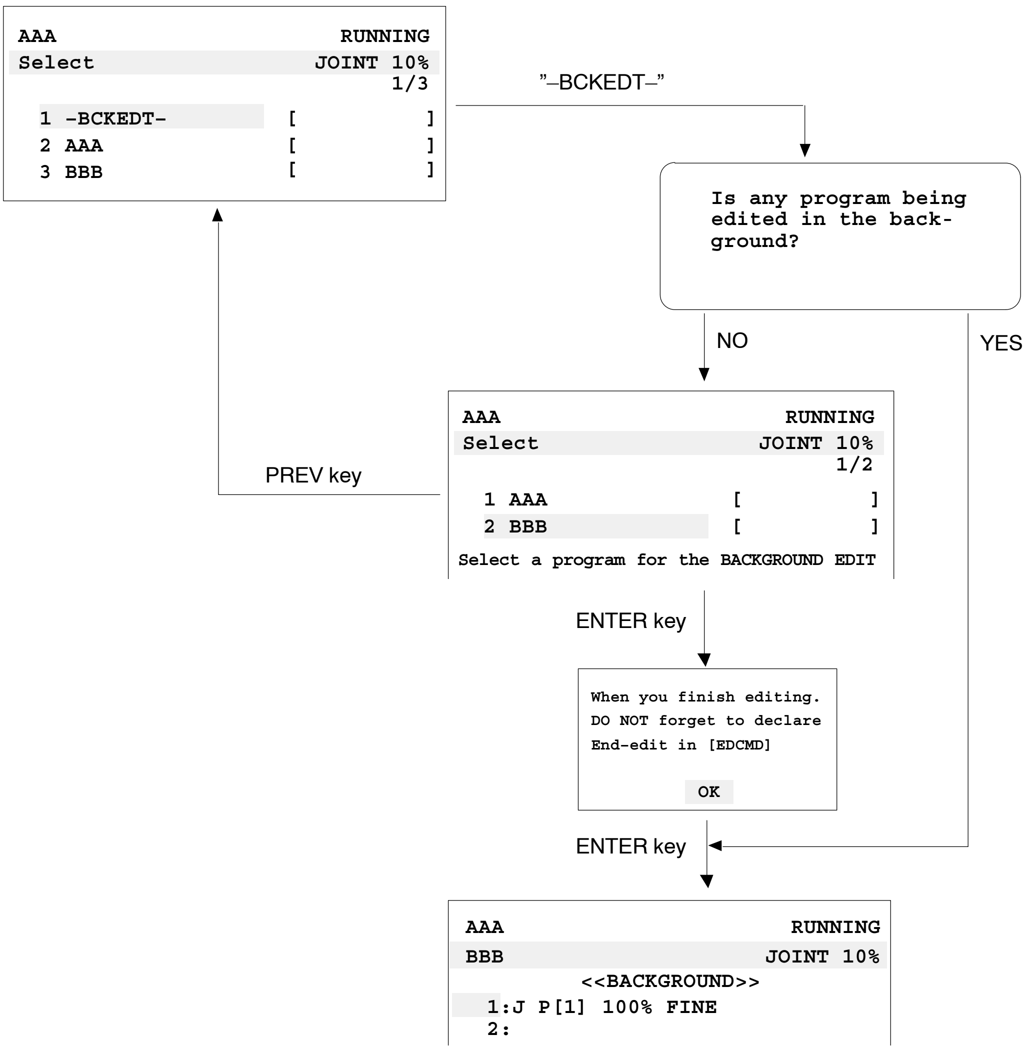
- No modifications to a program being edited in the background are reflected in the original program until the background editing is completed.
- To terminate background editing, press the F5 [EDCMD] key on the edit screen to display a menu, then select End_edit from the displayed menu.
  Here, the user can choose whether to reflect the results of background editing in the original program or discard the results of background editing.
- No multiple programs can be edited in the background at a time. The background editing of a program must be terminated by End_edit operation before another program can be edited in the background.
- If another program is selected without performing End_edit operation during background editing, the results of background editing are preserved.
  Background editing can be restarted by reselecting the special program name (”–BCKEDT–”) for background editing on the program directory screen.
- When the teach pendant is disabled, and the edit screen is displayed, the user can switch between the display of the program selected in the foreground (not background) and the display of the preserved results of background editing.
- When the teach pendant is enabled, the special program name for background editing can be selected from the program directory screen, and can be executed with the teach pendant.
- When the teach pendant is disabled, the special program for background editing cannot be externally selected and executed.
- When an external start signal is applied during background editing, the program selected in the foreground is started.
- The program started during automatic operation or executed by subprogram calling is the original program selected in the background.
- Even if a program is externally selected with the external program selection function (PNS) during background editing, the background editing can be continued without being interrupted.

The operation flows of the following cases are explained using figures below:

- When background editing is started with the teach pendant disabled
- When background editing is started with the teach pendant enabled
- When a program is externally selected during background editing
- When a start signal is externally applied during background editing
- When the teach pendant is enabled during background editing
- When the teach pendant is disabled during background editing
- When the screen is switched using the edit key on the teach pendant
- When background editing is terminated with the teach pendant disabled
- When background editing is terminated with the teach pendant enabled

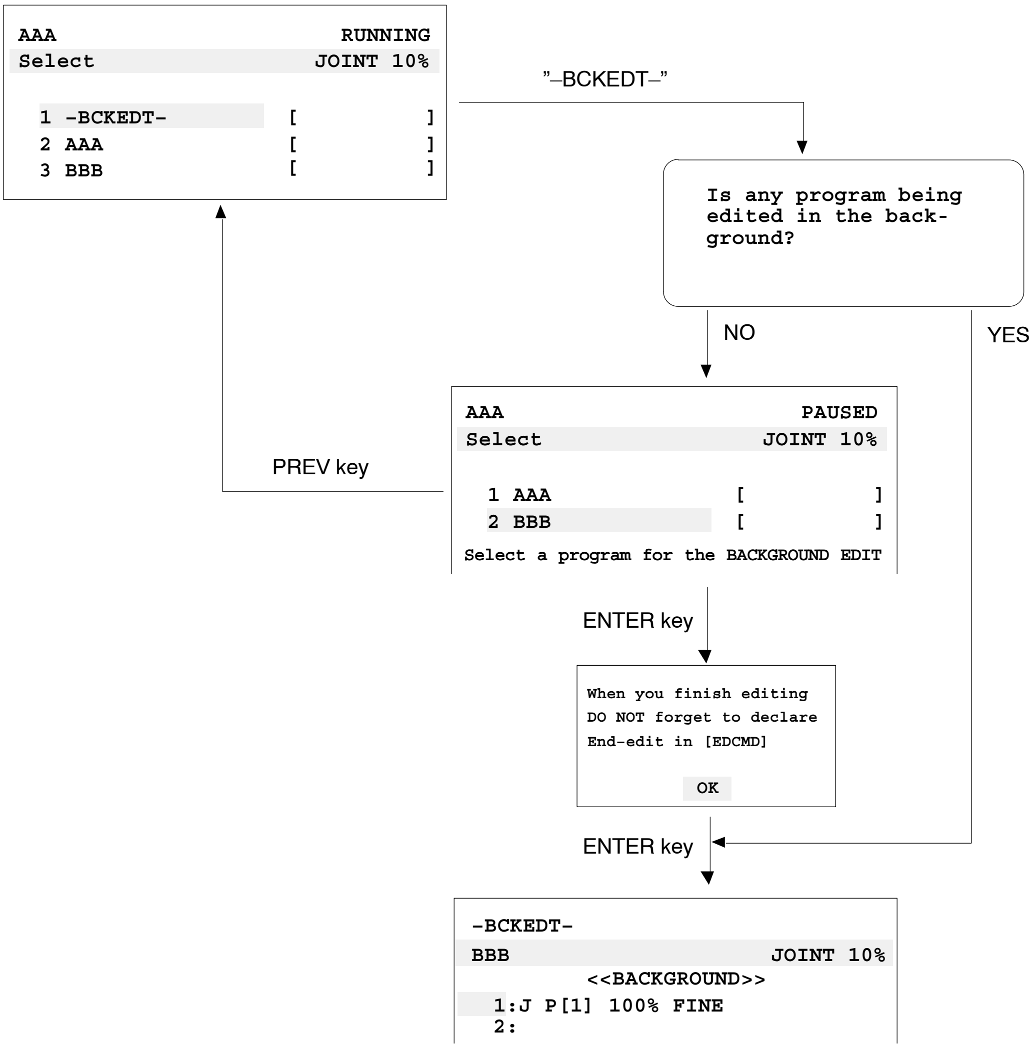
When background editing is started with the teach pendant disabled

When a program is selected in background editing, the program selected in the foreground is not modified. Even if no program is selected in the foreground, background editing is started.



When background editing is started with the teach pendant enabled

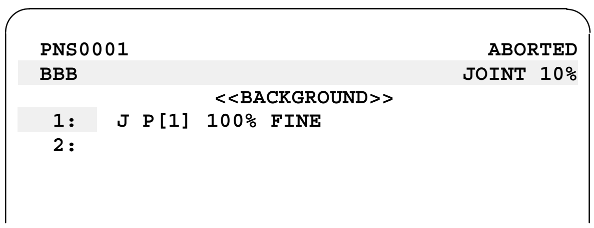
If the special program for background editing is selected when the teach pendant is enabled, the program is selected in the foreground, and its test execution is enabled.



When a program is externally selected during background editing

If a program is externally selected during background editing (with the teach pendant disabled), the status line displays the state of the selected program.

The state of background editing remains unchanged.



When a start signal is externally applied during background editing

If a start signal is externally applied during background editing (with the teach pendant disabled), the program selected in the foreground is started, and the status line displays RUNNING.

The state of background editing remains unchanged.

```
PNS0001                             RUNNING
BBB                               JOINT 10%
              <<BACKGROUND>>
  1:   J P[1] 100% FINE
  2:
```

When the teach pendant is enabled during background editing

If a program is selected in the foreground, background editing and the program being executed are suspended, and the program selected in the foreground is displayed on the screen.

If an alarm is issued from the program being executed, for example, the point of alarm generation can be immediately located and corrected by enabling the teach pendant according to this function.

To return to background editing, disable the teach pendant, then press the edit key or reselect ”–BCKEDT–” from the program directory screen.

```
PNS0001                             RUNNING
BBB                               JOINT 10%
              <<BACKGROUND>>
  1:   J P[1] 100% FINE
  2:
```

→

```
PNS0001                              PAUSED
PNS0001                           JOINT 10%

  1:
  2:
```

Disable the teach pendant, then
a. Press the EDIT key on the program edit screen.
b. Select ”–BCKEDT–” on the program directory screen.

If no program is selected in the foreground, the special program (”–BCKEDT–”) is selected to allow the program being edited in the background to be executed. The status line displays the state of ”–BCKEDT–”.

```

BBB                               JOINT 10%
              <<BACKGROUND>>
  1:
  2:
```

→

```
-BCKEDT-
BBB                               JOINT 10%
              <<BACKGROUND>>
  1:
  2:
```

When the teach pendant is disabled during background editing

If ”–BCKEDT–” is selected in the foreground, the foreground enters the program nonselection state when the teach pendant is disabled. (The status line disappears.) So, the program being edited in the background cannot be executed externally.

The background editing can be continued without modification.

```

BBB                               JOINT 10%
              <<BACKGROUND>>
  1:
  2:
```

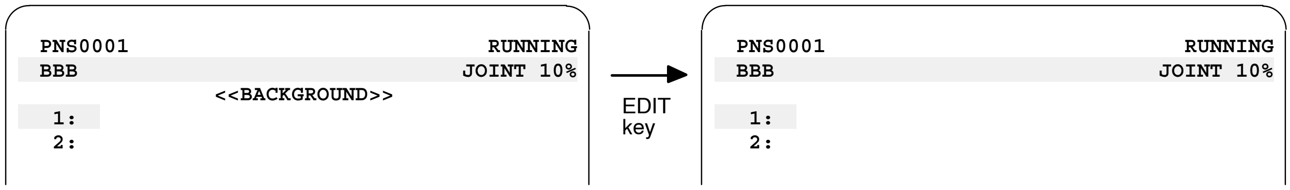
→

```
-BCKEDT-
BBB                               JOINT 10%
              <<BACKGROUND>>
  1:   J P[1] 100% FINE
  2:
```

When the screen is switched using the edit key on the teach pendant

If the teach pendant is disabled, and the program edit screen is displayed, pressing the EDIT key switches screen display between the display of the program selected in the foreground and the display of suspended background editing.

If there is a program in the foreground and background as well, the screen display switches between foreground display and background display each time the edit key is pressed, as shown below.



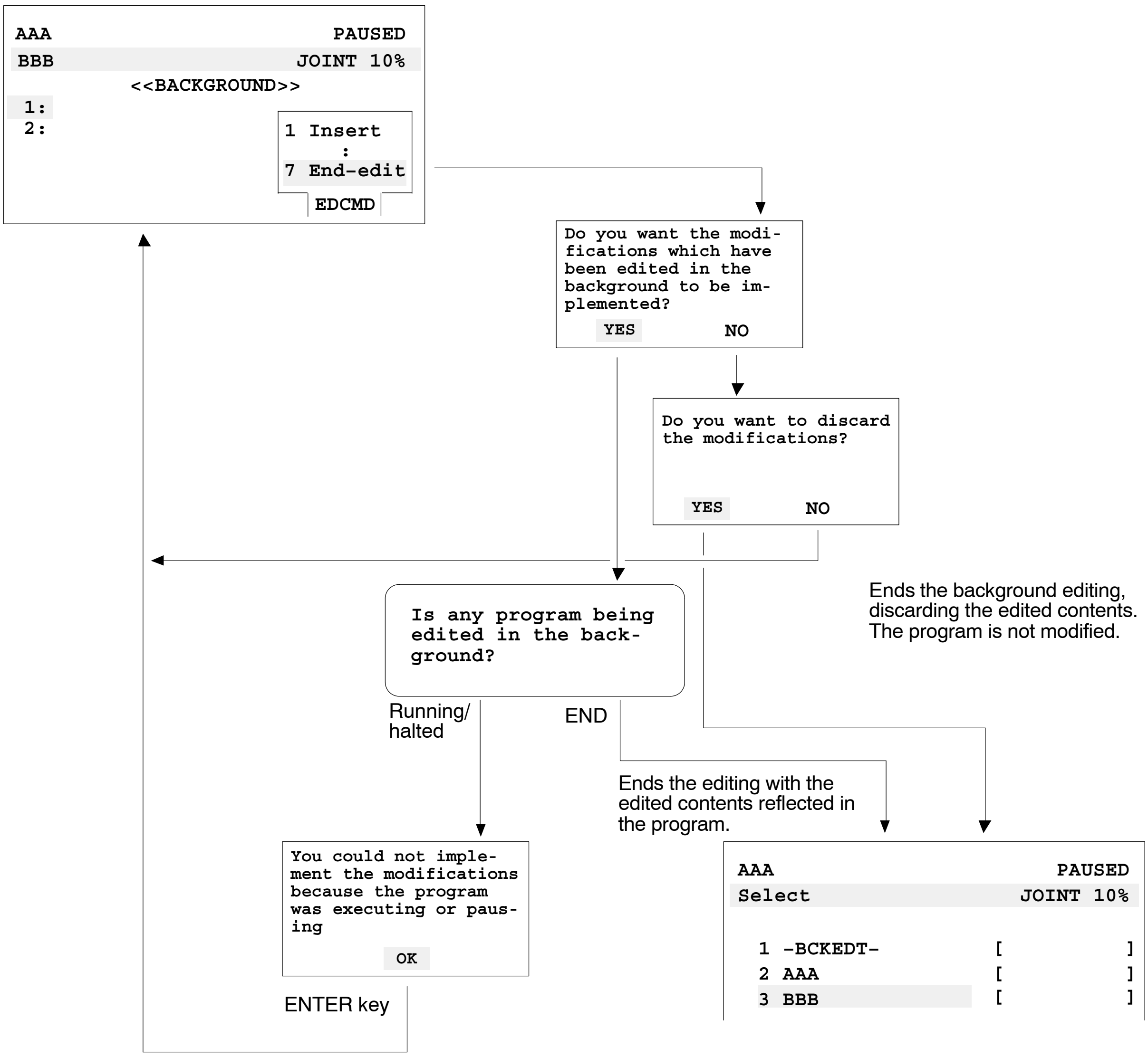
If no program is selected in the foreground, pressing the edit key does not switch screen display; the error Program is not selected occurs.

If no program is selected for background editing, pressing the edit key does not switch screen display; the error Not editing background program occurs.

This error occurs only when the background editing option is selected.

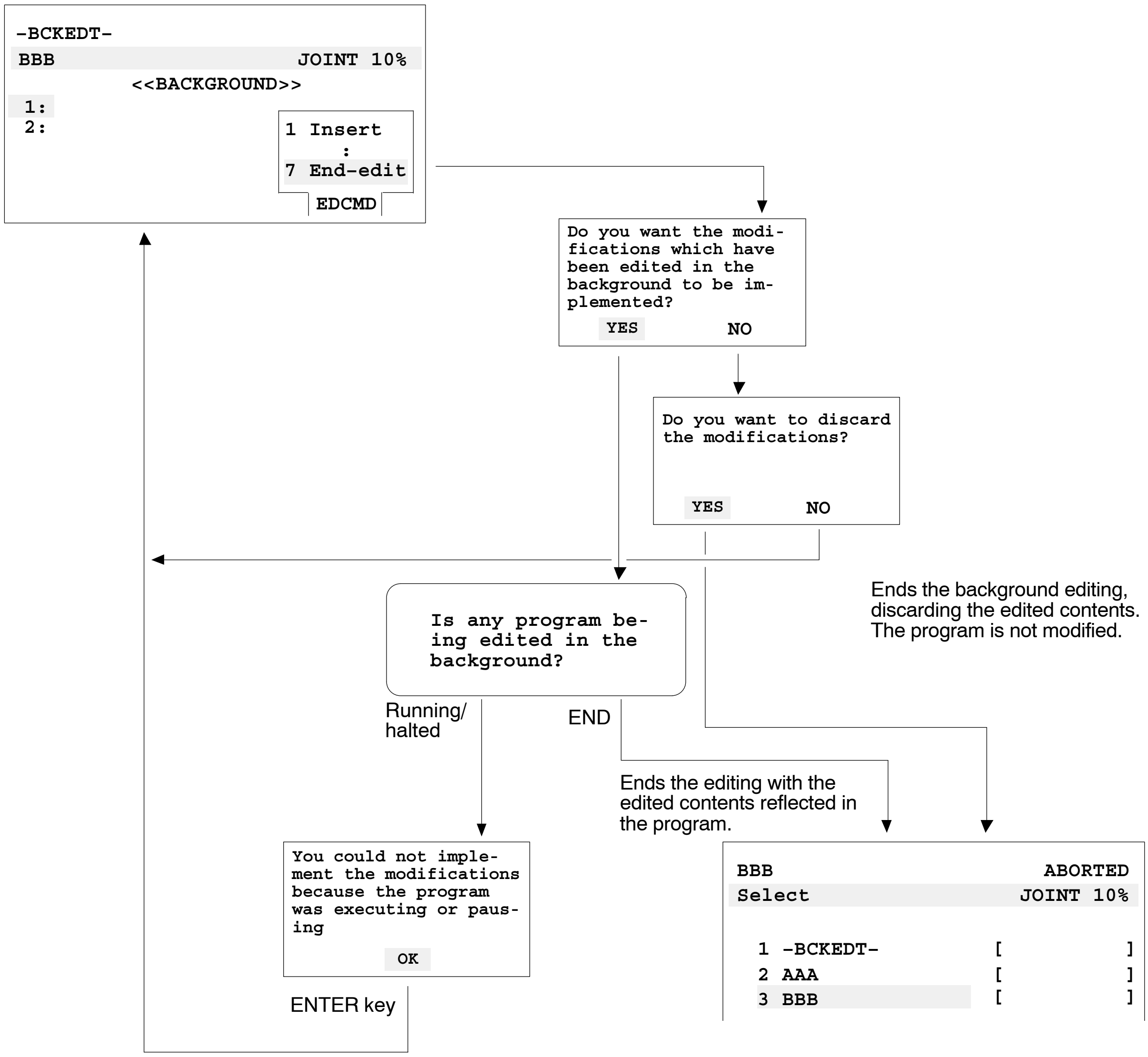
When background editing is terminated with the teach pendant disabled

When background editing is terminated, the program directory screen appears. At this time, the user can specify whether to reflect the results of background editing in the original program.



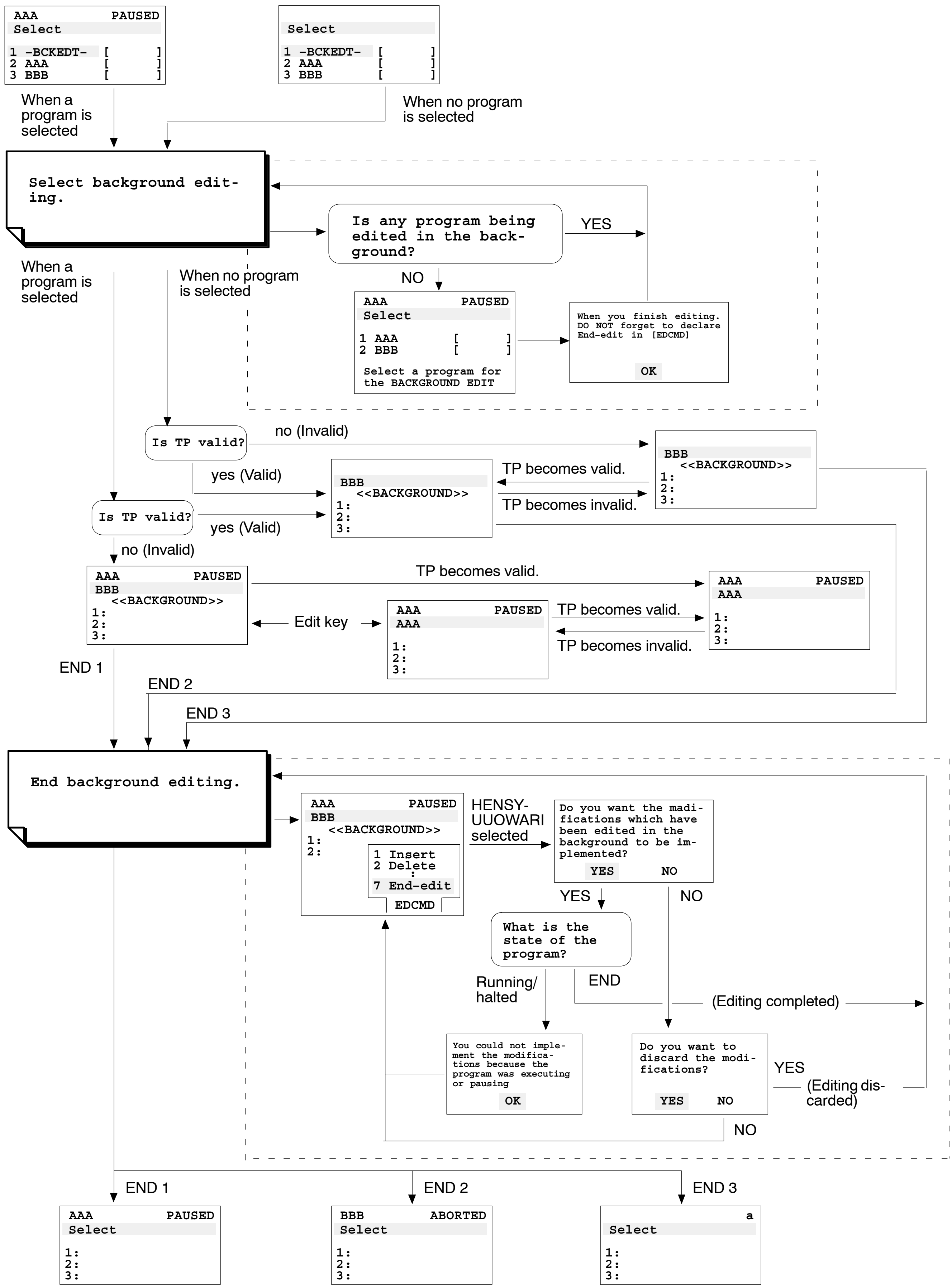
When background editing is terminated with the teach pendant enabled

When background editing is terminated, the program directory screen appears. The program edited in the background is selected in the foreground, and the status line displays the state of the program.



Operation flow

The operation flow of this function is shown on the next page.


AAA PAUSED
Select
1 -BCKEDT- [ ]
2 AAA [ ]
3 BBB [ ]
Select
1 -BCKEDT- [ ]
2 AAA [ ]
3 BBB [ ]
When a program is selected
When no program is selected
Select background editing.
Is any program being edited in the background?
YES
NO
AAA PAUSED
Select
1 AAA [ ]
2 BBB [ ]
Select a program for the BACKGROUND EDIT
When you finish editing. DO NOT forget to declare End-edit in [EDCMD]
OK
When a program is selected
When no program is selected
Is TP valid?
no (Invalid)
yes (Valid)
BBB
<<BACKGROUND>>
1:
2:
3:
TP becomes valid.
TP becomes invalid.
BBB
<<BACKGROUND>>
1:
2:
3:
Is TP valid?
yes (Valid)
no (Invalid)
AAA PAUSED
BBB
<<BACKGROUND>>
1:
2:
3:
TP becomes valid.
Edit key
AAA PAUSED
AAA
1:
2:
3:
TP becomes valid.
TP becomes invalid.
AAA PAUSED
AAA
1:
2:
3:
END 1
END 2
END 3
End background editing.
AAA PAUSED
BBB
<<BACKGROUND>>
1:
2:
1 Insert
2 Delete
:
7 End-edit
EDCMD
HENSY-UUOWARI selected
Do you want the madifications which have been edited in the background to be implemented?
YES NO
YES
NO
What is the state of the program?
Running/ halted
END
(Editing completed)
You could not implement the modifications because the program was executing or pausing
OK
Do you want to discard the modifications?
YES NO
YES
(Editing discarded)
NO
END 1
END 2
END 3
AAA PAUSED
Select
1:
2:
3:
BBB ABORTED
Select
1:
2:
3:
a
Select
1:
2:
3:

Notes

When using this function, note the points below.

- When a program is selected for background editing, the selected program is internally copied to the special program for background editing. So, memory larger than the size of a selected program needs to be allocated beforehand.
- When the background editing of a program is terminated, the original program is backed up, and the background program is reflected in the original program. So, memory larger than the size [(original program) + (increment produced by background editing)] needs to be allocated beforehand.
- If background editing cannot be terminated for a cause such as insufficient memory, the following error and its cause are displayed in the alarm display lines (line 2 and 3) on the teach pendant:
  TPIF–054 Could not end editing
  MEMO–126 No more available memory
- When the power to the robot is turned off then back on while background editing is being terminated (while the original program is being updated)
  To prevent the updating of the original program from being stopped halfway, restore the original program from the backup program when the power is turned on. If the results of background editing need to be reflected, check the results of background editing, then perform another editing termination operation.
  If an attempt to restore the original program fails, the following error is displayed:
  TPIF–055 Could not recovery original program
  In this case, check the results of background editing, then perform another editing termination operation.
  If the power is turned off then back on when editing is terminated, check the state of the original program before starting continuous operation.
- If the original program is executed when background editing is terminated, the robot may stop, depending on the timing of the execution. When terminating background editing, carefully check that the original program is not executed.
  Four cases can be considered for the timing relationship between background editing termination operation and program execution.
  - Case 1:The program is being executed when background editing is terminated.
    In this case, the message “You could not implement the modification because the program was executing or pausing” is displayed in the central part of the teach pendant, and the results of background editing cannot be reflected.
  - Case 2:The program is started exactly when the results of background editing have been reflected
    In this case, the program reflecting the results of background editing is executed.
  - Case 3:An attempt is made to start the program while the results of background editing are being reflected
    One of the following errors occurs, and the robot stops:
    SYST–011 Failed to run task
    MEMO–004 Specified program is in use
  - Case 4:When the original program is deleted, and a program is re–created to reflect the results of background editing, an attempt is made to start the program.
    One of the following error occurs, and the robot stops:
    SYST–011 Failed to run task
    MEMO–027 Specified line does not exist
- When the original program is write–protected (Write–protect is ON), editing cannot be terminated. In this case, one of the following error occurs:
  TPIF–054 Could not end editing
  TPIF–008 Memory protect violation
- Background editing can be terminated even when the special program for background editing is write–protected.
- The status line displays the execution state of a selected program. So, if a subprogram being executed is terminated forcibly, and the main program is selected in the foreground, the status line continues to display the subprogram name.
  If program start operation is initiated here, the execution of the selected main program is started, and the status line displays the execution state of the main program.

If the disabled edit key or teach pendant is enabled on the background screen in the state above, the status line does not display the subprogram but the main program selected in the foreground.

- When the teach pendant is disabled, a program can be created/deleted. However, when a program is created, the following error occurs; no selection is made in the foreground, and no direct transition to the edit screen is made:
  TPIF–104 Teach Pendant is disabled
- If the teach pendant is disabled after the special program for background editing is selected and executed with the teach pendant enabled, the end state is set.
  If the teach pendant is disabled when a subprogram is executed from the special program, the execution is terminated, and the program directory screen appears.
- When there is a suspended program in the background, the special program for background editing ("–BCKEDT–") cannot be read from the floppy disk. In this case, the following message appears:
  This program is being edited
  Before reading the special program from the floppy disk, terminate background editing.

5.7 Singular Point Check Function

If a move statement is taught, or a position modification is made based on rectangular coordinate position data when the robot is positioned near a singular point, the robot may move with an attitude different from the taught attitude when the move statement is executed. (See Section 4.3.2.)

To prevent such trouble, the singular point check function checks to see if a taught position is a singular point when the position is taught. Then, the function teaches such a position according to axial type based on the user's choice.

Function

To enable this function, set the system variable $MNSING_CHK to TRUE.

If a move statement is taught with SHIFT + POINT key or a position modification is made with SHIFT + TOUCH UP key when the robot is at a singular point, this function checks if the taught position is a singular point. This check is made when the following conditions are satisfied:

- The additional instructions do no include incremental instructions, position compensation instructions, and tool compensation instructions.
- The UF (user coordinate system number) of position data is 0.
- The registered position type is rectangular type.

If a check finds that the taught position is a singular point, the top two lines of the teach pendant display the following warning message:

TPIF–060 Can't record on cartesian (G:1)
MOTN–023 In singularity
i: Move group number at a singular point

At the same time, the following prompt message is displayed at the lower part of the teach pendant:

Record current position on joint

At this time, the function keys YES and NO are displayed. Select one of the two keys.

- YES: Deletes position data according to axial type.
- NO: Does not perform position teaching/modification.

The position data of a program that has multiple move groups is checked for singular points in ascending order of group numbers.

If multiple groups are at singular points, a warning message and prompt message are displayed for each group.

Notes

This function is not applicable to the teaching of typical palletizing loading points and passing points.

6. EXECUTING A PROGRAM

This chapter describes testing a program and automatic operation.

☐ Contents of this chapter

6.1 Program Halt and Recovery

Program halt refers to stopping a running program.

A program halt is caused by:

- An alarm occurring accidentally while the program is running.
- An intentional stop of a running program by the operator.

The operating robot stops in one of the following ways:

- Fast stop : The robot is quickly decelerated until it stops.
- Slow stop : The robot is slowly decelerates until it stops.

Program halt states are classified into two types:

- Forced termination (end): Display the termination status of a program execution. ABORTED is displayed on the screen of the teach pendant.
  If the main program is terminated while a subprogram is being executed, information on return of control to the main program is lost.

```
SAMPLE1          LINE 7            ABORTED
SAMPLE1                          JOINT 30%
```

- Halt (temporary stop): The execution of a program is stopped temporarily. PAUSED is displayed on the screen of the teach pendant.

  The temporarily stopped program can be restarted. The subprogram called with a program call instruction returns control to the main program.

```
SAMPLE1          LINE 7             PAUSED
SAMPLE1                          JOINT 30%
```

  To start from another line in the same program or another program,abort a program to release the paused state.

There are two methods to halt a program intentionally:

- Press the emergency stop button on the teach pendant or the machine operator's panel or release the deadman switch.
  Peripheral device I/O *IMSTP input
- Press the HOLD button on the teach pendant or use the input signal *HOLD of the peripheral I/O: These inputs halt the execution of the program.
- Select 1 ABORT(ALL) from the miscellaneous menu. Peripheral device I/O *CSTOPI input.
  This method aborts the program.

6.1.1 Halt by an emergency stop and recovery

To stop the robot immediately, press the emergency stop button on the machine operator's panel/box or teach pendant. In this time an emergency stop alarm occurs.

Pressing the emergency stop button causes the following:

- The robot stops operating immediately and the program is halted.
- An alarm occurs and the power to the servo system is turned off.

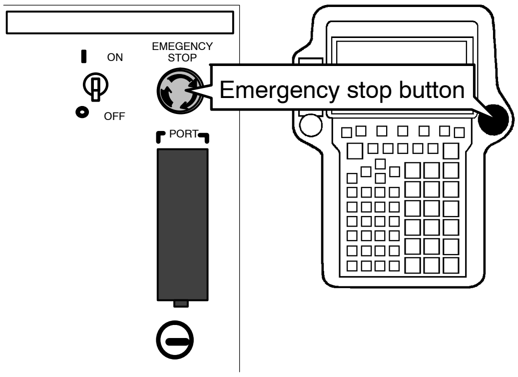
Procedure 6–1 Emergency stop and recovery

Emergency stop procedure

Step

1 Press the emergency stop button on the teach pendant or the machine operator's panel. This halts the running program, PAUSED is displayed on the teach pendant.

The emergency stop button is locked to keep it pressed (on state). The emergency stop alarm message is displayed on the screen of the teach pendant. The FAULT lamp lights.

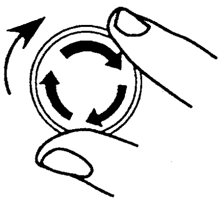



```
SRVO-002 Teach Pendant E-Stop
SAMPLE1          LINE 2              ABORTED
SAMPLE1                            JOINT 30%
```

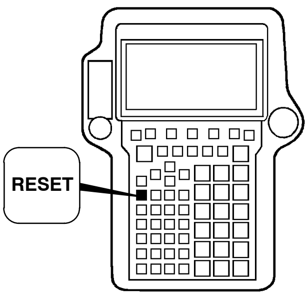
Recovery procedure

2 Eliminate the cause of the emergency stop. For example, correct the program.

3 Rotate the emergency stop button clockwise to unlock the button.



4 Press the RESET key on the teach pendant. The alarm message then disappears from the screen of the teach pendant, and the FAULT lamp goes off.



6.1.2 Halt by a hold and recovery

To decelerate the robot slowly until it stops, press the HOLD key on the teach pendant or the operator's panel.

Pressing the HOLD key causes the following:

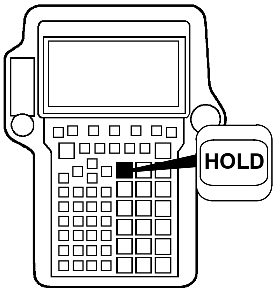
- The robot decelerates slowly until it stops (the program is halted).
- A setting can be made to cause an alarm to turn off the servo power. To make this setting, select SETUP General on the general item setting screen. (→ See Section 3.15.)

Procedure 6–2 Hold and recovery

Hold procedure

Step 1 Press the HOLD key on the teach pendant. The running program is halted, and PAUSED is displayed on the teach pendant.

The alarm message is only displayed when the halt alarm is enabled.



Recovery procedure

2 To release the halt state, restart the program.

Procedure 6–3 Terminating (aborting) a program forcibly

Abort a program

Step 1 To release the paused state and make a program aborted, press the function key to display the function menu.

2 Select ABORT(ALL). The program is aborted then the halt state is released.

```
1 ABORT (ALL)
2 Disable FWD/BWD
```

6.1.3 Halt caused by an alarm

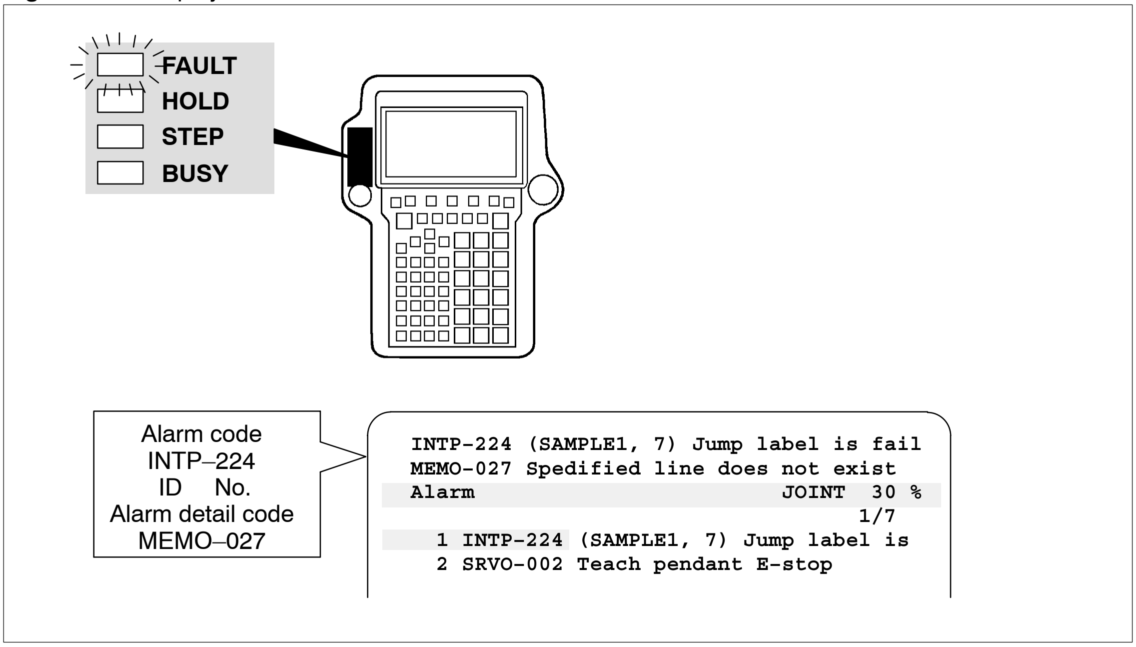
An alarm is issued when a failure is detected or when the emergency stop signal or another alarm signal is input from a peripheral device while the operator teaches or plays back a program. When an alarm is generated, it is indicated on the teach pendant, and processing such as robot operation and program execution is stopped to ensure safety.

Displaying an alarm

The operator can check whether an alarm has occurred by watching the FAULT lamps on the teach pendant and the first line and second line on the screen of the operator's panel.

The kind of a alarm is recognized by a alarm code. The cause and corrective action of a alarm can be known by a alarm code.(See APPENDIX C.1)

Figure 6–1. Display and Indication of an Alarm



Alarm history

To display the alarm history, select an alarm history screen [4 ALARM].(See APPENDIX C.1, "Alarm codes")

```
3 MANUAL FCTNS
4 ALARM
5 I/O
```

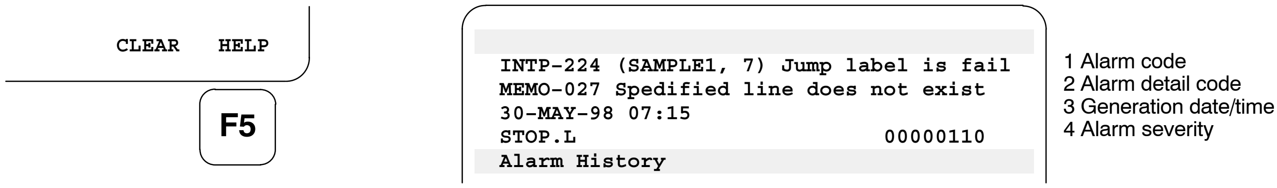
MENUS F3

```
INTP-224 (SAMPLE1, 7) Jump label is fail

INTP-224 (SAMPLE1, 7) Jump label is fail
MEMO-027 Spedified line does not exist
30-MAY-98 07:15
STOP.L                          00000110
Alarm history
                                     1/7
  1 INTP-224 (SAMPLE1, 7) Jump label is
  2 SRVO-002 Teach pendant E-stop
  3 R E S E T
  4 SRVO-027 Robot not mastered(Group:1)
  5 SYST-026 System normal power up

[ TYPE ]                 CLEAR   HELP
```

NOTE The WARN alarm history is not recorded when system variable $ER_NOHIS = 1.

Alarm detail information

Alarm has the detail information. To display the alarm detail information, press F5, HELP in the alarm history screen [4 ALARM].



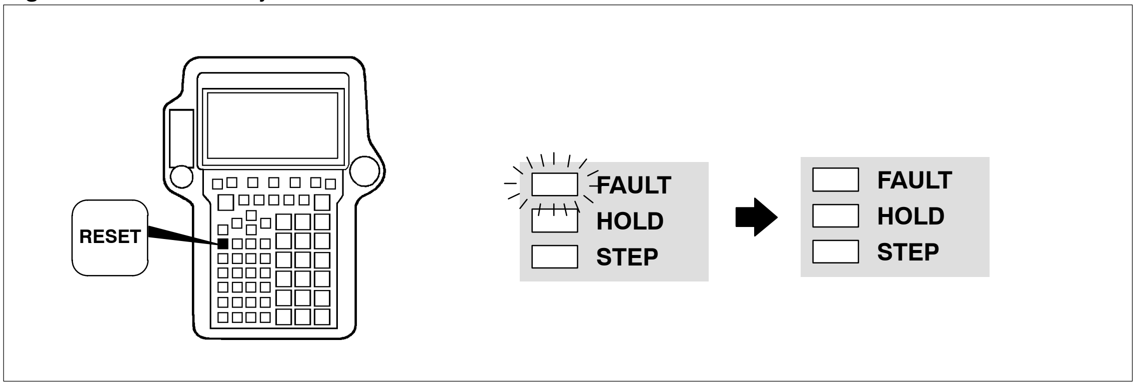
- Alarm code: Identifies an alarm.
- Alarm detail code: Identifies an alarm detail.
- Generation date: The generation date of the alarm is indicated. (It is not supported currently.)
- Alarm severity: Indicates the severity of an alarm.

Resetting an alarm

After eliminating the cause of an alarm, press the RESET key to reset the alarm. The alarm indicated in the first and second lines of the teach pendant disappears. When the servo power is turned off, it is turned on.

Resetting an alarm usually enables the robot.

Figure 6–2. RESET Key



Disabling the output of peripheral I/O alarm signals

The output of alarm signals (FAULT output) can be disabled.

- Set $ER_NO_ALM.$NOALMENBLE to 1 (enabled).
- Specify the number of alarms for which output is to be disabled in $ER_NO_ALM.$NOALM_NUM.
- Specify the codes of the alarms for which output is to be disabled in $ER_NO_ALM.$ER_CODE1 to $ER_NO_ALM.$ER_CODE10. (See Alarm code)

11 002 (Meaning: SERVO–002 alarm)
Alarm ID Alarm No.

Halt alarm

The halt alarm function issues an alarm and turns off the power to the servo system when the operator presses the HOLD key to halt the robot.

Specify the fault alarm function in [6 SETUP General] on the general item setting screen (see Section 3.15).

Alarm severity

The alarm severity indicates the severity of an alarm and the cause of the alarm. Whether program execution and robot operation are stopped, and whether the servo power is turned off depend on the alarm severity.

Table 6–1. Alarm Severity

	Program	Robot operation	Power to servo system	Range
NONE	none	none	none	———
WARN				———
PAUSE.L	pause	decelerate the robot slowly until it stops		Local
PAUSE.G				Global
STOP.L				Local
STOP.G				Global
SERVO		stop the robot immediately	off	Global
ABORT.L	abort	decelerate the robot slowly until it stops	none	Local
ABORT.G				Global
SERVO2		stop the robot immediately	off	Global
SYSTEM				Global

Range	Indicates the range in which an alarm is issued when more than one program is executed (multitasking function).	
	Local	An alarm is issued only to the program that caused the alarm.
	Global	An alarm is issued to all programs.

NOTE Some alarms do not observe the above rules.

Table 6–2. Description of Alarm Severity

Severity	Description
WARN	A WARN alarm warns the operator of a comparatively minor or unimportant failure. The WARN alarm does not affect the operation of the robot. When a WARN alarm occurs, no corresponding LED on the teach pendant or the machine operator's panel lights. To prevent a possible failure in the future, action should be taken for this alarm.
PAUSE	When a PAUSE alarm occurs, the execution of the program is halted, and the operation of the robot is stopped. Appropriate action must be taken for the alarm before the program is restarted.
STOP	When a STOP alarm occurs, the execution of the program is halted, and the robot is decelerated until it is stopped. Appropriate action must be taken for the alarm before the program is restarted.
SERVO	When a SERVO alarm occurs, the execution of a program is paused(or aborted) and the power to the servo system is turned off to stop the robot immediately. The most common cause of a SERVO alarm is hardware failure.
ABORT	When an ABORT alarm occurs, the execution of the program is forcibly terminated, and the robot is decelerated until it is stopped.
SYSTEM	A SYSTEM alarm is issued when a major system failure occurs. When a SYSTEM alarm occurs, every robot in the system is disabled. Contact the FANUC Service Division. After taking appropriate action for the alarm, turn on the power again.

6.2 Executing a Program

To execute a program is to play back a taught program. A taught program is played back just like a recorded video tape is played back.

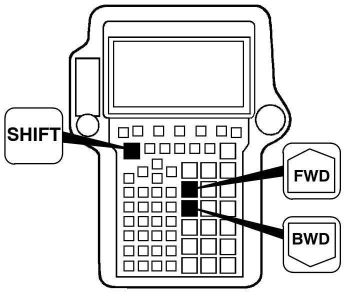
6.2.1 Starting a program

A program can be started by:

- Using the teach pendant (SHIFT key and FWD or BWD key)
- Setting the START button on the operator's panel. : option panel only
- Using the peripheral device (RSR 1 to 4 input, PROD_START input, and START input)

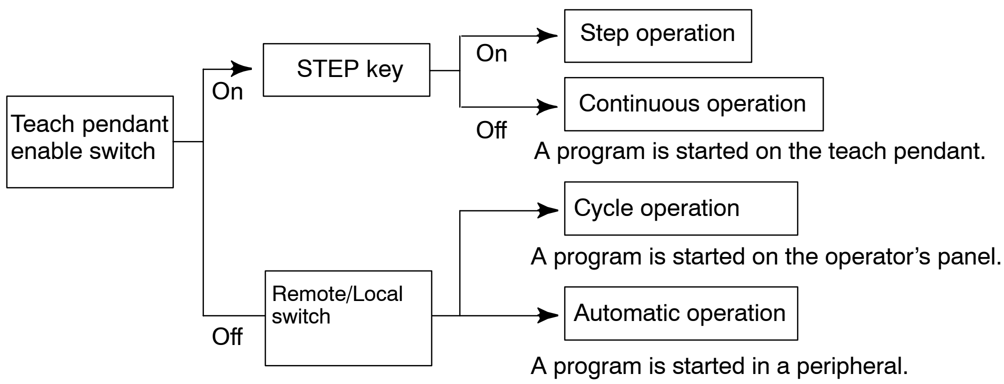
Figure 6–3. Starting a program

Teach pendant



For safety's sake, a program can be started only in a device having motion control. Motion control can be switched by using the teach pendant enable switch on the operator's panel and the remote Local mode switch.

Figure 6–4. How to Set the Right to Start a Program



CAUTION

When the start right is switched by using the enable switch on the teach pendant or the remote/Local mode switch, any programs that are currently running are temporarily halted.

6.2.2 Robot motion

The robot moves just as it is instructed by the motion instructions in the program. See Section 4.3, ”Motion Instructions”.

The following factors determine the motion of the robot:

- Feedrate override: Robot motion speed (operating speed)
- Cartesian coordinate system: Work area where the robot moves

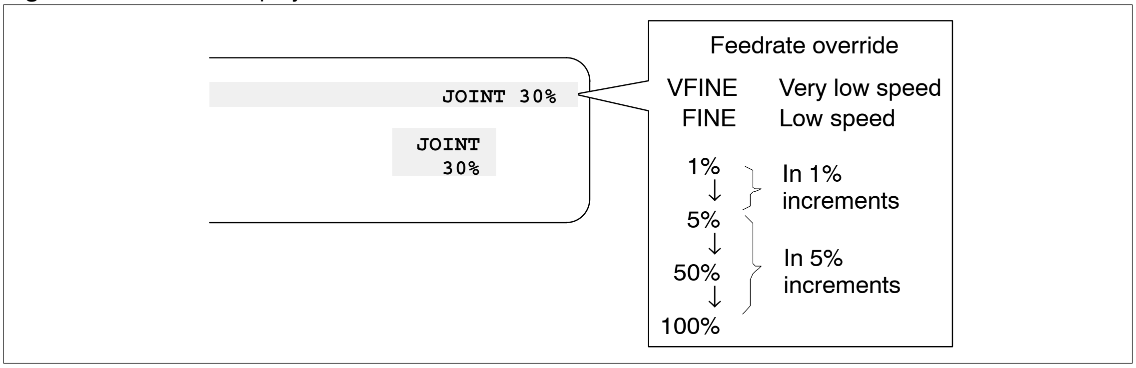
Feedrate override

The feedrate override determines the operating speed. The feedrate override is specified as a percentage of the feedrate specified in the program (programmed speed).

The current feedrate override is displayed in the upper right corner of the screen of the teach pendant, as shown in Figure 6–5.

Pressing the feedrate override key displays a popup window in reverse video in the upper right corner of the screen to call the operator’s attention. The popup window in reverse video automatically disappears a few seconds later or after another key is pressed.

Figure 6–5. Screen Display for Feedrate Override



A feedrate override of 100% would cause the robot to operate at the maximum speed specified in the current setting. Table 6–3 shows the change in feedrate override when the override key is pressed.

Table 6–3. Feedrate Override

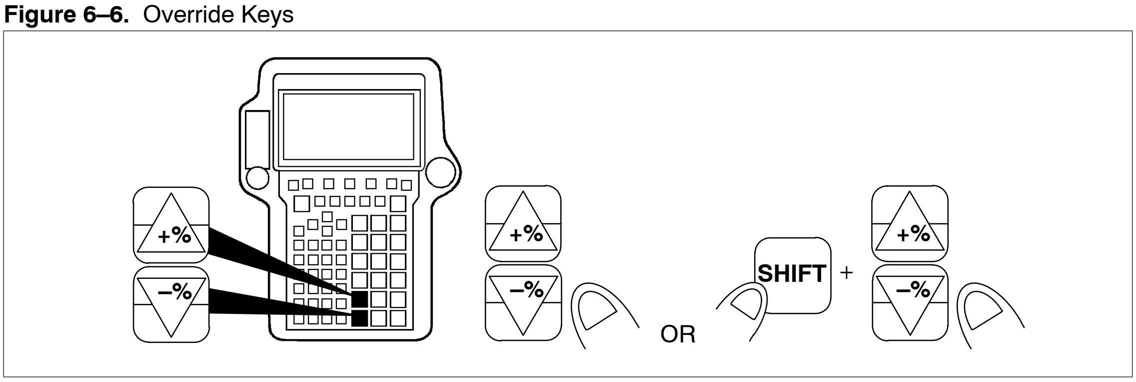
When the override key is pressed	VFINE → FINE → 1% → 5% → 50% → 100% In 1% increments In 5% increments
When the override key is pressed while pressing the SHIFT key(*1)	VFINE → FINE → 5% → 50% → 100%

*1 Enabled only when $SHFTOV_ENB is 1

To change the feedrate override, press the override key. Whenever the negative override key is pressed while the SHIFT key is pressed, the feedrate is decreased in the order: VFINE, FINE, 5%, 50%,100%. However, the feedrate is changed in this way only when system variable $SHFT OV_ENB = 1.

Note that FINE and VFINE are enabled only during a jog feed. When FINE or VFINE is specified, the robot moves at a feedrate override of 1%.

Figure 6–6. Override Keys



A feedrate override must be determined according to the condition of the machining cell, type of robot motion, and the skill of the operator. Therefore, an inexperienced robot operator should use a low feedrate override.

The feedrate override can only be increased up to the maximum value specified in $SCR.$RUNOVLIM.

When the safety speed signal (*SFSPD input) (→ see Section 3.3) is turned off, the speed override value falls to the $SCR.$FENCEOVRD value. In this case, the speed override can be increased only up to the upper limit specified in $SCR.$SFRUNOVLIM.
The system provides a function for allowing the original speed override to be restored when the safety fence is closed. (→ See Section 3.16.)

Operating speed

The operating speed is the speed at which the robot moves while the program is played back. The operating speed is obtained from the following expressions:

Figure 6–7. Operating Speed

Operating speed (joint control motion) (deg/sec) =

$$\frac{\text{Programmed feedrate}}{100} \times \frac{\text{Coefficient of a joint feedrate}}{2000} \times \text{Maximum joint feedrate} \times \frac{\text{Programmed override}}{100} \times \frac{\text{Feedrate override}}{100}$$

Operating speed (motion under path control) (mm/sec) =

$$\text{Programmed feedrate} \times \frac{\text{Programmed override}}{100} \times \frac{\text{Feedrate override}}{100}$$

Operating speed (motion under attitude control) (deg/sec) =

$$\text{Programmed feedrate} \times \frac{\text{Programmed override}}{100} \times \frac{\text{Feedrate override}}{100}$$

Programmed override $MCR_GRP.$PRGOVERRIDE (%)
Coefficient of a joint feedrate $PARAM_GROUP.$SPEEDLIMJNT

Checking a Cartesian coordinate system

When position data is played back according to Cartesian coordinates, the coordinate system number of the Cartesian coordinate system to be used is checked.

When one of the coordinate system numbers 0 to 9 is specified and the specified coordinate system number does not agree with the currently selected coordinate system number, the program is not executed.

The coordinate system number is specified for position data when the position is taught. To change a written coordinate system number, use the tool change function/coordinate system change function [option].

Tool coordinate system number (UT)

The number of a mechanical interface coordinate system or tool coordinate system is specified as a tool coordinate system number (UT). This number determines the tool coordinate system.

– 0 : The mechanical interface coordinate system is used.
– 1 to 9 : The tool coordinate system having the specified tool coordinate system is used.
– F : The coordinate system having the currently selected tool coordinate system number is used.

User coordinate system number (UF)

The number of a world coordinate system or user coordinate system is specified as a user coordinate system number (UF). This number determines the coordinate system for the work area.

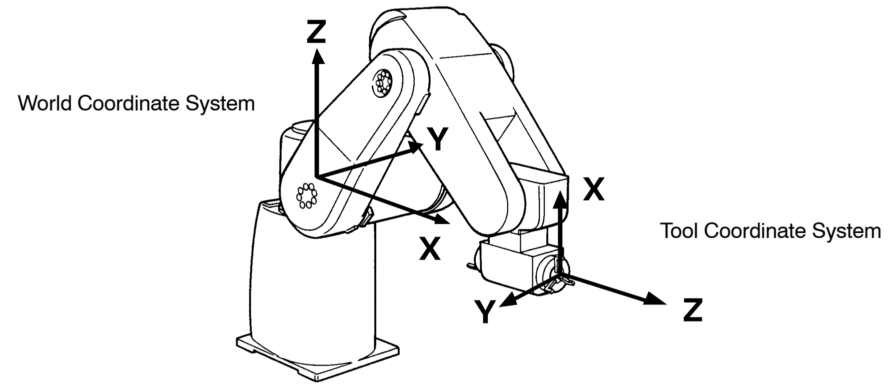
– 0 : The world coordinate system is used.
– 1 to 9 : The user coordinate system having the specified user coordinate system is used.
– F : The coordinate system having the currently selected user coordinate system number is used.

Position data information

Pressing the F5, [DETAIL] key displays position data information.

```
Position Detail                  JOINT 30%
P[1]    UF:0   UT:1   CONF:FT,0
 X: 1500.374  mm    W:  40.000  deg
 Y: -242.992  mm    P:  10.000  deg
 Z:  956.895  mm    R:  20.000  deg
EDCMD
```

Figure 6–8. Selecting a Tool Coordinate System



6.2.3 Resuming a program

Resuming a program means to restart a halted program. Before a program is halted, the system records the program. As a result, the following is possible:

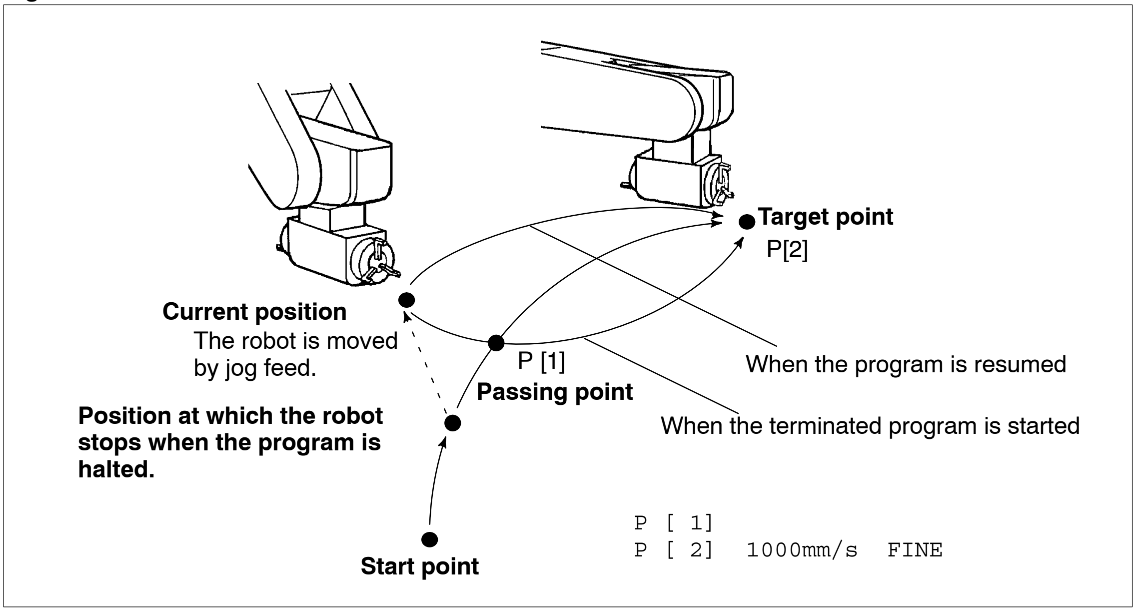
- Control can be passed to the main program called with the program call instruction.
- The path for a circular motion can be reproduced.

Path for circular motion

In circular motion, the robot moves from the current position to the target point along the path that passes through the passing point. After the robot motion is interrupted by program halt, the robot is moved by jog feed, and the program is resumed. In this case, the robot moves along a path that is similar to the one that was specified before the program was halted.
(The locus of an arc is recalculated on the assumption that the pass point is the current position after jogging, and that the start point is that used before the interruption.)

When a step test halted at the end of a circular motion is resumed after jog feed, the tool is returned to the end point of the circular motion, by means of a linear motion. (For a step test, see Section 6.3.2.) The motion is executed at the travel speed specified in the circular motion instruction.

Figure 6–9. Path for a Circular Motion



Releasing the halt state

The halt state of the program is released when:

- 1 PROGRAM ABORT is selected from the miscellaneous menu.
- Switching of the start right (→ See Section 6.2.1.)
- Creating new program when the teach pendant is enabled. For program creation, see Section 5.3.
- Selecting another program when the teach pendant is enabled. For program selection, see Section 5.4.1.

Moving the cursor in the halt state

When the cursor is moved to a desired line in the halted program and the program is to be resumed, the system asks the operator whether the program is to be resumed at the line to which the cursor has been moved. When YES is selected in response to this message, the program is halted at the line to which the cursor has been moved. When NO is selected, the cursor is returned to the line it was at before it was moved (original line), then the program is halted at that line. For both YES and NO, when the program is resumed, program execution starts at the line to which the cursor has been moved.

Procedure 6–4 Releasing the halt state

Condition ■ The program must be halted. (PAUSED is displayed on the screen.)

```
LINE 2              PAUSED
                 JOINT 30%
```

Step

1 Press the FCTN key to display the miscellaneous menu.

2 Select 1 PROGRAM ABORT. The program is terminated. (ABORTED is displayed on the screen.)

```
1 ABORT (ALL)
2
```

FCTN

```
LINE 2             ABORTED
                 JOINT 30%
```

Procedure 6–5 Moving the cursor in the halt state

Condition ■ The program must be halted. (PAUSED is displayed on the screen.)

```
LINE 2              PAUSED
                 JOINT 30%
```

Step

1 Move the cursor to the line where the program is to be resumed.

2 Restart the program.

The system asks the operator whether the program is to be resumed at the line to which the cursor has been moved.

```
  3: L  P[3] 1(
  4: L  P[4] 5(
  5: J  P[1] 1(
[End]
```

```
The cursor is on a different
line from where the program
PAUSED [2].
Are you sure you want to run
from this line ?
   YES        NO
```

3 Select YES to resume the program at the line to which the cursor has been moved. This line is then specified as the current line.

```
    YES           NO
```

```
SAMPLE1          LINE 4              PAUSED
SAMPLE1                           JOINT 30%
                                        4/6
 1: J  P[1] 100% FINE
 2: J  P[2] 70% CNT50
 3: L  P[3] 1000cm/min CNT30
 4: L  P[4] 500mm/sec FINE
 5: J  P[1] 100% FINE
[End]
```

Select NO to resume the program at the line the cursor was at before it was moved (original line). The cursor is then returned to the original line.

```
    YES           NO
```

Restart position check function

When a program is restarted in AUTO mode, this function compares the current robot position with the robot position present when the program was halted. If the comparison shows that the difference in position is beyond a set tolerance, the function issues a warning not to start the program.

If a warning is issued, select the restart method from the choices listed below. Make a choice with the teach pendant.

(1) Restart the program with no special action.

(2) Change the mode and return the robot to the stop position, then restart the program.

When restarting the program, on the restart position check screen of the setting menu, set the tolerable distance between the current robot position and the position at which the robot was halted.

```
SETUP RESUME TOL
                                      6/6

  1 Group                             : 1
  2 Enable Tolerance checking         : YES
  3 Distance Tolerance (mm)           [250.0]
  4 Orientation Tolerance (deg)       [ 20.0]

    Axes Tolerance
  5 Rotary Axes (deg)                 [ 20.0]
  6 Translational Axes (mm)           [250.0]



[ TYPE ]
```

1. Group

   For each group, you can enable or disable the restart position check function and set tolerances. Set a target group number for setting. When the restart position check function is enabled for more than one group, a warning is issued if a tolerance of one group is exceeded.

2. Enabling/disabling tolerance check

   To enable the restart position check function, select YES. (The default setting is YES.)

3. Tolerable distance (mm)

   At program restart, when the difference in distance between the current robot position and the position at which the robot was halted is greater than the value set here, a warning is issued, and the program is not started.

4. Tolerable attitude (deg)

   At program restart, when the difference in joint angle between the current robot position and the position at which the robot was halted is greater than the value set here, a warning is issued, and the program is not started.

5. Tolerance for axes: Rotation axis (deg)

   When the difference in angle between the current position of a rotation axis in the robot and the position at which the robot was halted is greater than the value set here at program restart, a warning is issued, and the program is not started.

6. Tolerance for axes: Linear axis (mm)

   When the difference between the current position of a linear axis in the robot and the position at which the robot was halted is greater than the value set here at program restart, a warning is issued, and the program is not started.

When a program is restarted, this function compares the current robot position with the position at which the robot was halted. If the comparison shows that any of the distance, attitude, and axis position data exceeds a tolerance, a warning is issued, and the program is not started. In this case, the following message appears on the teach pendant:

```
The robot position is out of
stop tolerance.
Please select action.
Choosing CONTINUE will
require cycle start.

   STOP        CONTINUE
```

(1) When STOP is selected

   When “STOP” is selected, this pop–up menu is disappeared, and the program is still paused. After select “STOP”, if start signal input, the tolerance check is executed and the pop–up menu is appeared again.

   To resume the program, please move the robot to the position within the tolerance by jog feed, then input start signal.

(2) When CONTINUE is selected

The popup menu disappears, and the program remains halted. When the start signal is input under these circumstances, the program is started.

If jog feed is performed after CONTINUE is selected, checking is made again when the program is restarted next.

CAUTION

This function cannot be used with the line tracking function and the constant joint path function at the same time.

6.3 Testing

Testing refers to checking the operation of the robot alone before automatically operating the robot in the site line. Testing the program is very important. It must be done to ensure the safety of the workers and the peripheral devices.

The following two methods can be used for testing:

- Step test: Execute the program line by line using the teach pendant or operator's panel.
- Continuous test: Execute the program from the current program line to the end of the program (up to the end–of–program symbol or program end instruction) using the teach pendant or operator's panel.

The teach pendant must be enabled before testing is performed using the teach pendant. The teach pendant is enabled when.

- The teach pendant enable switch is on.

Before test operation can be started from the operator's panel the operator's panel must be in the enabled state. The operator's panel can be placed in this state provided the following conditions are satisfied:

- The enable switch on the teach pendant is set to OFF.
- The operator's panel is the local mode.
- The peripheral device I/O *SFSPD input is on.

Before starting a program containing motion instructions, the following operation conditions must be satisfied:

- The input signal ENBL for the peripheral I/O must be on.
- An alarm must not be occurring

The typical test procedure is as follows:

1 Turn on the machine lock switch, perform step operation, and check program instructions and I/O.

2 Turn off the machine lock, and perform step operation from the teach pendant to check the robot operation, program instructions, I/O, and so forth.

3 Perform continuous operation at low speed.

4 Perform continuous operation at high speed and check the position of the robot and the operation timing.

NOTE When the enable key on the teach pendant is turned off while the shift key is held down, the program continues to run without a pause, even after the shift key is released. However, this function is enabled only when system variable $TPRUNACCEPT is set to 1. (The variable is factory–set to 0, indicating that this function is disabled.)

6.3.1 Specifying test execution

To specify test execution is to specify the requirements for test execution of a program.

```
TEST CYCLE Setup                JOINT  30 %
                                      1/7
GROUP:1
  1 Robot lock:               OFF
  2 Dry run:                  OFF
  3 Cart. dry run speed:        300.000 mm/s
  4 Joint dry run speed:         25.000 %
  5 Digital/Analog I/O:       ENABLE
  6 Step statement type:      STATEMENT
  7 Step path node:           OFF

 [ TYPE ] GROUP               ON       OFF
```

Table 6–4. Setting of test execution

ITEMS	DESCRIPTIONS
Robot lock	This function specifies whether the robot is disabled. – ON: The robot is disabled; it ignores all motion instructions. – OFF: The robot is enabled, it usually accepts motion instructions. When the robot lock function is ON, the power to the servo system is assumed to be on. Pressing the RESET key resets all the servo alarms. **NOTE** Even when the robot lock is ON, the robot can not be operated when the emergency stop button is pressed.
Dry run	When this function is enabled, the robot moves at the speed specified with “Cart. dry run speed.”
Cart. dry run speed	This parameter specifies a robot feedrate during a dry run. When the motion of the robot is under path control (linear or circular motion control), the robot constantly moves at the specified speed (unit: mm/ s).
Joint dry run speed	This parameter specifies a robot feedrate during a dry run. When the motion of the robot is under joint control, the robot constantly moves at the specified speed.
Jog dry run speed	The dry run speed (jog) indicates the robot move speed used when operation is performed with the dry run setting. When a robot motion is linear or circular, the speed indicated in this item is used from the beginning to the end of the robot motion.
Digital/Analog I/O	Digital/Analog I/O specifies whether to communicate with a peripheral device via digital I/O and group I/O signal lines or not. When this is set to disable, the robot does not send or receive the digital I/O signal with a peripheral device. Internally, all the I/O signals are given the simulated flag(S) and the simulated flag can not be released until the setting is set to enable.(See Section 6.4,”Manual I/O Control”) When you set the disable flag, the output to the peripheral device does not change. You can simulate the output without changing the state of the peripheral device. When you set the flag to enable, the output returns to the state it was in before the disable flag was set. Control of the peripheral device returns to the controller. When you set the disable flag, the input from the peripheral device to the controller is retained by the controller. When you set the flag to enable the input returns to the state it was in before the disable flag was set.
Step statement type	Step statement type specifies how to execute a program in single step mode. – STATEMENT : The program execution is paused at each line. – MOTION : The program execution is paused at every motion instruction. – ROUTINE : Almost the same as STATEMENT, however, the pause is not done in a program that is called by a CALL instruction. – TP & MOTION : At all KAREL instruction except for motion instructions, a program does not pause. **NOTE** ”TP & MOTION” is not used currently.
Step path node	When ”Step path node” is set to be ON, the robot pauses at every node during execution of the KAREL instruction,”MOVE ALONG”.

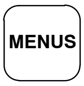
Procedure 6–6 Specifying test execution

Step

1 Press the MENUS key to display the screen menu.

2 Select 2 TEST CYCLE. The test cycle screen is displayed.

```
1 UTILITIES
2 TEST CYCLE
3 MANUAL FCTNS
```




```
TEST CYCLE Setup              JOINT  30 %
                                      1/7
GROUP:1
  1 Robot lock:            OFF
  2 Dry run:               OFF
  3 Cart. dry run speed:     300.000 mm/s
  4 Joint dry run speed:      25.000 %
  5 Digital/Analog I/O:    ENABLE
  6 Step statement type:   STATEMENT
  7 Step path node:        OFF

[ TYPE ] GROUP             ON      OFF
```

3 Specify requirements for test execution.

4 To change the group number, press F2 GROUP.

6.3.2 Step test

To perform a step test (step operation) is to execute the program line by line. After one line of the program is executed, the program is halted. After executing a logic instruction, the next line becomes the current line and the cursor moves to the next line, but for the motion instruction, the cursor stays at the line that execution is completed.

Specifying the step mode (single step)

To specify the step mode, press the STEP key on the teach pendant. When the step mode is specified, the STEP LED on the teach pendant is lit. The STEP LED is off when continuous operation is specified.

Figure 6–10. STEP Key

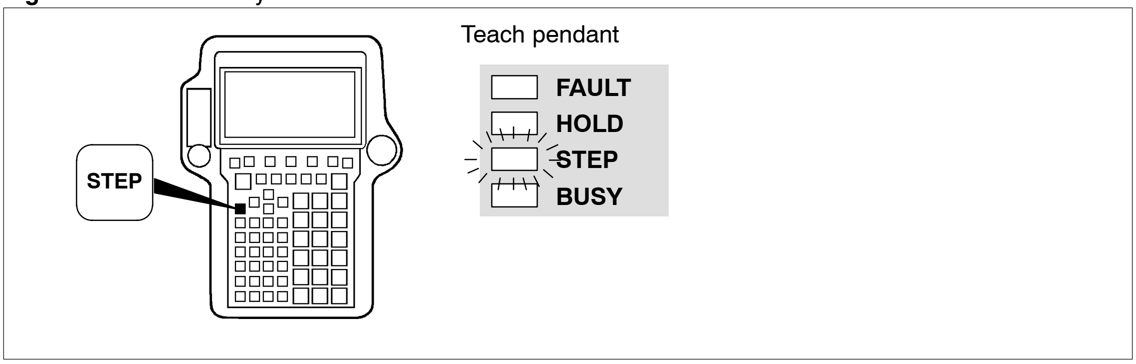
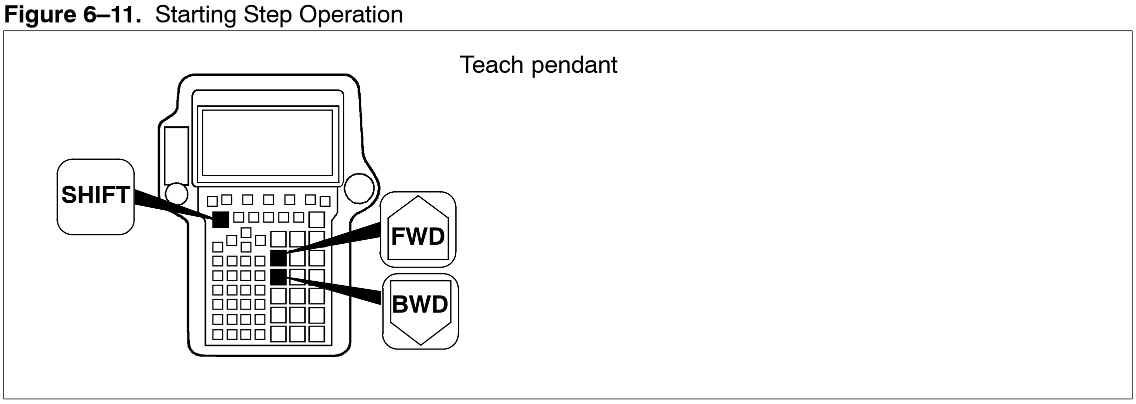


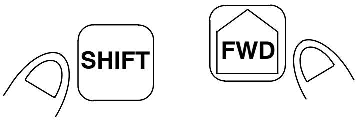
Figure 6–11. Starting Step Operation



Step operation can be performed in two ways: Forward execution and backward execution.

Forward execution

In forward execution, the program is executed in normal order. To perform forward execution of the program, press and hold down the SHIFT key, then press and release the FWD key on the teach pendant.



When a program is started, the program is executed for one line pointed to by the cursor, then the program is halted.

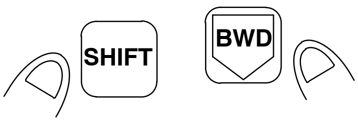
When a motion instruction is executed, the cursor is held at the executed line. When a logic instruction is executed, the cursor is moved to the next line.

Each time forward execution of the program is started, the next line of the program is executed.

When executing the circular motion instruction in step mode, the robot pauses near the through position on an arc. Moreover, if the robot is paused just before the through position, the robot does not stop at the through position after resuming a program.

Backward execution

In backward execution, the program is executed in reverse order. To perform backward execution of the program, press and hold down the SHIFT key, then press and release the BWD key on the teach pendant.



- During backward execution, only the motion instructions can be executed. However, a skip instruction forward execution instruction, backward execution instruction, soft float instruction, and other optional move instructions are ignored while the program is executed. After one line of the program is executed, the cursor is moved to the previous line.
- The instruction before the line where the following program instructions is taught can not be executed in backward execution. When you execute these instructions in backward execution, the cursor moves to the line following the line that contains taught these instructions:
  - Halt instruction (PAUSE)
  - Abort instruction (ABORT)
  - Program end instruction (END)
  - Jump instruction (JMP LBL[ ])
  - User alarm instruction (UALM[ ])
  - Execution instruction (RUN)
- The following program instructions cannot be executed:
  - Incremental instruction (INC)
- A blank line does not affect the execution of the program (Both Forward and Backward execution)

When the terminated program is restarted, the motion instruction in the line pointed to by the cursor is executed, then the program is halted.

Each time backward execution of the program is started, the program is executed using the motion format and feedrate specified in the current line, and the position data and positioning path of the motion instruction in the previous line.

- When the motion instruction in the current line specifies a circular motion, the robot moves to the target point along the path which passes through the passing point (Start point of an arc motion in normal program execution) specified in the current line.
- When the motion instruction in the previous line specifies a circular motion, the robot moves to the destination position specified in the previous line using the motion format and feedrate specified in the current line.

To disable backward execution of the program while the program is being executed, insert the halt instruction (PAUSE) into the desired location. After the halt instruction is executed, the cursor returns to the position it was at before the program was executed.

When the halt instruction is specified in the line before the line at the cursor is pointed at, backward execution of the program is disabled. To restart backward execution of the program, move the cursor to the line before the line that contains the halt instruction (two lines before the line at which the cursor is pointed).

Inter–program reverse program execution

With the inter–program reverse operation function, control can be returned from a subprogram to the main program that called the subprogram by performing reverse operation (SHIFT + BWD).

NOTE Even if a subprogram exists during reverse operation of a main program, the subprogram cannot be called.

NOTE When program termination occurs within a subprogram, control cannot be returned to the main program.

When reverse execution is performed from a subprogram to the main program, the cursor stops at the line of the instruction that calls the subprogram taught in the main program.

Sample program

Example: When reverse operation is performed starting from the fourth line of a subprogram

```
Main_Prg
1:
2:R[1]=R[1]+1
3:J P[1] 100% FINE
4:IF R[1]=100, JMP LBL[100]
5:CALL Sub_Prog
6:
       .
       .
[End]

Sub_Prog
1:DO[1]=ON
2:DO[2]=ON
3:L P[2] 1000mm/sec FINE
4:L P[3] 1000mm/sec FINE
[End]
```

1 Start reverse operation with the cursor positioned to the fourth line of the subprogram.
2 Reverse operation (SHIFT + BWS) from P[3] to P[2]. The cursor is positioned to the third line of the subprogram.
3 Reverse operation (SHIFT + BWS) to the fifth line of the main program (CALL SUBPROGRAM). The cursor is positioned to the fifth line of the main program.
4 Reverse operation (SHIFT + BWS) from P[2] to P[1]. The cursor moves from the fifth line to third line of the main program.

Program end in backward execution

If the system variable $BWD_ABORT is set to TRUE, when the first line of the program is finished to be executed during the backward execution, this program ends.

Procedure 6–7 Step test

Condition

- The teach pendant must be enabled.
- The single–step mode must be set.
- The system must be in the operation enable state.
- No one must be in the operating area. No obstacle must be placed in the operating area.

Step

1 Press the SELECT key. The program selection screen is displayed.

2 Select the program to be tested and press the ENTER key. The program edit screen is displayed.

3 Press the STEP key to select the step mode. The STEP LED lights. (Check that the STEP LED lights when the STEP key is pressed.)

4 Move the cursor to the program start line.

5 Press and hold down the deadman switch, then turn on the teach pendant enable switch.

> **WARNING**
>
> **The execution of the program instructions starts in the next step. The execution causes the robot to make a motion, which may produce unpredictable results. The operator should check that no persons and no unnecessary equipment is in the work area and that each part of the protective fence is sound. Otherwise, injury or property damage would occur. If the program needs to be stopped before it terminates, the operator should release the SHIFT key or deadman switch or press the HOLD or emergency stop button.**

6 Start the program.

- To perform forward execution of the program, press and hold down the SHIFT key, then press and release the FWD key. Do not release the SHIFT key until execution of the program is completed.
- To perform backward execution of the program, press and hold down the SHIFT key, then press and release the BWD key. Do not release the SHIFT key until execution of the program is completed.

7 After one line of the program is executed, the program is halted.

- When a motion instruction is executed, the cursor stops at the executed line. The next time forward execution of the program is performed, the next line of the program is executed.
- When a control instruction is executed, the cursor moves to the next line.

8 To release the step mode, press the STEP key.

9 Turn off the teach pendant enable switch, then release the deadman switch.

6.3.3 Continuous test

To perform a continuous test is to execute the program in the normal order from the current program line to the end of the program (end–of–program symbol or the program end instruction). Backward execution of the program is disabled during a continuous test.

A continuous test can be started using the teach pendant or operator’s panel.

To perform a continuous test using the teach pendant, press and hold the SHIFT key, then press and release the FWD key. The program is then executed from the current line.

To start continuous test operation (cycle operation) from the operator’s panel/box, momentarily press the start button on the operator’s panel. Program execution then starts from the current line.

NOTE The continuous text execution can be executed in the forward direction only.

Procedure 6–8 Continuous test (using the teach pendant)

Condition

- The teach pendant must be enabled.
- The continuous mode must be set. (The STEP lamp must be off.)
- The system must be in the operation enable state.
- No one must be in the operating area. No obstacle must be placed in the operating area.

Step

1 Press the SELECT key. The program selection screen is displayed.

2 Select the program to be tested and press the ENTER key. The program edit screen is displayed.

3 Set the continuous mode. Check that the STEP LED is off. (If the STEP lamp is on, press the STEP key to turn it off.)

4 Move the cursor to the program start line.

5 Press and hold down the deadman switch, then turn on the teach pendant enable switch.

WARNING
The execution of the program instructions starts in the next step. The execution causes the robot to make a motion, which may produce unpredictable results. The operator should check that no persons and no unnecessary equipment is in the work area and that each part of the protective fence is sound. Otherwise, injury or property damage would occur. If the program needs to be stopped before it terminates, the operator should release the SHIFT key or deadman switch or press the HOLD or emergency stop button.

6 Press and hold down the SHIFT key, then press the FWD key. Hold down the SHIFT key until the execution of the program is completed. When the SHIFT key is released, the program is halted.

NOTE When the enable key on the teach pendant is turned off while the shift key is held down, the program continues to run without a pause, even after the shift key is released. However, this function is enabled only when system variable $TPRUNACCEPT is set to 1. (The variable is factory–set to 0, indicating that this function is disabled.)

The program is executed to the end, then forcibly terminated. The cursor is returned to the first line of the program.

Procedure 6–9 Continuous test operation (started from the operator's panel) –option–

Condition

- The operator's panel must be in the enabled state.
- Continuous operation mode must be set. (The step lamp must not be lit.)
- The system must be ready for operation.
- Nobody must be within the work area. There must be no obstacles.

Step

1. Press the select key. The program list screen is selected.
2. Select a program to be tested, and press the enter key. The program edit screen appears.
3. Set continuous operation mode. Check that the step lamp is not lit. (If the STEP lamp is on, press the STEP key to turn it off.)
4. Position the cursor to the first line.
5. Place the system in local mode. (For how to switch to local mode, see the description of Remote/Local setting in Section 3.16, "SYSTEM CONFIG MENU."

> **WARNING**
> **The execution of the program instructions starts in the next step. The execution causes the robot to make a motion, which may produce unpredictable results. The operator should check that no persons and no unnecessary equipment is in the work area and that each part of the protective fence is sound. Otherwise, injury or property damage would occur. If the program needs to be stopped before it terminates, the operator should release the SHIFT key or deadman switch or press the HOLD or emergency stop button.**

6. Press the start button on the operator's panel. Program execution is performed up to the end of the program then terminated forcibly. The cursor returns to the first line of the program.

6.3.4 Program look/monitor

When the program is executed, the screen of the teach pendant becomes a monitor screen by which the execution of the program is displayed. In the monitor screen, the cursor moves to follow the line which is executed and you can not edit a program.

Program Monitor Screen

```
PROGRAM1          LINE 1          RUNNING
PROGRAM1                      JOINT  30 %
                                     1/10
    1:J P[1] 100% FINE
    2:J P[2] 100% FINE
    3:J P[3] 100% FINE
    4:J P[4] 100% FINE
    5:J P[5] 100% FINE
    6:J P[6] 100% FINE

          LOOK
```

Press F2,LOOK,then the program looking screen is displayed and the cursor of the program which is being executed stops(Program is kept to be executing). You can look at the desired part except the line which is executed with the arrow keys.

Program looking screen

```
PROGRAM1          LINE 8          RUNNING
PROGRAM1                      JOINT  30 %
                                     1/10
    1:J P[1] 100% FINE
    2:J P[2] 100% FINE
    3:J P[3] 100% FINE
    4:J P[4] 100% FINE
    5:J P[5] 100% FINE
    6:J P[6] 100% FINE
  Under the LOOK mode
        MONITOR
```

The message ”Under the LOOK mode” is highlighted at the prompt line while looking at the program. To return to the monitor screen, press F2,MONITOR. When the monitor screen is displayed,the cursor specifies the line which is executed at that time.

If the execution of the program is paused or ended,the program edit screen is displayed in place of the program looking screen.

Program edit screen

```
PROGRAM1          LINE 6          ABORTED
PROGRAM1                      JOINT  30 %
                                     6/10
    1:J P[1] 100% FINE
    2:J P[2] 100% FINE
    3:J P[3] 100% FINE
    4:J P[4] 100% FINE
    5:J P[5] 100% FINE
    6:J P[6] 100% FINE

 POINT                           TOUCHUP>
```

6.4 Manual I/O Control

Under manual I/O control, signals are transmitted between the robot and peripherals before the program is executed.

The manual I/O control refers to the following items:

- Forced output
- Simulated output and simulated input
- Wait instruction

6.4.1 Forced output

Forced output is to manually turn digital output signals on or off. For the group output and the analog output, specify the value.

Procedure 6–10 Forced output

Condition
- Assignment of the signals to be output must be completed.

Step
1. Press the MENUS key to display the screen menu.
2. Select 5, I/O. The I/O screen is displayed.

Manual forced digital output

3. Press the F1, [TYPE] key to display the screen change menu.
4. Select Digital. The digital output screen or digital input screen is displayed.
   If the input screen is displayed, press the F3, (IN/OUT) key to change the input screen to the output screen.

```
4 ALARM
5 I/O
6 SETUP
MENUS

Digital
[TYPE]
F1
```

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

[TYPE]  CONFIG  IN/OUT   ON      OFF
```

WARNING
Forced output activates connected equipment. Before executing the forced output, the operator should check which equipment is connected to the digital output and what operation the forced output would cause. Otherwise, injury or property damage could occur.

5. Move the cursor to the status field for the signal number to be changed, then press the F4 (ON) or F5 (OFF) key to change the signal output setting.

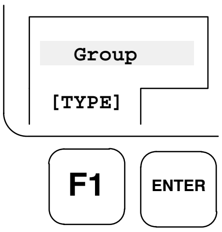
```
IN/OUT   ON      OFF
F4
```

```
I/O Digital Out                   JOINT 30%

  DO[1]    U   ON
  DO[2]    U  OFF
```

Manual forced group output

6 Press F1, [TYPE]. The screen change menu is displayed.

7 Select Group. The group output screen is displayed.




```
 I/O Group Out                      JOINT  30 %
       #   SIM VALUE
   GO[   1]  U      1 [             ]
   GO[   2]  U     10 [             ]
   GO[   3]  U     23 [             ]
   GO[   4]  *      * [             ]
   GO[   5]  *      * [             ]
   GO[   6]  *      * [             ]
   GO[   7]  *      * [             ]
   GO[   8]  *      * [             ]
   GO[   9]  *      * [             ]

 [ TYPE ] CONFIG  IN/OUT  FORMAT
```

8 Move the cursor to the setting field of the signal number you want to change, enter the value. Pressing F4,FORMAT toggles between the decimal expression and the hexadecimal expression.

```
 I/O Group Out                      JOINT  30 %
       #   SIM VALUE
   GO[   1]  U      3 [             ]
   GO[   2]  U     10 [             ]
```

6.4.2 Simulated I/O

The Simulated I/O function changes the state of signals internally without making digital, analog or group I/O communicate with peripherals. This function is used to execute the program or to test the I/O instruction when connection of I/O with peripherals is not completed. Simulated input/output can be used for digital, analog and group I/O.

To enable simulated input/output, set the simulated flag, S.

Simulated output

The simulated output function internally changes the signal state using the I/O instruction of the program or manual output, but does not change the state of output to peripherals. This function holds the state of output to peripherals when the simulated flag is set.

When the simulated flag is reset, the output is restored to the original state.

Simulated input

The simulated input function internally changes the signal state with the I/O instruction of the program or manual input. The state of input from peripherals is ignored, and the signal state is not changed internally.

When the simulated flag is reset, the input enters the current state.

Refer to 6.3.1,”Specifying test execution” to specify whether I/O signal is disable in the test execution.

Procedure 6–11 Simulated input / output

Condition ■ The input/output signal has been allocated.

Step

1 Press the MENUS key. The screen menu is displayed.

2 Select I/O. The I/O screen is displayed.

3 Press F1, [TYPE]. The screen change menu is displayed.

4 Select Digital. Digital I/O screen is displayed.

```
 4 ALARM
 5 I/O
 6 SETUP
```

MENUS

```
  Digital
 [TYPE]
```

F1

```
I/O Digital In                      JOINT 30%
                                        1/168
      #   SIM  STATUS
 DI[1]    U   OFF  [digital signal 1  ]
 DI[2]    U   OFF  [digital signal 2  ]
 DI[3]    U   OFF  [digital signal 3  ]
 DI[4]    U    ON  [digital signal 4  ]
 DI[5]    U    ON  [digital signal 5  ]
 DI[6]    U   OFF  [digital signal 6  ]
 DI[7]    U   OFF  [digital signal 7  ]
 DI[8]    U    ON  [digital signal 8  ]
 DI[9]    U    ON  [digital signal 9  ]

[TYPE]  CONFIG  IN/OUT    ON        OFF
```

5 Move the cursor to the SIM field for the signal number to be changed and press the F4 (S) or F5 (U) key to change the simulated setting.

```
 I/O Digital In

  DI[1]    U    OFF

IN/OUT SIMULATE  UNSIM
```

F4

```
 I/O Digital In                      JOINT 30%

  DI[1]    S   OFF  [digital signal 1  ]

[TYPE]  CONFIG  IN/OUT SIMULATE  UNSIM
```

6 Move the cursor to the status field for the number of the signal to be simulated output and press the F4 (ON) or F5 (OFF) to change the simulated output setting.

```
 I/O Digital In

  DI[1]    U    OFF

IN/OUT    ON        OFF
```

F4

```
 I/O Digital In                      JOINT 30%

  DI[1]    S   ON   [digital signal 1  ]

[TYPE]  CONFIG  IN/OUT    ON        OFF
```

6.4.3 Standby release

When a standby instruction in a program waits until the I/O conditions are satisfied, the standby release function skips this instruction, and halts program execution at the next line. Standby release is enabled only when a program is being executed.

Standby release is performed by choosing from the miscellaneous function menu.

Procedure 6–12 Standby release

Condition ■ Program execution is currently in the I/O wait state.

```
Sample3                              JOINT 30%
                                         11/20
10:  J P[5] 100% FINE
11:  WAIT RI[5]=ON
12:    RO[1]=ON
```

Step

1 Press the function key to display the miscellaneous function menu.

2 Select 7 RELEASE WAIT.
The I/O wait is skipped, and the cursor moves to the next line. The program is then halted.
When program execution is restarted, the next instruction is executed.

```
7 RELEASE WAIT
```

FCTN INPUT

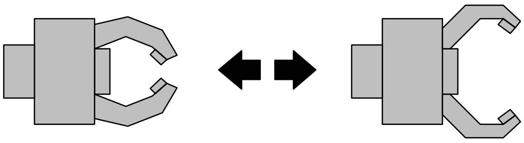
6.5 Operating the Hand Manually

To operate the hand manually using the teach pendant, hand instruction must be assigned to the manual operation screen when macro instructions are set.

When teaching the operation of the hand, check whether the hand can actually hold a workpiece at the target position by operating the hand.

The macro instructions are optional functions.

Figure 6–12. Operating the Hand Manually



Procedure 6–13 Operating the hand manually

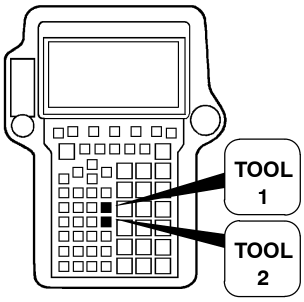
Condition

- The teach pendant must be enabled.
- No one must be in the operating area. All obstacles must be removed from the operating area.
- The hand instruction must be defined as a macro for manual operation (MF).

Step

1 Press and hold down the deadman switch, then turn on the teach pendant enable switch.

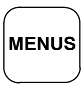
- Press the TOOL1 (or TOOL2) key. The Tool 1 (or Tool 2) screen is displayed.




```
Tool 1                          JOINT  10 %
                                      1/3
     Instruction
   1 Open hand 1
   2 Close hand 1
   3 Relax hand 1
                  EXEC
```

- Otherwise, select MANUAL FCTNS from the screen menu to display the manual operation screen.

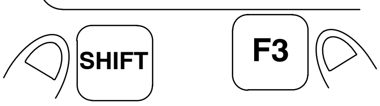
```
2 TEST CYCLE
3 MANUAL FCTNS
4 ALARM
```




```
MANUAL Macros                   JOINT  10 %
                                      1/3
     Instruction
   1 Open hand 1
   2 Close hand 1
   3 Relax hand 1
 [ TYPE ]         EXEC
```

2 Move the cursor to the desired macro instruction. Press and hold down the SHIFT key, then press the F3 (EXEC) key.

```
[TYPE]        EXEC
```



6.6 Automatic Operation

Peripheral I/O can be used to automatically start a program and operate a production line. (See Section 3.7, ”Setting Automatic Operation”.)

- The robot start request signals (RSR1 to RSR4 inputs) select and start a program. When a program is being executed or halted, the selected program is placed in the wait state. It is started once the currently executed program terminates.
- The program number selection signals (PNS1 to PNS8 inputs and PNSTROBE input) select a program. When a program is being halted or executed, these signals are ignored.
- The automatic operation start signal (PROD_START input) starts execution of the currently selected program from the first line. When a program is being halted or executed, this signal is ignored.
- Cycle stop signal (CSTOPI input) forcibly stops the currently executed program. Any programs enqueued by RSR are canceled.
  - If CSTOPI for ABORT on the system setting menu is set to FALSE, the program currently being executed is executed up to the end of the program, then is terminated forcibly. Programs placed in the wait state by RSR are cleared. (Standard setting)
  - If CSTOPI for ABORT on the system setting menu is set to TRUE, the program currently being executed is immediately terminated forcibly. Programs placed in the wait state by RSR are cleared.
- The external start signal (START input) starts a currently halted program.
  - If START for CONTINUE only on the system setting menu is set to FALSE, the currently selected program is started from the current line. A temporarily stopped program is also started. (Standard setting)
  - If START for CONTINUE only on the system setting menu is set to TRUE, any temporarily stopped program is started. This signal is ignored when there is no temporarily stopped program.

To start a program by peripheral I/O input, the robot must be in the remote mode. The remote mode is set when the following remote conditions are satisfied:

- ■ The teach pendant enable switch is turned off.
- ■ Place the system in remote mode. (For how to switch to remote mode, see the description of #139–3–1 in Section 3.16, ”SYSTEM CONFIG MENU.”
- ■ Peripheral device I/O *SFSPD input is on.
- ■ ENBL input of peripheral I/O is on.
- ■ System variable $RMT_MASTER is 0 (peripherals).

NOTE The value of $RMT_MASTER may be 0 (peripheral device), 1 (CRT/keyboard), 2 (host computer), or 3 (no remote device).

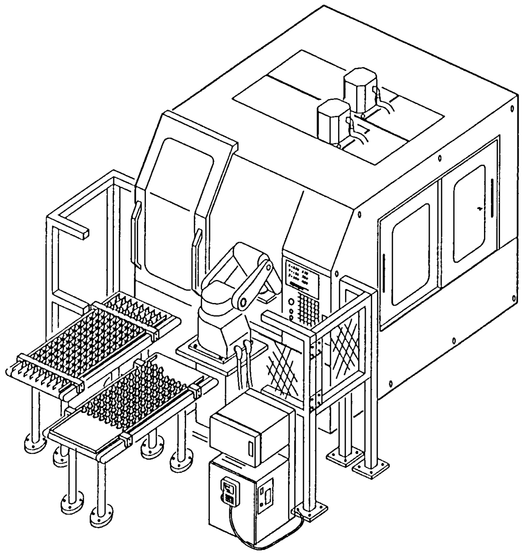
To start a program containing motion instructions, the following ready conditions must be satisfied:

- ■ ENBL input of peripheral I/O must be on.
- ■ The servo power is turned on. (No alarm is being issued.)

It is convenient to monitor the input acceptable signal (CMDENBL output) for starting a program using the peripheral I/O. The CMDENBL signal is output when the following conditions are satisfied:

- ■ Remote condition
- ■ Operation enable condition
- ■ Continuous mode (step mode is disabled)

Figure 6–13. Automatic Operation of Handling System



6.6.1 Automatic operation by robot start request (RSR)

The robot start request (RSR) function allows a remote device to select and start a program through the peripheral device I/O. This function uses eight robot start request signals (RSR1 to RSR8).

1 When a signal from RSR1 to RSR8 is input, the control unit determines whether the input RSR signal is valid. If the signal is invalid, it is ignored. When a program started by a non–RSR signal, such as a start signal from the teach pendant or the operator's panel, or when a dedicated signal START is being executed or halted, an RSR signal input is ignored.
Whether RSR is valid or invalid is set in system variables $RSR1 to $RSR8. These values can be changed on the RSR setting screen or by using a programmed RSR instruction.

2 Eight RSR registration numbers can be assigned to RSR. A base number is added to each RSR registration number to indicate an RSR program number (four–digit integer).
For example, when the RSR2 signal is input, a program having the following name is selected:
RSR + (RSR2 registration number + base number) (four digits)

NOTE The name of a program to be started must be of "RSR + RSR program number" format. (Example: RSR0121)

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

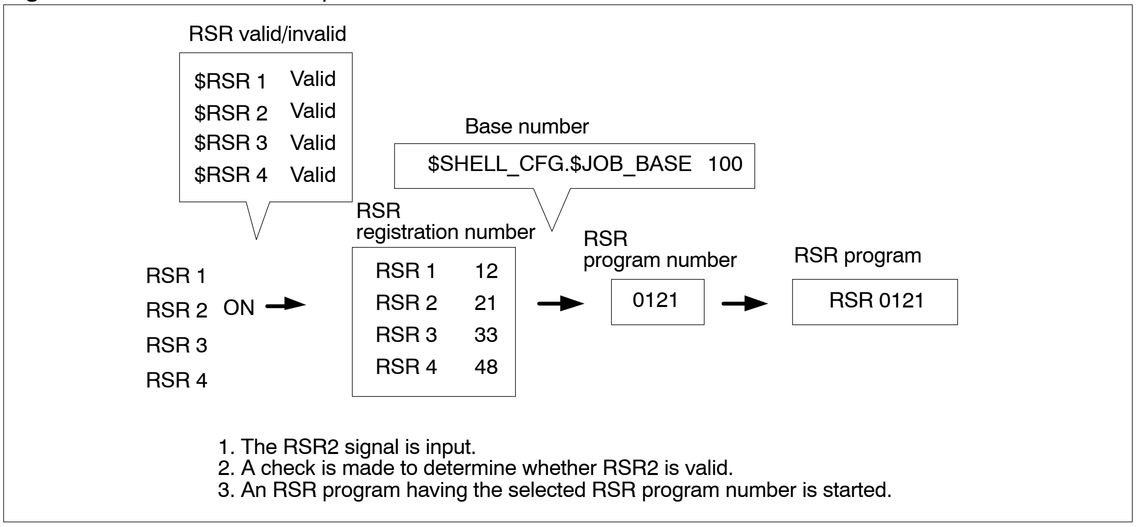
3 The RSR acknowledge output signal (ACK1 to ACK8) corresponding to one of the RSR1 to RSR8 input signals is output as a pulse signal. Even when one of the ACK1 to ACK8 signals is being output, RSR input is accepted.

4 When programs are in the terminated state, a selected program is started. When another program is being executed or halted, the request (job) is placed in a queue. It is started when the program currently being executed terminates.
Jobs (RSR programs) are executed in the order in which the programs were enqueued.

5 Programs in the queue are canceled (cleared) by the cycle stop signal (CSTOPI input) or forced program termination.

The start of a program by RSR is enabled in the remote mode. (Normally, in remote mode, the CMDENBL input is on.)

Figure 6–14. Robot Start Request



Procedure 6–14 Automatic operation by robot start request (RSR)

Condition

- RSR settings are completed. (See Section 3.7.1.)
- Remote mode is set.
- The system is ready for operation.
- Nobody must be within the work area. There must be no obstacles.

WARNING

Applying this procedure starts automatic operation which causes the robot to move. An unpredictable operation could occur. Check to ensure that nobody is in the work area, that there are no unnecessary objects in the work space, and that the safety fence is normal. Also, check that all the automatic operation conditions are set correctly. Otherwise, personal injury or damage to the facilities could occur.

Step

1 Set the enable switch on the teach pendant to OFF.

2 Place the system in remote mode. (For how to switch to remote mode, see the description of #139–3–1 in Section 3.16, "SYSTEM CONFIG MENU."

3 Send the robot start signal (RSR1 to RSR8 input) of a target RSR number to the control unit. The RSR program is placed in a queue.

4 To stop the program currently being executed, use the emergency stop button or hold button, or the immediate stop (*IMSTP input), hold (*HOLD input), or cycle stop (CSTOPI input) signal.

5 To cancel a job in the queue, use the cycle stop signal (CSTOPI input).

6 To restart a halted program, use the external start signal (START input).

6.6.2 Automatic operation with program number selection (PNS)

The program number selection (PNS) function enables selection or checking of a program, using the peripheral I/O, from the remote controller. Eight input signals, PNS1 to PNS8, specify a PNS program number.

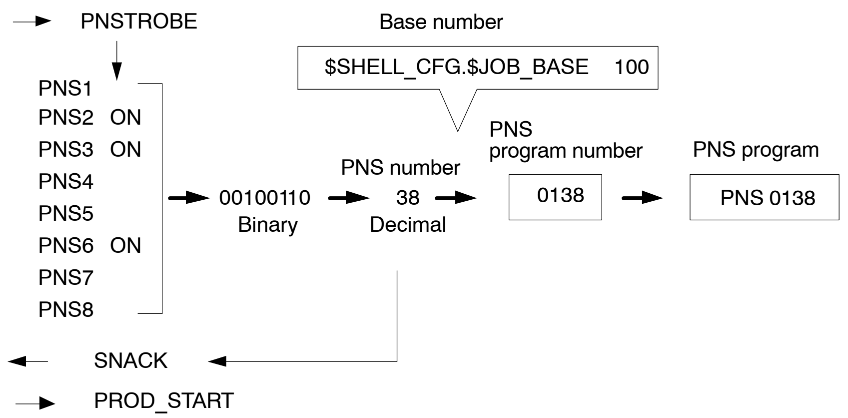
1 When the PNSTROBE pulse signal is input, the control unit reads the PNS1 to PNS8 input signals. When a program is being executed or halted, the signals are ignored. While the PNSTROBE pulse input signal is on, no program can be selected from the teach pendant.

2 The received PNS1 to PNS8 inputs are converted into a decimal number to obtain a PNS number. A program number (four digits) can be obtained by adding a base number to the PNS number, as shown below:
  (Program number) = (PNS number) + (base number)
  The selected program has the following name:
  PNS + (program number)
  When zero is input through the PNS1 to PNS8 input signals, no program is selected on the teach pendant.

NOTE The name of a started program must be of (PNS + PNS program number) format. (Example: PNS0138)

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

Figure 6–15. Program Number Selection



1. The PNSTROBE signal is input.
2. The PNS1 to PNS8 signals are read and converted into a decimal number.
3. A PNS program having the selected PNS program number is regarded as the currently selected program.
4. The PROD_START input signal starts the selected PNS program.

3 The selected program number output signals (SNO1 to SNO8) are output for PNS confirmation. The PNS acknowledge output (SNACK) signal is output as a pulse signal. This signal causes the external device to read SNO1 to SNO8 output signals.
  Even while the SNACK signal is being output, PNS and PROD_START input signals are accepted.

4 When confirming that the output values of SNO1 to SNO8 match the input values of PNS1 to PNS8, the remote control unit sends the automatic operation start input (PROD_START) signal.

5 The control unit receives the PROD_START input signal, then starts the program.

Program start by PNS is enabled in remote mode. (Normally, in remote mode, the CMNDENBL input signal is on.)

Procedure 6–15 Automatic operation by program number selection

Condition

- PNS setting must be completed (See Section 3.7.2).
- The remote condition must be satisfied.
- The operation enable condition must be satisfied.
- No one must be within the work area. There must be no obstacles.

> **WARNING**
> **Start automatic operation as follows: When the robot starts operation, an unexpected situation may occur. To prevent any problem from occurring, be sure to check that no one is in the work area, that the work are a is free from unnecessary equipment, that the safety barrier is in place, and that all the automatic operation conditions are correctly specified. Otherwise, the robot may injure a person or damage the equipment in the work area.**

Step

1. Turn off the teach pendant enable switch.
2. Place the system in remote mode. (For how to switch to remote mode, see the description of #139–3–1 in Section 3.16, "SYSTEM CONFIG MENU."
3. Send the program number selection signals (PNS1 to PNS8 inputs) indicating a target PNS number and the PNS strobe signal (PNSTROBE input) to the control unit. A PNS program is then selected.
   The control unit outputs the selected program number signals (SNO1 to SNO8 inputs) and PNS acknowledge signal (SNACK output) for confirmation.
4. Send an external start signal (PROD_START input). The selected program is then started.
5. To stop the program currently being executed, use the emergency stop button or hold button, or the immediate stop (*IMSTP input), hold (*HOLD input), or cycle stop (CSTOPI input) signal.
6. To restart a halted program, use the external start signal (START input).

6.6.3 External override selection function

The external override selection function changes feedrate override by turning on or off digital input (SDI) signals.

Two SDI signals are defined. These two signals can be combined in four different ways. So four types of feedrate override can be selected.

```
OVERRIDE SELECT                       JOINT 30%

  1  Function Enable:  ENABLE

  2  Signal1:             SDI[  1][ON ]
  3  Signal2:             SDI[ 32][OFF]

     Signal1      Signal2     Override
  4    OFF          OFF          15%
  5    OFF          ON           30%
  6    ON           OFF          65%
  7    ON           ON          100%

 [TYPE]              ENABLE     DISABLE
```

When the function changes the feedrate override, the feedrate override is not displayed, namely, the popup menu is not displayed at the upper right corner of the screen.

To enable the external override selection function, the following requirements must be satisfied:

- The external override selection function must be enabled. (OVERRIDE SELECT on the setting screen)
- The remote mode must be set.

When the external override selection function is enabled, the following occurs:

- The override key of the teach pendant is practically disabled.(The changed value is quickly returned to the setting value by the external override selection.)
- The override instruction has no effect to the override value.
- You can not change the settings of SDI signal number and Override. Before these settings can be modified, Function Enable:DISABLE must be set.
- When this function is effective at turning off the power of the controller,the override will get the value which had been set by this function when turning on it again.
- It is possible to specify the same number as two SDI signal numbers. In this case, only the combination of ON–ON or OFF–OFF has the meaning.

Moreover, note the following:

- After this function is disabled because the remote condition is not satisfied,the override keeps to remain the value specified by this function in effective until the value is changed by the teach pendant or override instruction.

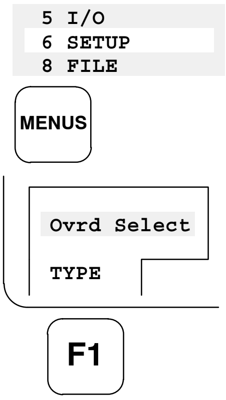
Set this function on the external override selection setting screen (6 OVERRIDE SELECT).

Procedure 6–16 Selecting an external override

Step

1 Press the MENUS screen to display the screen menu, then select 6 SETUP.

2 Select Ovrd Select from the screen change menu.



External override selection setting screen

```
OVERRIDE SELECT                      JOINT 30%

  1  Function Enable:  DISABLE

  2  Signal1:             DI[***][***]
  3  Signal2:             DI[***][***]

     Signal1      Signal2     Override
  4    OFF          OFF          10%
  5    OFF          ON           10%
  6    ON           OFF          10%
  7    ON           ON           10%

 [TYPE]              ENABLE     DISABLE
```

3 Set items.

a Enable or disable the function.

b Assign SDI signals.

```
OVERRIDE SELECT                      JOINT 30%

  2  Signal1:              DI[ 11][ON ]
  3  Signal2:              DI[***][***]

 [TYPE]              ENABLE     DISABLE
```

The states of SDI signals are indicated. When *** is displayed, the setting of the function cannot be changed.

c Feedrate override to be changed by turning on or off the signals

```
OVERRIDE SELECT                      JOINT 30%

  1  Function Enable:  ENABLE

  2  Signal1:             DI[ 11][ON ]
  3  Signal2:             DI[ 12][OFF]

     Signal1      Signal2     Override
  4    OFF          OFF          15%
  5    OFF          ON           30%
  6    ON           OFF          65%
  7    ON           ON          100%

 [TYPE]              ENABLE     DISABLE
```

6.7 Online Position Modification

Online position modification (optional function) replaces all the position data and move speeds in the move instructions within a certain range in a program at one time, according to the position modification condition, during program execution.

The following program information can be modified:

- Position data (position compensation)
- Move speed

Position data is modified by adding a position compensation value. A movement speed is modified by rewriting it. Up to ten position modification conditions can be defined.

Position compensation value

A position compensation value is the difference between the current position and the correct position. The position data coded in the move instructions within a specified range of a program is rewritten by adding a position compensation value to the data.

If the position data resulting from modification falls outside the allowable axial movement range, an alarm is generated when the program is executed.

> **CAUTION**
> If position compensation is performed during execution, it may take a while for the compensation to be reflected in actual operation.

The specifiable ranges (+/–) for the position compensation values are set in system variables $PRGADJ.$X_LIMIT to $R_LIMIT. The standard value is +/–26 mm for (X, Y, Z) and +/–0.5 degrees for (W, P, R). Any position compensation value falling outside these ranges cannot be set.

Move speed

Move speeds in the move instructions within a specified range of a program are replaced with specified speeds. The move speed for axial movement is replaced by the value specified in Joint speed, while the move speed for linear and circular movement is replaced by the value specified in Motion speed.

> **CAUTION**
> Once a speed has been rewritten, the original speed cannot be restored.

Position modification status

The position modification statuses are classified into the following three types:

– EDIT indicates that the current position modification condition is being edited. It is not reflected in the program. This state is indicated when no position modification condition is set or when a valid position modification condition is edited.

– ENABLED indicates that the current position modification condition is reflected in the program.

– DISABLED indicates that the position modification condition reflected in the program has been canceled.

The result of ENABLED is reflected immediately if the program is being executed.

When the position modification condition is modified after ENABLED, changes made to the program are determined, and state EDIT is indicated.

Online position modification is set by using 1 UTILITIES Prog Adjust on the utility screen. Online position modification conditions include the following information:

Position modification condition list screen

```
 UTILITIES Prog Adj              JOINT  10%
     Program       Lines      Status  5/10
  1  Sample 1      22-29      ENABLED
  2  Sample 1      39-49      ENABLED
  3  Sample 3      10-14      DSIABLED
  4  Sample 4    123-456      EDIT
  5  *******         0-0      *******
  6  *******         0-0      *******
  7  *******         0-0      *******
  8  *******         0-0      *******
  9  *******         0-0      *******
 10  *******         0-0      *******

 [TYPE]  DETAIL                         >

 COPY   CLR_ADJ  CLR_ALL                >
```

Position modification condition detail screen

```
 SAMPLE             LINE 0
 UTILITIES Prog Adj              JOINT  10%
                                       1/14
 Current schedule:   1 status:EDIT
  1  Program name:              Sample
  2  Starting line number:       1
  3  Ending line number:         30
  4  Offset relative to:         USER
  5  X adjustment:               5,000mm
  6  Y adjustment:               0.000mm
  7  Z adjustment:              -2.500mm
  8  W adjustment:               0.000deg
  9  P adjustment:               0.000deg
 10  R adjustment:               0.000deg
 11 Linear/Circular speed:           0mm/s
 12 Joint speed:                     0 %
 13 Motion group:                All
 14 Adjust Y for:                ROBOT

 [ TYPE ]  UNITS  SCHED  [CHOICE]        >
```

Table 6–5. Online Position Modification Settings

Item	Description
Program	Specifies the name of the target program for position modification.
Range	Specifies the range (the start and end lines) of the program lines to which position modifications are to be applied. **NOTE** The end line number must be greater than or equal to the start line number specified in item 2. When only one line is to be modified, the end line number must equal the start line number.
Offset relative to	User Modification is performed in reference to the user coordinate system. Tool Modification is performed in reference to the tool coordinate system.
Status	The position modification status indicates whether a specified position modification condition is reflected in the program. ● EDIT : The position modification condition is being edited. ● ENABLED : The position modification condition is reflected in the program. ● DISABLED : The position modification condition is not reflected in the program.
X to R adjustment	Compensation values X to R indicate the position compensation amounts. Values (X, Y, Z) are in mm or inches, while values (W, P, R) are in degrees. The values specified here are included in the position data.
Motion speed **Joint speed**	These speed items replace the move speeds. Motion speed replaces the linear and circular movement speed with a specified speed. Joint speed replaces the axial movement speed with a specified speed. **CAUTION** Once the move speed is rewritten, the original speed cannot be restored. `[TYPE]   UNIT   SCHED   ENABLE        >` `COPY   CLR_ADJ  CLR_ALL               >`
Motion group	Select an operation group to be subjected to modification.
Adjust Y for	This item is displayed only when an additional built–in traveling axis is set up as the seventh axis in group 1. The direction of the additional built–in axis is indicated in motion group. Specify the compensation target for the indicated direction. Robot: Modify only the position of the robot. Additional axis: Modify the position of the additional axis. All: Modify both the positions of the robot and the additional axis. If offset relative to is set to "Tool," only the robot can be selected.

Table 6–6. Online Position Modification Function Key Menu

Function key label	Description
UNIT	The position modification unit function changes the units of the position modification values (mm or inches).
SCHED	The schedule function is used to input the number of the position modification condition to be edited next.
ENABLE	ENABLED reflects the current position modification condition in a target program. The position data and move speeds are rewritten according to the position modification condition. This function key can be specified only when EDIT or DISABLED is indicated.
DISABLE	DISABLED cancels the current position modification condition reflected in a target program. The position data used before modification is restored. This function key can be specified only when ENABLED is indicated. The original move speed cannot be restored.
COPY	The position modification condition copy function copies a selected position modification condition into another condition number. After copying, EDIT is indicated as the modification status.
CLR_ADJ	The position modification condition erase function erases all the position modification and speed values set in a selected position modification condition. The program name and range are not erased. When erase is performed, the modified program is not restored to its original state.
CLR_ALL	This function key erases a selected position modification condition entirely including the program name and range. When erase is performed, the modified program is not restored to its original state.

Procedure 6–17 Online position modification

Condition

- There is a program to be modified.

Step

1 Press the menus key to display the screen menu.

2 Select 1 UTILITIES.

3 Press F1, [TYPE] to display the screen selection menu.

4 Select Prog Adjust. then, the position modification condition list screen appears.

```
 1 UTILITIES
 2 TEST CYCLE
```

MENUS

```
 Prog Adjust

 type
```

F1

Position modification condition list screen

```
 UTILITIES Prog Adj               JOINT  10%
     Program       Lines      Status  5/10
  1  Sample 1      22-29      ENABLED
  2  Sample 1      39-49      ENABLED
  3  Sample 3      10-14      DSIABLED
  4  Sample 4    123-456      EDIT
  5  *******         0-0      *******
  6  *******         0-0      *******
  7  *******         0-0      *******
  8  *******         0-0      *******
  9  *******         0-0      *******
 10  *******         0-0      *******

 [TYPE]  DETAIL                          >

 COPY   CLR_ADJ  CLR_ALL                 >
```

5 Position the cursor to the line number of a program to be modified. If the program to be modified is not indicated, select "***".

6 Press F2, DETAIL. Then, the position modification condition detail screen appears. When "***" is selected, EDIT is indicated as the status.

```
 UTILITIES Prog Adj
     Program       Lines
  1  Sample 1      22-29
  2  Sample 1      39-49
  3  Sample 3      10-14
  4  Test-pro    123-456
  5  *******         0-0

 [TYPE]  DETAIL
```

F2

Position modification condition detail screen

```
 UTILITIES Prog Adj               JOINT 100%

 Current schedule:   5 status:EDIT
  1  Program name:                Sample 2
  2  Starting line number:          0
  3  Ending line number:            0
  4  Offset relative to:            USER
  5  X adjustment:                  0.000mm
  6  Y adjustment:                  0.000mm
  7  Z adjustment:                  0.000mm
  8  W adjustment:                  0.000deg
  9  P adjustment:                  0.000deg
 10  R adjustment:                  0.000deg
 11 Linear/Circular speed:               mm/s
 12 Joint speed:                       10 %
 13 Motion group:                   All
 14 Adjust Y for:                   ROBOT

 [ TYPE ]  UNITS  SCHED  [CHOICE]        >
```

7 Set each items as desired.

NOTE When only one program line is to be modified, enter the same value for both the start and end lines.

```
UTILITIES Prog Adj            JOINT  10%

Carrent schedule:5       status:EDIT
 1  Program name:            Sample 2
 2  Starting line number:     1
 3  Ending line number:       30
 4  X adjustment:             5,000mm
 5  Y adjustment:             0.000mm
 6  Z adjustment:            -2.500mm
 7  W adjustment:             0.000deg
 8  P adjustment:             0.000deg
 9  R adjustment:             0.000deg
10  Motion speed:                  mm/s
11  Joint speed:                   %

[TYPE]   UNIT   SCHED   ENABLE        >

COPY   CLR_ADJ  CLR_ALL                >
```

8 After completing the modification condition settings, press F4 ENABLE to reflect the position modifications in the target program. The result of ENABLE is reflected immediately if the program is being executed.

```
SCHED  ENABLE      >
```

F4

NOTE To modify a position modification condition after making it valid, cancel the condition once, then modify it.

NOTE When move instructions include a position register or incremental instruction, modifications are not reflected.

9 To cancel a set modification condition, press F5 DISABLE. When DISABLE is used, the current position modification condition must be valid.

```
SCHED        DISABLE >
```

F5

CAUTION

Once a move speed has been changed, the original speed cannot be restored even by pressing DISABLE.

10 To set the position modification condition of another condition number, press F3 SCHED.

```
SCHED        DISABLE >
```

F3

11 Press PREV to redisplay the position modification list screen.

```
UTILITIES Prog Adj              JOINT 100%

Current schedule:   5 status:EDIT
 1  Program name:             Sample 2
 2  Starting line number:      1
 3  Ending line number:        30
 4  Offset relative to:        USER
 5  X adjustment:              5.000mm
 6  Y adjustment:              0.000mm
 7  Z adjustment:             -2.500mm
 8  W adjustment:              0.000deg
 9  P adjustment:              0.000deg
10  R adjustment:              0.000deg
11 Linear/Circular speed:      2000 mm/s
12 Joint speed:                10  %
13 Motion group:               All
14 Adjust Y for:               ROBOT

 [ TYPE ]  UNITS  SCHED  [CHOICE]       >
```

12 To copy the set modification condition to another modification condition number, position the cursor to the condition number of the copy source, and press F1, COPY on the next page. Enter the condition number of the copy destination.
Immediately after a copy operation, EDIT is indicated as the status. Modify the items as necessary.

```
COPY  CLR_ADJ  CLR_ALL
```

F1

13 To erase the set modification condition, press F2 CLR_ADJ on the next page.

```
COPY  CLR_ADJ  CLR_ALL
```

F2

7. STATUS DISPLAY

The user can check various statuses of the robot with status display. Several types of screens are used for status display.

☐ Contents of this chapter

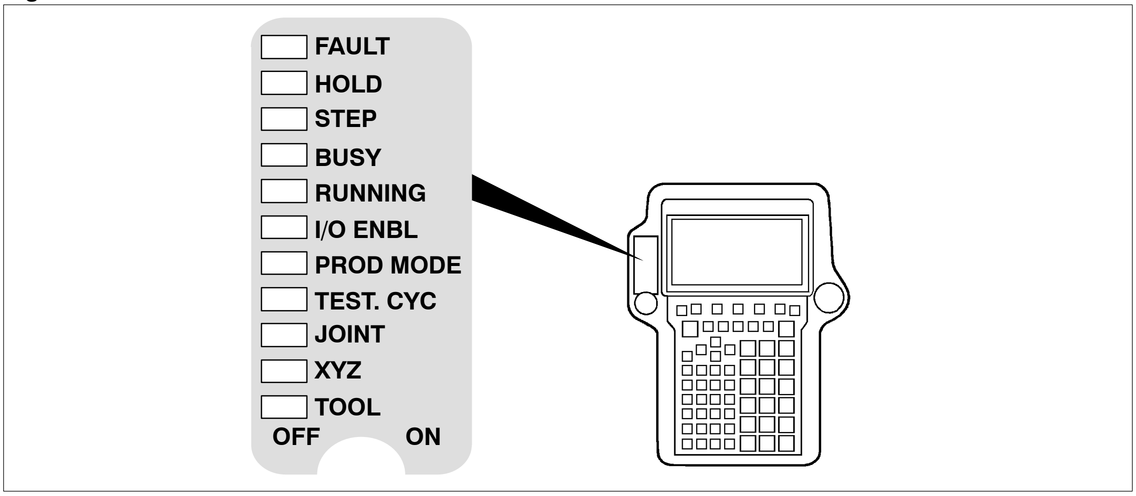
7.1 LEDs on the Teach Pendant

The LEDs on the teach pendant indicate the following statuses:

Table 7–1. LEDs on the Teach Pendant

LED	Description
FAULT	This LED indicates that an alarm has been issued. When the alarm is released, this LED goes off.
HOLD	This LED goes on while the HOLD key on the teach pendant or operator's panel is pressed or while the peripheral I/O signal, *HOLD, is applied.
STEP	This LED goes on when the single step mode is set. This LED goes off when the continuous operation mode is set.
BUSY	This LED indicates that a program or other processing is being executed.
RUNNING	This LED indicates that a program is being executed.
JOINT	This LED goes on when the manual–feed coordinate system is a joint jog coordinate system.
XYZ	This LED goes on when the manual–feed coordinate system is a Cartesian jog coordinate system (jog coordinate system or cartesion coodinate system or user coordinate system).
TOOL	This LED goes on when the manual–feed coordinate system is a tool jog coordinate system.

Figure 7–1. LEDs on the Teach Pendant



7.2 User Screen

A message instruction for the program being executed is displayed on this screen. (See Section 4.14.6.) When a message instruction is executed, the screen display automatically switches to the user screen.

Procedure 7–1 User screen display

Step

1 Press the MENUS key.

2 Select "9 USER."

NOTE When a message instruction is not executed, nothing is displayed on this screen.

NOTE Even after the program is forcibly terminated, the message remains on the screen.

7.3 Registers

A register is a variable for holding an integer or fraction. Two hundreds registers are provided. The register screen is used to display and set registers.

Procedure 7–2 Displaying register screen

Step

1 Press the MENUS key to display the screen menu.

2 Press "NEXT, " then select "DATA."

Alternatively, instead of steps 1 and 2 above, the user can press the DATA key.

3 Press F1 "TYPE."

4 Select "Registers." The register screen appears.

```
DATA Registers                         JOINT 30%
                                            1/200
   R [ 1:                     ] =  0
   R [ 2:                     ] =  0
   R [ 3:                     ] =  0
   R [ 4:                     ] =  0
   R [ 5:                     ] =  0
   R [ 6:                     ] =  0
Press ENTER
[TYPE]
```

WARNING

Registers are used in a program. Never change the value of a register before checking how the register is used in the system. Otherwise, the program can be adversely affected.

5 To enter a comment, use the following procedure:

a Move the cursor to a desired register number field, then press ENTER key.

b Select a comment input method.

c Press a desired function key, then enter a comment.

d Upon completion of input, press the ENTER key.

6 To change the value of a register, move the cursor to the register value field, then enter a desired value.

1 2 ENTER

```
DATA Registers                         JOINT 30%
                                            1/200
   R [ 1:                     ] = 12
   R [ 2:                     ] =  0
   R [ 3:                     ] =  0
   R [ 4:                     ] =  0
   R [ 5:                     ] =  0
   R [ 6:                     ] =  0
Press ENTER
[TYPE]
```

Programming example

7 Registers are used in programs when the following are specified:

– Register instruction (See Section 4.5.1)

– Indirect specification of arguments (See Section 4.2)

```
SAMPLE4                              JOINT  30 %
                                           1/8
    1:   R[1]=0
    2:   LBL[1]
    3:   CALL PRG_A
    4:   R[1]=R[1]+1
    5:   IF R[1]<=10,JMP LBL[1]
    6:   CALL PRG_B
    7:   ABORT
[End]

[ INST ]                              [EDCMD ]>
```

Program A is repeated 11 times, program B is executed, then program execution terminates.

7.4 Position Registers

A program register is a variable for holding position data. One hundreds position registers are provided. The position register screen is used to display and set registers.

Procedure 7–3 Position register setting

Step

1 Press the MENUS key to display the screen menu.

2 Press “0 NEXT,” then select “3 DATA.”

Alternatively, instead of steps 1 and 2 above, the user can press the DATA key.

3 Press F1, [TYPE] to display the screen change menu.

4 Select “Position Reg.” The position register screen appears.

```
DATA Position Reg                 JOINT 30%
                                      1/100
  PR[ 1:                   ] = *
  PR[ 2:                   ] = *
  PR[ 3:                   ] = *
  PR[ 4:                   ] = *
  PR[ 5:                   ] = *
  PR[ 6:                   ] = *
Press ENTER
[ TYPE ]         RECORD POSITION  CLEAR
```

> **WARNING**
> **Position registers are used in a program. Never change the value of a position register before checking how the register is used in the system. Otherwise, the program can be adversely affected.**

5 To enter a comment, use the following procedure:

a Move the cursor to a desired position register number field, then press ENTER key.

b Select a character input method.

c Press a desired function key, then enter a comment.

d Upon completion of input, press the ENTER key.

6 To change the value of a position register, move the cursor to the position register value field. Then, press F3 “RECORD” while holding down the SHIFT key.

```
[ TYPE ]         RECORD

SHIFT            F3
```

```
DATA Position Reg                 JOINT 30%
                                      1/100
  PR[ 1:                   ] = R
  PR[ 2:                   ] = *
  PR[ 3:                   ] = *
  PR[ 4:                   ] = *
  PR[ 5:                   ] = *
  PR[ 6:                   ] = *
Press ENTER
[ TYPE ]         RECORD POSITION  CLEAR
```

– “R” indicates that a position register already holds a taught value.

– An asterisk (*) indicates that it does not.

NOTE In a multi–motion group system, teaching a position register records the position data for all axes regardless of the current motion group.

7 To delete position data loaded into a position register, press F5 "CLEAR" while holding down the SHIFT key.

```
RECORD POSITION   CLEAR
```

SHIFT F5

```
DATA Position Reg                JOINT 30%
                                     1/100
  PR [ 1:REF_POSITION    ] = R

   PR[1] will be cleared. O.K ?
                        YES      NO
```

8 Select "YES." The position data of the desired position register is cleared.

```
     YES       NO
```

F4

```
DATA Position Reg                JOINT 30%
                                     1/100
  PR [ 1:REF POSITION    ] = *

   PR[1] has been cleared

[ TYPE ] MOVE_TO RECORD POSITION  CLEAR
```

9 To find out the current values of position data, press F4 "POSITION." The position detail data screen appears. To change a value, move the cursor to the desired field, then enter a new value.

```
RECORD POSITION CLEAR
```

F4

```
Position Detail                  JOINT 30%
PR[1]   GP:1   UF:F   UT:F   CONF: FUT O
 X: 1500.374  mm    W:  40.000  deg
 Y: -342.992  mm    P:  10.000  deg
 Z:  956.895  mm    R:  20.000  deg
DATA Position Reg
                                    1/10
  PR [ 1:REF POSITION    ] = R


                     CONFIG  DONE  [REPRE]
```

10 To change the configuration, press F3 "CONFIG." Move the cursor to a desired field, then change joint placement data using the ↓ and ↑ keys.

```
CONFIG  DONE  [REPRE]
```

F3

```
Position Detail                  JOINT 30%
PR[1]   GP:1   UF:F   UT:F   CONF: FUT O
 X: 1500.374  mm    W:  40.000  deg
 Y: -342.992  mm    P:  10.000  deg
 Z:  956.895  mm    R:  20.000  deg
DATA Position Reg
                                    1/10
  PR [ 1:REF POSITION    ] = R

Select Flip or Non-fliip by UP/DOWN key
                 POSITION  DONE  [REPRE]
```

11 To change the storage form of the position data, press F5,[REPRE] and select the storage form.

```
          1= *
          1 Cartesian
          2 Joint
 CONFIG  DONE  [REPRE]
```

F5

```
Position Detail                 JOINT  30 %
 PR[1]
 J1     34.304 deg  J4      27.089 deg
 J2     56.008 deg  J5     -10.503 deg
 J3   -121.672 deg
DATA Position Reg
```

NOTE JOINT display is valid when the robot is adjusted to the zero–degree position or when non–kinematic operation such as table operation control is executed.

12 To change the display to the additional axes (subgroup), press F2 PAGE.

```
Position   Detail
 PR[1]   UF:F UF:F          CONF:NUT 000
 E1      0.000 mm
 E2    100.204 mm
 E3     -0.894 mm
DATA Position Reg
                                     1/100
PR[  1:                 ]=R
PR[  2:                 ]=*
PR[  3:                 ]=*
PR[  4:                 ]=*
PR[  5:                 ]=*
PR[  6:                 ]=*
Enter value
           PAGE   CONFIG   DONE   [REPRE]
```

13 Upon completion of setting, press F4 "DONE."

```
CONFIG   DONE  [REPRE]
```

F4

```
DATA Position Reg              JOINT  30 %
                                    1/10
PR[  1:REF POS 1       ]=R
PR[  2:REF POS 2       ]=R
PR[  3:REF POS 3       ]=R
PR[  4:REF POS 4       ]=R
PR[  5:                ]=*
PR[  6:                ]=*

[ TYPE ] MOVE_TO RECORD POSITION  CLEAR
```

14 The position register can be used in the program as the following case:

– Position data of motion instruction(See Section 4.3.2)
– Position register instruction and offset instruction,etc. (See Section 4.5 and Section 4.3.5)

Programming example

```
SAMPLE5                        JOINT  30 %
                                     1/8
  12:  LBL[1]
  13:  OFFSET CONDITION PR[1]
  14:L PR[2] 1000cm/min CNT100 Offset
  15:  PR[3,6]=R[10]
  16:L PR[3] 1000mm/s CNT100
  17:L PR[4] 1000mm/s CNT100 Offset

[ INST ]                        [EDCMD]>
```

7.5 Palletizing Registers

The palletizing register screen displays the current values of the palletizing registers.

Procedure 7–4 Displaying a palletizing register screen

Step

1 Press the MENUS key to display the screen menu.

2 Press “0 NEXT,” then select “3 DATA.”
Alternatively, instead of steps 1 and 2 above, the user can press the DATA key.

3 Press F1 “TYPE” .

4 Select “Pallet regis.” The pallet register screen is displayed.

```
Data:Pallet Regist              JOINT  30 %
                                        1/10
PL[  1:BOX PALLET      ]=[   1,    1,    1]
PL[  2:                ]=[   1,    1,    1]
PL[  3:                ]=[   1,    1,    1]
PL[  4:                ]=[   1,    1,    1]
PL[  5:                ]=[   1,    1,    1]
PL[  6:                ]=[   1,    1,    1]
PL[  7:                ]=[   1,    1,    1]
PL[  8:                ]=[   1,    1,    1]
PL[  9:                ]=[   1,    1,    1]
PL[ 10:                ]=[   1,    1,    1]
Enter value
[ TYPE ]
```

CAUTION
Palletizing registers are used in a program. Never change the value of a palletizing register before checking how the register is used in the system. Otherwise, the program can be adversely affected.

5 To enter a comment, use the following procedure:

a Place the cursor on the comment line, then press ENTER key.

b Select the way of naming the comment.

c Press a desired function key, then enter characters.

d Upon completion of input, press the ENTER key.

6 To change the value of a palletizing register, move the cursor to the palletizing register value field, then enter a new value.

```
            JOINT 30%
                 1/32
  ] = [ 1, 1, 1 ]
```

2 ENTER

```
DATA: Pallet Regist                JOINT 30%
                                        1/10
 PL [ 1:PALLET          ] = [ 2, 1, 1 ]
 PL [ 2:                ] = [ 1, 1, 1 ]
 PL [ 3:                ] = [ 1, 1, 1 ]
 PL [ 4:                ] = [ 1, 1, 1 ]
 PL [ 5:                ] = [ 1, 1, 1 ]
```

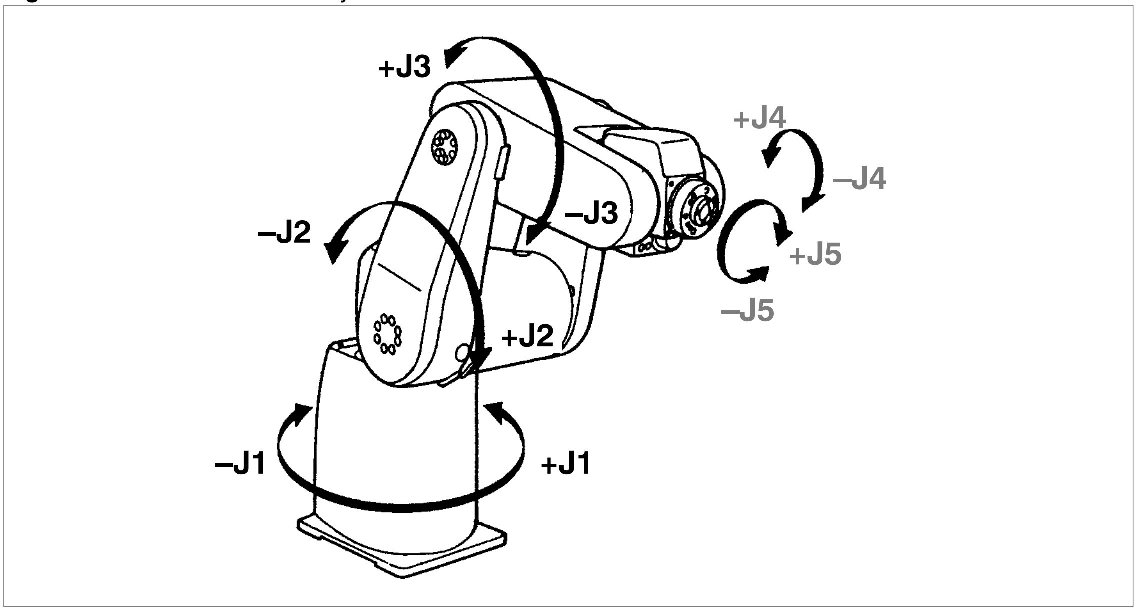
7.6 Current Position

The current position of the robot shows the location and the orientation of the robot in the work space. The current position can be represented in the cartesian frame and the joint frame.

Joint coordinates

Joint coordinates represent the current position by the angular displacement from the base side of each axis.

Figure 7–2. Joint Coordinate System



Displaying joint coordinates

```
POSITION                     JOINT  30 %
 Joint                       Tool: 1

 J1:    0.000 J2:    0.000 J3:
 J2:    0.000 J5:    0.000
 E1:    *****


[ TYPE ]   JNT    USER   WORLD
```

NOTE If the system has an additional axis, E1, E2 and E3 indicate the position data of the additional axis.

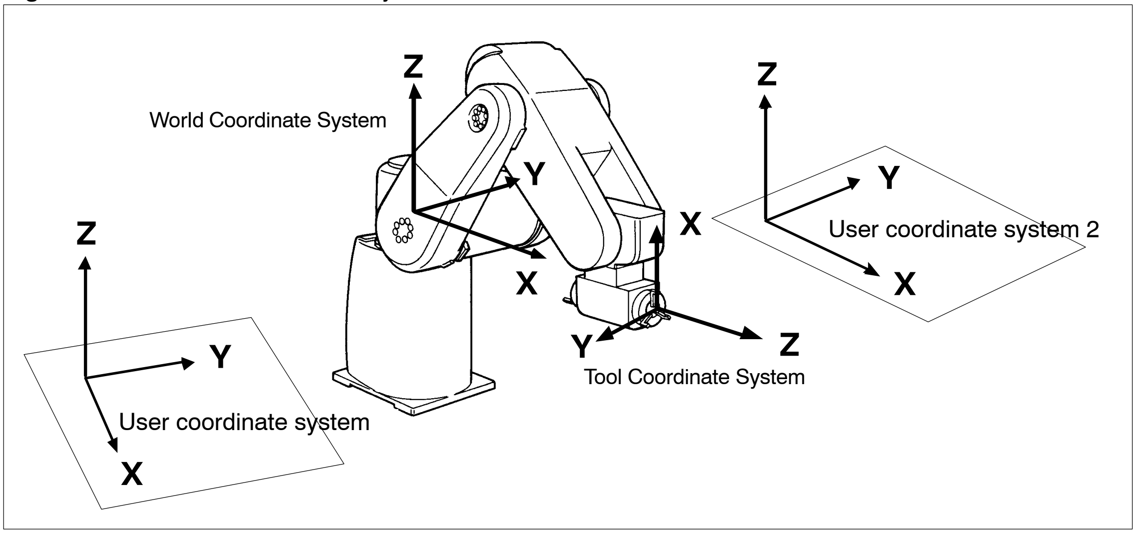
Displaying cartesian coordinates

The current position represented in cartesian coordinates is defined by the tool frame which is defined on the wrist to specify the location and orientation of the tool ,and the cartesian frame which is fixed in the work space. Cartesian coordinates is represented by the world frame or the user frame.

```
4 STATUS
5 POSITION
6 SYSTEM
```

MENUS

Figure 7–3. Cartesian coordinate system



Displaying world coordinate system

```
POSITION                          JOINT  30 %
 World                            Tool: 1
 Configuration: FUT 0
  x: 1380.000  y: -380.992  z:  956.895
  w:   40.000  p:  -12.676  r:   20.000
 E1:    *****


[ TYPE ]   JNT    USER   WORLD
```

Displaying user coordinate system

```
POSITION                          JOINT  30 %
 User               Frame: 0  Tool: 1
 Configuration: FUT 0
  x: 1500.374  y: -342.992  z:  956.895
  w:   40.000  p:   10.000  r:   20.000
 E1:    *****


[ TYPE ]   JNT    USER   WORLD
```

Procedure 7–5 Displaying current position screen

Step

1 Press the MENUS key to display the screen menu.

2 Select NEXT, then select POSITION from the next menu.

MENUS

```
9 USER
0 -- NEXT --
```

```
4 STATUS
5 POSITION
6 SYSTEM
```

3 The current position screen can be also displayed by pressing the POSN key.

– To display joint coordinates, press F2 “JNT.”

– To display user coordinates, press F3 “USER.”

– To display world coordinates, press F4 “WORLD.” (p.8–1)

7.7 System Variables

All the system variables can be seen with the system variable screen. Settings of the system is stored in the system variables.

> **WARNING**
> **The operation of the robot and control unit is controlled with system variables. Only a person who knows details of the influence of changes in system variables should set system variables. If a person without detailed knowledge attempts to set the system variables, the robot and control unit would malfunction.**

Procedure 7–6 Displaying system variable screen

Step

1 Press the MENUS key. The screen select menu is displayed.
2 Select NEXT, then select SYSTEM.
3 Press F1,[TYPE].
4 Select Variables. The system variable screen is displayed.

```
SYSTEM Variables                JOINT  10 %
                                     1/98
  1 $AP_MAXAX        536870912
  2 $AP_PLUGGED      4
  3 $AP_TOTALAX      16777216
  4 $AP_USENUM       [12] of Byte
  5 $AUTOINIT        2
  6 $BLT             19920216
  7 $CRT_DEFPROG     *uninit*
  8 $CSTOP           TRUE
  9 $DEFPULSE        4
 10 $DEVICE          'P3:'

[ TYPE ]
```

5 To change the settings of the system variables,move the cursor to the desired field and press the ENTER key after entering the value, or select the desired item from the function labels
6 When one of the system variables has plural items which belong to this variable(hierarchical structure),move the cursor to the desired system variable and press the ENTER key. Then the list of items which belongs to this variable is displayed.

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

```
SYSTEM Variables                JOINT  10 %
                                    49/98
 47 $ORIENTTOL       10.000
 48 $OVRDSLCT        OVRDSLCT_T
 49 $PARAM_GROUP     MRR_GRP_T
 50 $PASSWORD        PASSWORD_T
```

ENTER

```
SYSTEM Variables                JOINT  10 %
$PARAM_GROUP                        49/98
  1 $BELT_ENABLE     FALSE
  2 $CART_ACCEL1     192
  3 $CART_ACCEL2     0
  4 $CIRC_RATE       1
  5 $CONTAXISNUM     0
  6 $EXP_ENBL        TRUE
[ TYPE ]
```

7 To return to the upstairs layer,press the PREV key.

7.8 Program Timer

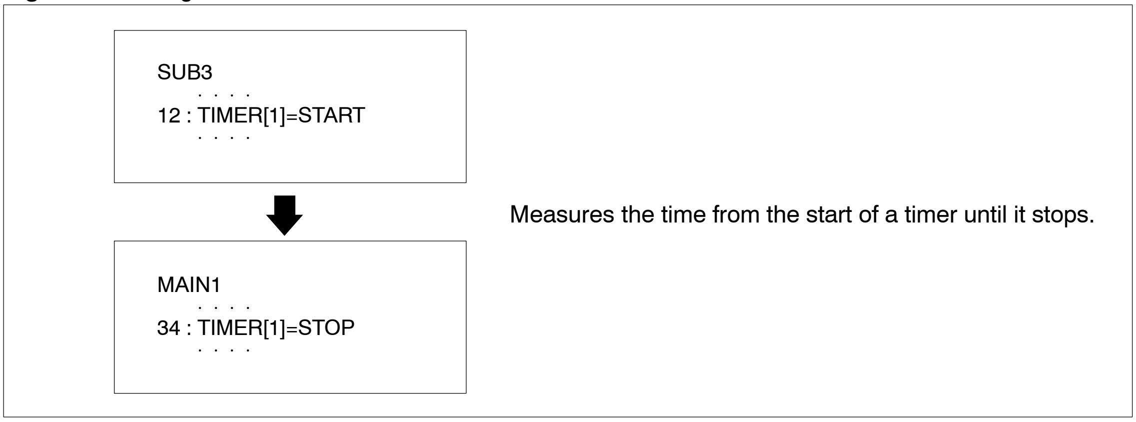
A program timer is a timer for measuring the execution time from one line to another in a program. Ten program timers can be used as standard.

A program timer can be started and stopped by using a timer instruction (see Section 4.14.3). It also stops at forced termination and upon a halt.

The program timer detail screen displays the following information:

- Program name and line number for which a timer was started most recently
- Program name and line number for which a timer was stopped most recently

Figure 7–4. Program Timer Measurement



Program timers are indicated by using 4 STATUS/Prg Timer on the program timer screen.

Procedure 7–7 Displaying program timers

Step

1 Press the menus key to display the screen menu.

2 Press 0 NEXT, and select 4, STATUS.

3 Press F1, [TYPE] to display the screen selection menu.

4 Select Prg Timer. Then, the program timer screen appears.

```
PRG TIMER LISTING                    JOINT  10%
                                           1/10
                         count     comment
 1  Timer[1]        3.20(s)[TIMER TEST    ]
 2  Timer[2]        0.00(s)[              ]
 3  Timer[3]        0.00(s)[              ]
 4  Timer[4]        0.00(s)[              ]
 5  Timer[5]        0.00(s)[              ]
 6  Timer[6]        0.00(s)[              ]
 7  Timer[7]        0.00(s)[              ]
 8  Timer[8]        0.00(s)[              ]
 9  Timer[9]        0.00(s)[              ]

[TYPE]   DETAIL
```

5 To display detail information, press F2, DETAIL. Then, the program timer detail screen appears.

```
PRG TIMER DETAIL                     JOINT  10%
                                            1/1

    Timer[1]
     Comment         :[   TIMER TEST]
     Count           :    3.20(sec)
     Start program   :[    TEST]
           line      :         1
     Stop program    :[    TEST]
           line      :         3



[TYPE]   LISTING
```

6 To enter a comment, position the cursor to the comment field, and press the enter key. Select the input method, and enter characters using function keys.

7 As the start program, a program for which the timer was started most recently is indicated.
As the stop program, a program for which the timer was stopped most recently is indicated.

7.9 System Timer

A system timer is a timer for indicating the system operation time. The times for four items are indicated. Four types of timers are provided for each operation group.

Table 7–2. System Timer Display

Item	Description
Power–on time	Time during which the power to the control unit is on
Servo–on time	Time during which the system is ready for operation (servo on) after the release of an alarm.
Operation time	Program execution time. The halt period is not included.
Standby time	Time required to execute a standby instruction

To display the system timers, use 4, STATUS Sys Timer on the system timer screen.

Procedure 7–8 Displaying the system timer screen

Step

1 Press the menus key to display the screen menu.
2 Select 4, STATUS on the next page.
3 Press F1, [TYPE].
4 Select Sys Timer. Then, the system timer screen appears.

```
SYS TIMER                          JOINT  10%
                                         1/4

     GROUP:1
       Timer type  Total(h)  Lap(m)
     On Power time:     0.2      0.0[OFF]
     Servo on time:     0.2      0.0[OFF]
     Running  time:     0.0      0.0[OFF]
     Waiting  time:     0.0      0.0[OFF]




  [TYPE]   GROUP#   ON/OFF   RESET
```

5 To switch between operation groups, press F2, GROUP#, and enter a group number.
6 To enable or disable lap time measurement, position the cursor to a desired item, and press F3, ON/OFF to switch the setting.
7 To reset the lap time, position the cursor to a desired item, and press F4, RESET.

7.10 Execution History

The function of the program execution history records the execution history of the program which has been executed or which is being executed at the end, and enables you to see the execution history after the program is finished or paused.

For example,this function enables you to recognize the execution status of the program at power failure after the cold start is done in case that power supply is turned off for any causes while the program is executed.

NOTE You can not see the execution history of the program which is been executed.

The following informations can be referred with the execution history screen.

- Executed program name and line number(The status of the latest executed program is displayed at the first line.)
- Direction of execution
  - FWD: The line was executed by the forward execution.
  - BWD: The line was executed by the backward execution.
- Status of execution
  - Not exec: The line was read but the line has not been executed.
  - Paused: (The program was paused while executing the line)
  - Done: The execution of the line has been completed.
  - Aborted: The program has finished to be executed.

The maximum number of the execution history which can be recorded is 200. The number of record lines can be changed using the maximum number setting screen, selectable from the control start menu. When the maximum number of lines that can be recorded has been reached, subsequent history data recording is performed by automatically erasing the recorded data, starting from the oldest.

Note the following when you use this function:

- When a macro is executed by using the manual function, user key, etc except the program,the execution history of it is not be recorded. When the program assigned to be a macro is executed in the program edit screen, the assigned program name in place of the macro name is recorded as the execution history.
- When the KAREL program is executed, its execution history is not recorded.
- The execution history of the program automatically started at power on is not recorded.

Procedure 7–9 Displaying program execution history

Step

1 Press the MENUS key. The screen select menu is displayed.

2 Select STATUS from the next page.

3 Press F1,[TYPE].

4 Select Exec–hist. The execution history screen is displayed.

```
Execution history              JOINT  10 %
    Program name  Line. Dirc.  Stat.
     PNS0001         3  FWD    Done
     PNS0001         6  BWD    Paused
     PNS0001         7  FWD    Paused
     PNS0001         6  FWD    Done
     PNS0001         5  FWD    Done

[ TYPE ]                          CLEAR
```

NOTE If a single program has been executed, F2, NEXT TASK and F4, ALL CLEAR are not displayed on the execution history screen.

5 Only when the displayed status of a program is "Aborted", the execution history can be cleared by pressing SHIFT + F5,CLEAR.

6 When multitasking is used, pressing SHIFT + F5 CLEAR displays the history of another task.

7 When multitasking is used, the execution history of all the tasks can be cleared by SHIFT + F5 CLEAR provided Abort is indicated for all the tasks.

7.11 Memory Use Status Display

This screen displays the use status and hardware configuration of the control unit memory. The display includes the following information:

Table 7–3. Memory Use Status Display(Pools)

ITEMS	DESCRIPTIONS
TPP	Displays the use of area to hold programs.
PERM	Displays the use of area to hold system variables and registers.
SYSTEM	Displays the use status for a part of the system software.
TEMP	Displays the use status of work area used by system software.

Table 7–4. Memory Use Status Display(Hardware)

ITEMS	DESCRIPTIONS
F–ROM	Storage capacity of the F–ROM module used in control unit
D–RAM	Storage capacity of the D–RAM (RAM) module used in control unit
C–MOS	Storage capacity of the C–MOS (RAM) module used in control unit

When the [STATUS memory] screen is selected, the following screen appears on the teach pendant. This screen indicates the information collected immediately before it appears.

A list screen displays the use status of program area, permanent area and temporary area.

Memory Status List Screen

```
STATUS Memory                  JOINT  10 %
                Total      Available
   Pools      ----------------------
   TPP  CMOS   550.0 KB     540.0 KB
   PERM CMOS   999.8 KB     364.4 KB
   TEMP DRAM  1726.9 KB    1216.2 KB
   FR FROM          KB           KB
    Description:
    TPP:   Used by .TP, .MR, .JB, .PR
    PERM:  Used by .VR, RD:, Options
    TEMP:  Used by .PC, .VR, Options

[ TYPE ] DETAIL                     HELP
```

A detailed screen displays use status of all the area mentioned above and displays the hardware information.

Memory Status Detailed screen

```
STATUS Memory                  JOINT  10 %
           Total      Free   Lrgst Free
 Pools   -----------------------------
   TPP      550.0 KB  540.0 KB  540.0 KB
   PERM     999.8 KB  364.4 KB  364.3 KB
   SYSTEM   985.8 KB    9.1 KB    9.1 KB
   TEMP    1726.9 KB 1216.2 KB 1213.2 KB
   FR             KB        KB
 Hardware -----------------------------
   FROM       6.0 MB   (*)
   DRAM       8.0 MB
   CMOS       1.0 MB

[ TYPE ] BASIC                      HELP
```

To move from a list screen to a detailed screen, press F2, DETAIL.
To move from a detailed screen to a list screen, press F2, BASIC.

Explanation of each area is displayed by pressing F5, HELP on both screens. To display the previous screen, press PREV key.

NOTE This function indicates the use status of the memory. It does not change the use status.

8. FILE INPUT/OUTPUT

This chapter describes file transfer to and from a communication device.

☐ Contents of this chapter

8.1 File Input/Output Units

With the robot control unit, the following file I/O devices can be used:

- Memory card
- Floppy disk

The standard setting specifies the use of memory cards. When floppy disks are to be used, follow the steps shown below to change the file I/O device. The use of a memory card allows files to be saved and read quickly, which can improve the work efficiency greatly.

Procedure 8–1 Changing file I/O devices

Step

1 Press MUNES to display the screen menu.

2 Select 7 FILE. The file screen appears.

```
 6 SETUP
 7 FILE
 8
```

MENUS

```
FILE                              JOINT  10%
MC: *.*                                1/17
  1 *       *    (all   files)
  2 *       KL   (all KAREL source)
  3 *       CF   (all command files)
  4 *       TX   (all text files)
  5 *       LS   (all KAREL listings)
  6 *       DT   (all KAREL data files)
  7 *       PC   (all KAREL p-code)
  8 *       TP   (all TP programs)
  9 *       MN   (all MN programs)
 10 *       VR   (all variable files)
Press DIR to generate directory
[TYPE]  [DIR]   LOAD   [BACKUP]  [UTIL] >
```

3 Press F5 UTIL, and select Set Device. Then, the following screen appears:

```
          Set Device
LOAD  [BACKUP]  UTIL
```

F5

```
                                  JOINT  10%
  1 Floppy disk
  2 Back up (FRA:)
  3
  4
FILE

  1 *       *    (all   files)
  2 *       KL   (all KAREL source)
  3 *       CF   (all command files)
  4 *       TX   (all text files)
  5 *       LS   (all KAREL listings)
  6 *       DT   (all KAREL data files)
Press DIR to generate directory
[TYPE]  [DIR]   LOAD   [BACKUP]  [UTIL] >
```

4 Select a file I/O device to be used. An abbreviation for the currently selected file I/O device appears in the upper left part of the screen.

```
FILE
FLPY:
```

Abbreviation	File I/O device
MC :	Memory card
FLPY :	Floppy disk
FRA :	Area used for automatic backup of the F–ROM in the controller

NOTE When selecting FLPY:, set the floppy disk drive on the port setting screen beforehand. (See Section 8.2.)

8.1.1 Memory card

A flash ATA memory card and SRAM memory card can be used.

CAUTION

Flash ATA memory card

1 It is recommended that files on a flash ATA memory card be backed up to media such as floppy disks to protect the flash ATA memory card contents against accidental loss.

SRAM memory card

1 The SRAM memory card requires a backup battery. When an SRAM memory card is purchased, the battery is not installed. Always install the battery in the card before attempting to use it.
2 Once the battery in the SRAM memory card reaches the end of its service life, the data on the card will be lost. Therefore, always make a backup of the card contents.

When a memory card is to be used, select the memory card according to the description of changing the file I/O devices (see Section 8.1).

8.1.2 External memory unit

Two types of floppy disk drive (FDD) are available:

- Floppy Cassette adapter (A16B–0150–B001)
- Handy File (A16B–0159–B002)

A 3.5–inch floppy disk is used. Before a new floppy disk can be used, it must be formatted by the following method:

Table 8–1. Format Specification of Floppy Disk

Type of disk	3.5–inch, 2HD or 2DD
Floppy Cassette adapter	2HD, FANUC format, 71 files maximum
Handy File	2HD, FANUC format, 71 files maximum 2HD, MS–DOS format 2DD, MS–DOS format

The disk drive is connected via the RS–232–C port. Port 1 on the disk drive is used for connection. (For communication port setting, see Section 8.2.) Table 8–2 lists the standard disk drive settings.

Table 8–2. Standard Settings for Floppy Disk Drives

Device	Speed	Stop bit	Parity bit	Data code	Time–out value
Floppy Cassette adapter	9600 baud	2 bit	None	ISO	0 sec
Handy File	9600 baud	2 bit	None	ISO	0 sec
Handy FMS–DOS	9600 baud	1 bit	None	ISO	0 sec

When a floppy disk is to be used, select the floppy disk according to the description of changing the file I/O devices (see Section 8.1).
In addition, set the floppy disk drive used for communication port setting (see Section 8.2).

CAUTION

Do not eject the floppy disk from the external memory device accessing the floppy disk, otherwise, you could damage the contents of the floppy disk.

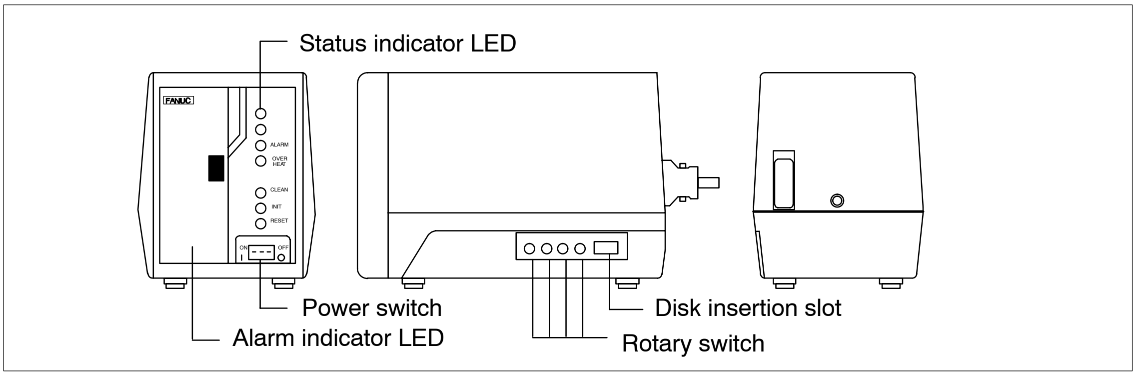
CAUTION

If a printer, floppy disk drive, vision system, or other device is connected to the control unit, the device should be turned on after the robot is turned on. Otherwise, the device can be damaged.

8.1.3 Floppy cassette adapter

The Floppy Cassette adapter is an external memory unit connected to the R–J3*i* MODEL B controller to save files stored in the internal memory of the controller to a floppy disk or read files from a floppy disk. For detailed information about the Floppy Cassette adapter (A16B–0150–B001), refer to the “FANUC FLOPPY CASSETTE ADAPTER Operator’s Manual” (B–66040E).

Figure 8–1. Floppy Cassette Adapter



Rotary switch setting

For port setting on the Floppy Cassette adapter, rotary switches 1 to 4 on the side panel are used. The standard settings for connection with the R–J3*i* MODEL B controller are “3, 1, 0, 0” from right side.

Table 8–3. Port Setting on Floppy Cassette Adapter

	Speed	Stop bit	Parity bit	Number of files	Data code
Standard setting	9600	2 bit	None	71	ISO
Switch	(1) 3		(2) 1	(3) 0 (4) 0	

Status indicator LEDs

The status indicator LEDs on the Floppy Cassette adapter indicate operation statuses.

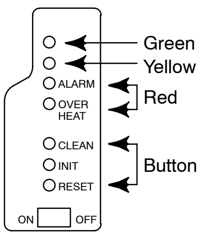


Table 8–4. Status Indicator LEDs and Switches

Green	Yellow	Status
Blinking alternately		No floppy disk is inserted, or the door is not closed.
On	On(*1)	Ready (with write protection not applied)
On	Blinking	The floppy disk is being formatted.
Blinking	On	The floppy disk is being cleaned.
On	Blinking	Data is being written.
Blinking	On(*1)	Data is being read.
Blinking simultaneously		A file is being deleted.
Button		**Function**
CLEAN		Used to clean the head
INIT		Used to format a floppy disk
RESET		Used to release an alarm

NOTE *1 Turned off when the disk is write protected.

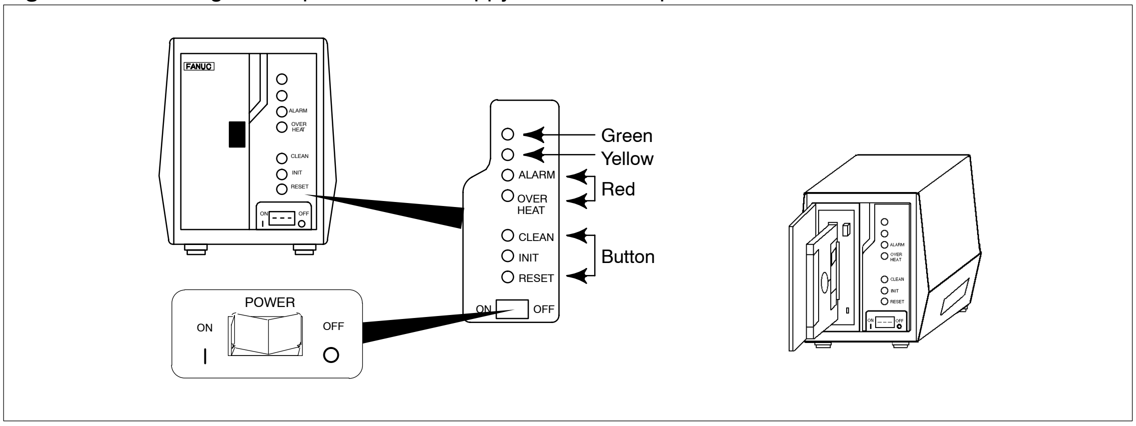
Procedure 8–2 Setting of the Floppy Cassette adapter

Step

1 Connect the Floppy Cassette adapter to the controller.

2 Turn on the power to the Floppy Cassette adapter. The green LED and yellow LED blink alternately.

Figure 8–2. Turning on the power to the Floppy Cassette adapter



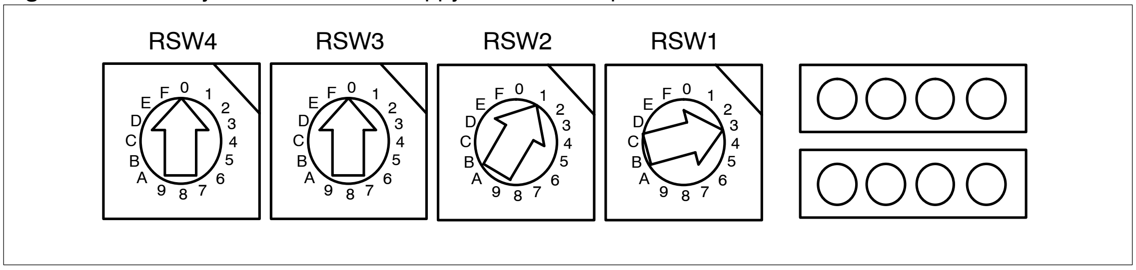
3 Insert a floppy disk, then close the door. The green LED and yellow LED light to indicate that the Floppy Cassette adapter is ready for operation. If the disk is write protected, the yellow LED does not light.

NOTE The Floppy Cassette adapter cannot be used if the door is not closed.

Setting ports

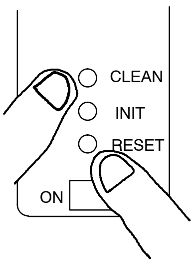
4 To set the port, open the cover which is on the left side of the floppy cassette adapter and adjust the rotary switches.

Figure 8–3. Rotary switches on the Floppy Cassette adapter



Initializing the floppy disks

5 To format the floppy disks, press and release the RESET button while holding down the INIT button. Yellow LED starts blinking to inform you of the start of initialization.

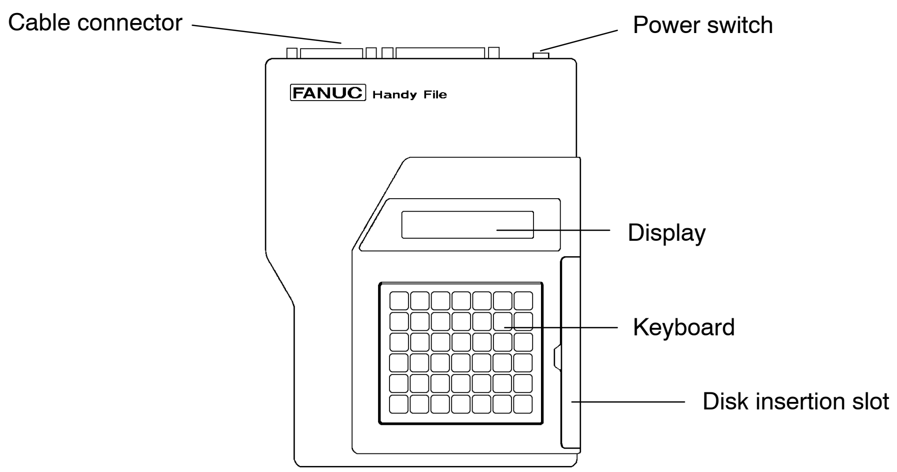


6 If an alarm is issued, press the RESET button.

8.1.4 Handy file

The Handy File is an external memory unit connected to the R–J3*i* MODEL B controller to save files stored in the internal memory of the controller to a floppy disk or read files from a floppy disk. For detailed information about the Handy File (A16B–0159–B002), refer to the “FANUC Handy File Operator’s Manual” (B–61834E).

Figure 8–4. Handy File



The settings of the Handy file are as follows. In the way of setting, there are some differences between FANUC format and MS–DOS format.

Table 8–5. Port Setting for Handy File

Setting item	FANUC format		MS–DOS format	
Protocol	Protocol B		Robot	
ISO parity bit	exist		none	
Speed	9600 baud		9600 baud	
Stop bit	2 bits		1 bits	
Parity bit	none		none	
Data code	Receive Send	ISO / EIA ISO	Receive Send	ISO / EIA ISO
Channel	RS–232–C		RS–232–C	
Subprogram	none		none	

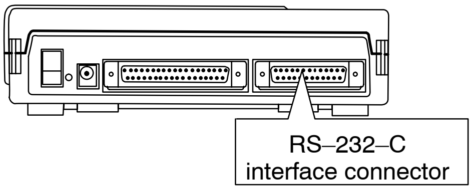
NOTE To initialize the floppy disk in MS–DOS format,use the protocol B. After initializing it, set the protocol to ROBOT again. After initialization, set the robot as the protocol. When the FANUC format is set, the disk can be initialized without changing the protocol.

NOTE When ROBOT is set as the protocol, communication with the R–J3*i* MODEL B may be broken during operation with the Handy File, even though all the settings have been made correctly. In this case, press the following keys on the Handy File:

SHIFT → WRITE SET → END

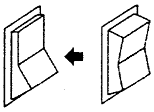
Procedure 8–3 Setting a Handy File

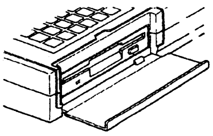
Step 1 Connect the Handy File to the controller.



2 Turn on the power to the Handy File.

3 Insert a floppy disk, then close the door. The Handy File is now ready for operation.

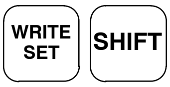




```
No file
Ready
```

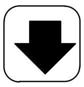
Port setting

4 The setting menu is used for port setting. Press the WRITE/SET key while holding down the SHIFT key. The setting item menu appears.




```
Select setting item
#1 : Input/Output
```

In setting, switch between menu items with the ↓ and ↑ keys. To select an item, press the ENTER key.



```
Select setting item
#2 : Protocol
```

5 Select “#2: Protocol” to display the protocol setting menu.


```
Protocol :
#1 : Protocol B
```

In setting, switch between menu items with the ↓ and ↑ keys. To select an item, press the ENTER key.

6 Select ”1 Protocol B”.

NOTE Use ”1 Protocol B” to FANUC format and Use ”2 Robot” to MS–DOS format.

7 Upon completion of protocol setting, press the END key. The setting item menu is displayed.


```
Select setting item
#3 : Baud rate
```

8 Set all the setting items as the same way as the above. When all menu items have been set, press the END key.


```
No file
Ready
```

Formatting the floppy disk

9 When the floppy disk is not formatted, a message is displayed.

```
! FD format error
Initialize FD.
>FUNC: SELECT FUNCTI.
```

10 The function menu is used to format the floppy disk. To display the function menu, press the READ/FUNC key while holding down the SHIFT key.

READ /FUNC SHIFT

```
Select function
#1 : Initialize FD
```

11 Select "#1 : Initialize FD" to format the floppy disk.

ENTER

```
Select format of FD
#1 : 2HD, 1.02MB FANUC
```

12 Select a format.

```
Set number of file
> Maximum =
```

13 Set a maximum number of files. For this example, enter "71."

NOTE Only when the FANUC format is selected, enter the maximum number of the files.

7 1 ENTER

```
Set number of file
> Maximum = 71
```

```
Initialize FD :
Press START key
```

14 Press the START key to start formatting the floppy disk.

START

```
Initialize FD :
> Executing
```

```
Initialize FD :
> Complete
```

15 Upon completion of floppy disk formatting, press the END key.

END

```
Select function
#1 : Initialize FD
```

16 To terminate the function menu, press the END key.

END

```
No file
Ready
```

NOTE When you initialize the floppy disk in MS–DOS format, select Protocol B as the communication protocol. After initializing it, select Robot as the communication protocol again.

Cleaning the head

17 The function menu is used to clean the head. Select "Cleaning" to clean the head.

```
Select function
#2 : Cleaning
```

18 Press the START key to start cleaning the head. Upon completion of head cleaning, press the END key.

8.2 Setting a Communication Port

The control unit performs data transfer to and from external devices through communication ports by performing serial communication via the RS–232C or RS–422 interface.

The following communication ports are used.

- Port 1: RS–232–C XRS–422 On the operator's box

RS–422

The use of the RS–422 interface has the following advantage:

- While the RS–232–C standard supports a cable length of only about 10 to 20 m, the RS–422 standard allows a cable to be extended to about 50 m.
- RS–422 is less susceptible to noise than RS–232–C.

Application example

- When the communication cable must be routed over a long distance, use the RS–422 interface.

NOTE The RS–422 interface uses electrical signals that are completely different from those of the RS–232–C interface. When the robot control unit and a personal computer are connected via the RS–422 interface, a commercially available RS–422–to–RS–232–C converter may be required since personal computers do not generally have a RS–422 interface.

NOTE It is impossible to use port 3 (RS–232–C) and port 4 (RS–422) simultaneously.

Communication ports are set by using [6 Setting; port setting] on the port setting screen.

Table 8–6. Standard Communication Devices for Communication Ports

Communication port	Communication device
Port 1	Handy File (FANUC format)

Table 8–7. Setting a Communication Port

ITEMS	DESCRIPTIONS
Device	This item specifies a communication device to communicate with the R–J3*i* MODEL B controller. The standard communication devices that can communicate with the R–J3*i* MODEL B are listed below: ● FANUC Handy File (A13B–0159–B002) **NOTE** The Handy File can be set to the MS–DOS or FANUC format. ● FANUC FLOPPY CASSETTE ADAPTER (A13B–0150–B001) ● PS–100/200 Disk ● FANUC PRINTER (A86L–0001–0103) ● Sensor Fanuc Eye V120 ● Host Comm Used when the R–J2 Mate is connected to the host computer to use the data transfer function. ● NO Use ● KCL/CRT ● Debug Console ● Factory Terminal ● TP Demo Derice ● Current position ● Development ● CIMPLI CITY **NOTE** When the communication device is changed, other settings such as a baud rate are changed to the corresponding standard values. Later on, the user can change each setting as desired.
Speed (Baud rate)	Baud rate is the transmission rate and it is the number of codes which can be transmitted per second. Enter the transmission rate specified for the peripheral unit being used.
Parity bit	To detect an error in data transfer, this item sets a mode of vertical parity check, which adds one extra bit to each transferred character. – Odd : The number of 1's in each transferred character must be an odd number. – Even : The number of 1's in each transferred character must be an even number. – None : No parity check is made. Enter the parity check mode specified for the peripheral unit being used.
Stop bit	This item specifies the number of stop bits to be added at the end of the transferred characters, for data transfer synchronization. – 1 bit : One stop bit is added. – 1.5 bits : One and a half stop bits are added. – 2 bits : Two stop bits are added. Enter the number of stop bits specified for the peripheral unit being used.
Time–out value (sec)	This item sets a maximum time during which control over transfer with a communication device must be exercised. If no data transfer occurs for a specified period of time, the communication line is disconnected.

Table 8–8. Standard Settings for Communication Devices

Device	Speed	Parity bit	Stop bit	Time–out value
Handy File	9600	None	2 bits	None
Handy F MS–DOS	9600	None	1 bit	None
FANUC Floppy	9600	None	2 bits	None
Printer	4800	None	1 bit	None
Sensor	4800	Odd parity	1 bit	None
Host Comm	4800	Odd parity	1 bit	None
Factory Terminal	9600	None	1 bit	None
KCL/CRT	9600	None	1 bit	None
TP Demo Device	9600	None	1 bit	None

Procedure 8–4 Setting a communication port

Step

1 Press the MENUS key to display the screen menu.

2 Select “6, SETUP.”

3 Press F1, [TYPE] to display the screen change menu,

4 Select “Port Init.” The port selection screen appears.

```
 5 I/O
 6 SETUP
 7 FILE
```

MENUS

```
 Port Init

 TYPE
```

F1

```
 SETUP Port Init                 JOINT  30 %
                                         1/3
      Connector   Port    Comment
   1 RS-232-C      P3: [      Handy File]


 [ TYPE ]         DETAIL
```

5 Move the cursor to a desired connecter port field, then press F3 “DETAIL.” The port setting screen appears.

```
[TYPE]        DETAIL
```

F3

```
 SETUP Port Init                 JOINT  30 %
   1 Device              [      Handy File]
   2 Speed(Baud rate)                [ 9600]
   3 Parity bit                      [ None]
   4 Stop bit”                     [  2bits]
   5 Time out value(sec)              [    0]
   6 Interface                    [RS-232-C]

 [ TYPE ]          LIST   [CHOICE]
```

6 To set a communication device, move the cursor to the “Device” field, then press F4, [CHOICE]. Select a desired communication device from the menu.

```
[CHOICE]
```

F4

```
                                JOINT  30 %
  1 Handy File         5 Sensor
  2 FANUC Floppy       6 Host Comm
  3 PS-100/200 Disk    7 No Use
  4 Printer            8 ---next page---
 SETUP Port Init
   1 Device             [      Handy File]
```

7 Select a communication device whose settings need to be changed. When the communication device is entered, the standard values are entered in the other setting fields.

```
 1  Handy file
 2  FANUC floppy
 3  PS-100/200 floppy
 4  Printer
SETUP Port Init
```

ENTER

```
SETUP Port Init                   JOINT 30%
 PORT
  1  Device          [FANUC floppy     ]
  2  Speed (Baud late)           [ 9600 ]
  3  Parity bit                  [ None ]
  4  Stop bit                   [ 2bits ]
  5  Time out value (sec)        [    0 ]
  6  Interface                 [RS-232-C]

[ TYPE ]           LIST   [CHOICE]
```

The other setting fields can be changed field by field. When the “Device” field is changed to another communication device, the standard values for that device are entered in the other setting fields.

NOTE To indicate that a port is not used, set ”No Use” in the corresponding field of communication equipment.

8 Upon completion of setting, press F3 “LIST.” The port selection screen appears.

```
[ TYPE ]          LIST

F3
```

```
SETUP Port Init                    JOINT 30%
 Connector port         Comment        1/1
  1 PORT              [FANUC floppy       ]




[TYPE]    DETAIL
```

NOTE When setting the communications device, the error message,”The port was not initialized.”,may be displayed and the settings of the port is returned to the previous settings. In this case, confirm the following.

- Has the communication device to be set already been set for another port?
  → The same communication device cannot be set for more than one port.
- To set ”Host Comm” to the field of device, software option, data transfer, is needed.
- For setting a sensor, the sensor interface option is required.

8.3 Files

A file is a unit of data storage in the memory of the R–J3*i* MODEL B controller. The following types of file are used mainly:

- Program file (*.TP)
- Default Logic File (*.DF)
- System file (*.SV) Used to store the settings of the system.
- I/O Config Data File (*.IO) Used to store the settings of Input/Output configuration.
- Date file (*.VR) Used to store data such as register data

8.3.1 Program file

A program file (*.MN) contains a sequence of instructions for the robot. These instructions are called program instructions. Program instructions control robot operations, peripheral devices, and each application.

A program file is automatically stored in the C–MOS RAM of the controller. A directory of program files is displayed on the program selection screen ("SELECT").

NOTE The directory of program files is not displayed on the file screen. The file screen enables you to select the external memory device which includes the desired files and manipulate the files.

On the program selection screen, operations such as copy, delete, and rename can be performed. (For program operations, see Section 5.5.)

- Registering a program (See Subsection 5.3.1.)
- Deleting a program (See Section 5.5.)
- Copying a program (See Section 5.5.)
- Changing program detail information (including the renaming of a program) (See Section 5.5.)

A program file also includes the information items listed below. These information items can be checked on the program selection screen by pressing F5 [ATTR].

- Comment : The function of a program is summarized.
- Write protection : This prevents the program from being modified and deleted.
- Modification Date : Indicates the latest date when the program was modified.
- Program size : The size of the program is indicated in bytes.
- Copy source : The name of the source program from which the program was copied is indicated. When the program is an original program, this information item is blank.

8.3.2 Default logic file

The default logic file (*.DF) includes the settings of the default logic instruction assigned to each function key (F1 to F4 key) in the program edit screen.

The default logic file is divided to the following kinds:

- DEF_MOTN0.DF Stores the settings of the default motion instructions. F1 key

The following three files stores the settings of the default logic instruction assigned to each function keys which is displayed in the next page.

- DF_LOGI1.DF F2 key
- DF_LOGI2.DF F3 key
- DF_LOGI3.DF F4 key

8.3.3 System file/application file

A system file/application file (*.SV) contains a system control program for operating the application tool software, or contains data used with the system. The following types of system file are used:

- SYSVARS.SV: Used to store the settings of the system variables relative to the frames,reference points,joint operating area and brake control.
- SYSSERVO.SV : Used to store servo parameter data
- SYSMAST.SV : Used to store mastering data
- SYSMACRO.SV : Used to store the settings of the macro command.
- FRAMEVAR.SV : Used to store the settings of the reference position which is used at setting the frame,comments,etc.

8.3.4 Data file

Date file (*.VR,*.IO,*.DT) is the file which stores the data used by the system. The following kinds are in the data file:

- Data file (*.VR)
  - NUMREG.VR : Used to store the data of the register.
  - POSREG.VR : Used to store the data of the position register.
    (Only when position register software option is used.)
- I/O configuration data file (*.IO)
  - DIOCFGSV. IO: Used to store the settings of the I/O assignment.
- Robot setting data file (*.DT)
  This file is used to store those settings that are made on the robot setting screen.
  The file name varies depending on the model.

8.3.5 ASCII file

An ASCII file (*.LS) is a file of ASCII format. ASCII files cannot be read. The contents of an ASCII file can, however, be displayed and printed using a personal computer.

8.4 Saving Files

The function of saving files stores the data which exists in the RAM memory in the controller to the external storage device such as the floppy disks,etc. The following screens on the teach pendant can be used to save the files.

- Program selection screen: A specified program is saved to a floppy disk as program files.
- File screen: The specified program file,system file,etc can be saved to a floppy disk. The following files can be saved: When a batched save operation is executed, program files, system files, and application files can all be saved at the same time.
  - Program file
  - System file
  - Default logic file
  - Standard command file
- "5 SAVE" in the function menu: It is possible to preserve it on the floppy disk as program file and a system file, etc. of the program and the data, etc. displayed on the screen. The following files can be preserved:
  - Program file
  - System file
  - Data file
  - Default logic file
  - Standard command file

8.4.1 Saving with program selection screen

Program selection screen enables you to save the specified program as the program file.

Procedure 8–5 Requirements for saving program files

Condition

- ■ The file input/output device is set correctly. (See Section 8.1.)
- ■ When a program is to be saved to a floppy disk, the floppy disk drive is ready (Section 8.1), and a correct port setting is already made (Section 8.2.)

Step

1 Press the MENUS key to display the screen menu.

2 Select NEXT and then select "1 SELECT" on the next page. The program selection screen appears.

```
 1 SELECT
 2 EDIT
```

MENUS

```
Select                              JOINT 30%
             56080 bytes free           5/5
1  PROG1         PR [PROGRAM001          ]
2  PROG2         PR [PROGRAM002          ]
3  SAMPLE1       JB [SAMPLE PROGRAM1     ]
4  SAMPLE2       JB [SAMPLE PROGRAM2     ]
5  SAMPLE3       JB [SAMPLE PROGRAM3     ]




[TYPE]  CREATE  DELETE  MONITOR  [ATTR]  >

 COPY   DETAIL  LOAD    SAVE      PRINT  >
```

3 Press NEXT,>, and press F4,SAVE on the next page. The program save screen appears.

```
LOAD   SAVE   PRINT >
```

F4

```
                                          JOINT 30%
 1 Words
 2 Upper Case
 3 Lower Case                        ---Insert---
 4 Options
 Select

     ---Save Teach Pendant Program---
  Program Name   [SAMPLE3      ]

 Enter program name
  PRG     MAIN     SUB     TEST
```

4 Enter the name of a program to be saved, then press the ENTER key. The specified program is saved to the floppy disk.

```
 Select

     ---Save Teach Pend
  Program Name   [SAMPLE3
```

ENTER

NOTE Do not include a file extension in the program name.

5 When a program having a same name as you want to save exists in the floppy disk, the file can not be saved.

```
 File already exists
```

CAUTION

If the current device already has a file having the specified name, the save function cannot overwrite that file. Before a new file is saved, the current file should be deleted from the device.

6 When the floppy disk is filled, change the floppy disk and press F4,CONTINUE.

```
 No room to save file
                        CONTINUE CANCEL
```

8.4.2 Saving all the program files using the file screen

File screen enables you to save the program file or system file which is saved in the RAM memory in the floppy disk.

The following files can be saved by pressing F4, BACKUP:

- Program file (*.TP): Used to store all the programs file which has contents of programs.
- Default logic file (*.DF): Used to store the settings of default logic instructions.
- System file (*.SV): Used to store the following files:
  - System variable file (SYSVARS.SV)
  - Servo parameter file (SYSSERVO.SV)
  - Mastering data file (SYSMAST.SV)
  - Macro data file (SYSMACRO.SV)
  - Frame setup file (FRAMEVAR.SV)
- I/O configuration data file (DIOCFGSV.IO)
- Register data file (NUMREG.VR)
- Robot setting data file

To interrupt the saving,press the PREV key while saving.

NOTE At control start time, F4 is set to RESTOR instead of BACKUP. When RESTORE/BACKUP is selected from the auxiliary menu, BACKUP is displayed.

FCTN

```
2 RESTORE/BACKUP
```

Procedure 8–6 Saving files using the file screen

Condition

- The file input/output device is set correctly. (See Section 8.1.)
- When a program is to be saved to a floppy disk, the floppy disk drive is ready (Section 8.1), and a correct port setting is already made (Section 8.2.)

Step

1 Press the MENUS key to display the screen menu.

2 Select “7 FILE.” The file screen appears.

```
6 SETUP
7 FILE
8
```

MENUS

```
FILE                              JOINT  30 %
P3: *.*
  1 *          *    (all files)
  2 *          KL   (all KAREL source)
  3 *          CF   (all command files)
  4 *          TX   (all text files)
  5 *          LS   (all KAREL listings)
  6 *          DT   (all KAREL data files)
Press DIR to generate directory

[ TYPE ] [ DIR ]  LOAD  [BACKUP][UTIL ]>

 DELETE   COPY  DISPLAY                  >
```

Saving program files

3 Press F4 “BACKUP”, then select “TPE programs.”

```
 1 System files
 2 TPE programs
 3 Application

 LOAD     BACKUP    [UTIL] >
```

F4

```
FILE                                JOINT 30%
                                        1/13
  7 *          PC (all KAREL p-code)
  8 *          TP (all TP programs)
  9 *          MN (all MN programs)
 10 *          VR (all variable files)

Save FLPY:\SAMPLE1.TP ?
         EXIT    ALL     YES     NO
```

– F2, EXIT Ends saving program files

– F3,ALL Saves all the program file and default logic instruction file.

– F4 YES Saves the specified file (program,default logic instruction).

– F5,NO Does not save the specified file (program,default logic instruction). After the file has been saved, the system asks whether the next program file is to be saved.

4 Select the desired function key. In this case,program file (*.MN) is saved in the floppy disk.

```
           EXIT    ALL
```

F3

```
Saving FLPY:\SAMPLE1.TP, please wait...
```

5 When the file which has the same name as you specified already exists in the floppy disk,the following message is displayed.

```
FLPY:\SAMPLE1.TP already exists
                OVERWRITE  SKIP  CANCEL
```

– F3,OVERWRITE The specified file is overwritten and saved.

– F4,SKIP Does not save the specified file.

– F5,CANCEL Ends saving files.

Saving the system file.

6 Press F4,SAVE and select System files. The following file is displayed.

```
 1 System files
 2 TPE programs
 3 Application

 LOAD     BACKUP    [UTIL] >
```

F4

```
 FILE Backup                      JOINT  30 %
 FLPY:\*.*
  Saving the following files to FLPY:\
        DIOCFGSV.IO
        FRAMEVAR.SV
        NUMREG.VR
        SYSVARS.SV
        SYSSERVO.SV
        SYSMAST.SV
        SYSMACRO.SV
 Backup to disk?
                              YES       NO
```

7 To save all the system files,press F4,YES. System files (DIOCFGSV.IO, FRAMEVAR.SV, NUMREG.VR, SYSVARS.SV,SYSSERVO.SV,SYSMAST.SV,SYSMACRO.SV) are saved in the floppy disk.

```
          YES       NO
```

F4

```
 Backing up to disk: FLPY:\SYSVARS.SV
```

8 When the file having the same name as you want to save exists in the floppy disk, the following message is displayed.

```
FLPY:\SYSVARS.SV already exists
                OVERWRITE  SKIP  CANCEL
```

- F3,OVERWRITE The specified file is saved by overwriting.
- F4,SKIP The specified file is not saved.
- F5,CANCEL Saving files is ended.

9 When the floppy disk is filled with files,exchange the floppy disk and press F4,CONTINUE.

```
Disk is full, change to empty disk
                        CONTINUE CANCEL
```

Batched save

10 Press F4 BACKUP, then select ALL of above.

```
ALL of above

LOAD     BACKUP    [UTIL]  >
```

F4

```
FILE                           JOINT  10 %
FLPY: *.*                            1/17
  1 *      *    (all   files)
  2 *      KL   (all KAREL source)
  3 *      CF   (all command files)
  4 *      TX   (all text files)
  5 *      LS   (all KAREL listings)
  6 *      DT   (all KAREL data files)
  7 *      PC   (all KAREL p-code)
  8 *      TP   (all TP programs)
  9 *      MN   (all MN programs)
 10 *      VR   (all variable files)
Del Handy File, backup all files?
                          YES     NO
```

NOTE Since F4, BACKUP does not appear in the control start (not control start 2), batched save operation cannot be used.

11 When F4, YES is selected, all the files in the external memory unit are erased, then all the data is saved. Processing is interrupted using the backward key. An interrupt occurs once the current file has been processed.

> **CAUTION**
> Before a batched save operation, all files in the external memory unit are erased. Before executing a batched save operation, check the files in the external memory unit.

```
YES      NO
```

F4

8.4.3 Saving with a function menu

By selecting SAVE from a function menu, the data of a screen currently displayed can be saved into the floppy disk. The data of the following screens can be saved:

- Program edit screen Program file (*.TP)
- System variable screen System variable file (SYSVARS.SV)
- Positioning screen Mastering data file (SYSMAST.SV)
- Macro instruction setting screen Macro data file (SYSMACRO.SV)
- Frame setup screen Frame setup data file (FRAMEVAR.SV)
- Register screen Register data file (NUMREG.VR)
- Position register screen Position register data file (POSREG.VR)
- Pallet register screen Pallet register data file (PALREG.VR)
- I/O screen I/O configuration data screen (DIOCFGSV.IP)
- Edit screen for each default logic instruction. Each default logic instruction. (*.DF)

Procedure 8–7 Saving with a function menu

Condition

- The file input/output device is set correctly. (See Section 8.1.)
- When a program is to be saved to a floppy disk, the floppy disk drive is ready (Section 8.1), and a correct port setting is already made (Section 8.2.)

Saving program files.

Step

1 Display the program edit screen or the program selection screen.

```
 Select                              JOINT  30 %
                 49828 bytes free      1/5
  No.  Program name        Comment
   1  PROG001       PR [PROGRAM001       ]
   2  PROG002       PR [PROGRAM002       ]
   3  SAMPLE1       JB [SAMPLE PROGRAM 1]
   4  SAMPLE2       JB [SAMPLE PROGRAM 2]
   5  SAMPLE3       JB [SAMPLE PROGRAM 3]


 [ TYPE ] CREATE DELETE  MONITOR [ATTR ]>
```

2 To display a function menu,press the FCTN key.

3 Select "2 SAVE."
A selected program file is saved.

```
 1 QUICK/FULL MENUS
 2 SAVE
 3 PRINT SCREEN
```

FCTN

4 When the program having the same name as you want to save exists in the floppy disk,the file can not be saved.

```
 File already exists
```

5 When the floppy disk is filled with the files,exchange the floppy disk and press F4,CONTINUE. All the data being saved at exchanging floppy disk is saved into the exchanged floppy disk.

```
 Disk is full, change to empty disk.
                        CONTINUE CANCEL
```

Saving other files.

Step 1 Display the screen you want to save.

```
DATA Registers                     JOINT  30 %
                                        1/32
R[  1:COUNTER1          ]=12
R[  2:                  ]=0
R[  3:                  ]=0
R[  4:                  ]=0
R[  5:                  ]=0
R[  6:                  ]=0

[ TYPE ]
```

2 Display the function menu by pressing the FCTN key.

3 Select ”2 SAVE.”
The contents of the screen being displayed is saved.

```
1 QUICK/FULL MENUS
2 SAVE
3 PRINT SCREEN
```

FCTN

4 When the file having a same name exists in the floppy disk, the file is overwrite.

5 When the floppy disk is filled with the files,exchange the floppy disk and press F4,CONTINUE. All the data being saved at exchanging floppy disks is saved into the exchanged floppy disk.

```
FLPY-005 Disk is full

DATA Registers                     JOINT  30 %
```

8.4.4 File manipulation

On the file screen, files saved on a floppy disk can be listed and a file can be copied or deleted.

Procedure 8–8 File manipulation

Condition

- The file input/output device is set correctly. (See Section 8.1.)
- When a program is to be saved to a floppy disk, the floppy disk drive is ready (Section 8.1), and a correct port setting is already made (Section 8.2.)

Step

1 Press the MENUS key. The screen menu is displayed.

2 Select 7 FILE. The file screen is displayed.

```
 6 SETUP
 7 FILE
 8
```

FCTN

```
FILE                              JOINT  30 %
FLPY: *.*
  1 *          *   (all files)
  2 *          KL  (all KAREL source)
  3 *          CF  (all command files)
  4 *          TX  (all text files)
  5 *          LS  (all KAREL listings)
  6 *          DT  (all KAREL data files)
Press DIR to generate directory
[ TYPE ] [ DIR ]  LOAD  [BACKUP][UTIL ]>

 DELETE   COPY  DISPLAY                 >
```

Displaying the list of files.

3 Press F2,[DIR].

```
                                  JOINT  30 %
 1 *.*                    5 *.LS
 2 *.KL                   6 *.DT
 3 *.CF                   7 *.PC
 4 *.TX                   8 ---next page---
FILE
```

4 Select ”*.*”(all files).The list of the files being saved into the floppy disk is displayed.

```
FILE                              JOINT  30 %
FLPY: *.*
  1 PRG1      TP      768
  2 PRG2      TP      384
  3 SYSVARS   SV    25600
  4 SYSMACRO  SV      324
  5 NUMREG    VR      708
  6 DIOCFGSV  IO      476
  7 *         *   (all files)
  8 *         KL  (all KAREL source)
 DELETE   COPY  DISPLAY                 >
```

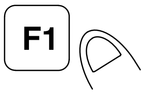
WARNING

Before a program set as a macro instruction is copied from a control unit onto another control unit, the macro setting screens of the two control units should be compared. Be sure that the lists of the two control units match. The program should be copied only when the lists match. Otherwise, an unpredictable result would occur that could injure personnel or damage equipment.

Deleting files

5 Select the file you want to delete and press F1,DELETE.

```
 2 PROGRAM2 MN       384
 3 PROGRAM3 MN       768
 4 SYSVARS  SV     25600

  DELETE   COPY  DISPLAY
```




```
Delete FLPY:\PROGRAM3.TP?
                            YES      NO
```

NOTE Deleting a program from memory of the control unit does not automatically delete the identical program from a floppy disk.

CAUTION

The operator should check that the current device has the file to be deleted. Otherwise, a wrong file can be deleted.

6 Press F4,YES. The file will be deleted.

```
FILE                          JOINT  30 %
FLPY:\*.*                          3/19
  1 PROGRAM1 TP      768
  2 PROGRAM2 TP      384
  3 <Deleted>
  4 SYSVARS  SV    25600
  5 SYSMACRO SV      324
Deleted file FLPY:\PROGRAM3.TP
 DELETE   COPY  DISPLAY                >
```

8.4.5 ASCII save

ASCII save function saves the program which is saved in the R–J3*i* MODEL B controller in binary (internal) format to the external memory device in ASCII format. This ASCII format is like the output of the printer.

Necessary devices and software version

The programs, which are saved to the floppy disk using this function, can be loaded to the personal computer and can be edited by it. Moreover, the program which is saved to the floppy disk in ASCII format can not be directly loaded into the Robot controller.(When it is converted to the internal expression by other option software on the personal computer, it can be loaded into the controller.)

Preparation for ASCII save operation

Before an ASCII save operation, check that no printer is connected to any port on the port setting screen. If a printer is connected to a port, set the port to No Use. (See Section 8.2.) A printer is connected to port 2 as standard.

File input/output device

The ASCII save function saves a file of ASCII format to a file input/output device selected according to Section 8.1. When using the Handy File, perform the operation described below. The floppy cassette adaptor cannot be used. When using other file input/output devices, proceed to Operation 8–9 for ASCII save execution.

Settings of Handy File

The software version of the FANUC Handy File needs to be 07G or more.

Set the FANUC Handy File so the floppy disk initialized in MS–DOS format can be used.(For details,refer to FANUC Handy File OPERATOR'S MANUAL.)

Table 8–9. Example for port setting of Handy File

Setting items	MS–DOS format	
Protocol	Robot	
Speed	9600 baud	
Stop bit	1 bit	
Parity bit	None	
Data code	Receive	ISO / EIA
	Send	ISO
Channel	RS–232–C	
Subprogram	None	

Select Handy F MS–DOS as the port settings on the Robot controller side and set the Handy File according to the above table.

> **CAUTION**
> Files saved in ASCII format on a FANUC format disk cannot be read into the personal computer, and so cannot be sent back to the R–J3*i* MODEL B control unit. Therefore, always use MS–DOS format.

Initializing floppy disks

When the floppy disk which has been already initialized is prepared,there is no necessity to initialize it again. When you want to use files in the floppy disk on the personal computer side, you should use the floppy disk which has been initialized according to the format of the computer. (Refer to the FANUC Handy File operator's manual for operation)

Or, use the floppy disk with the Handy File after initializing it in MS–DOS format with the personal computer etc.

Setting of the Robot Controller

Select Handy F MS–DOS as the port connected to the FANUC Handy File with the port setting screen.

Set $ASCII_SAVE to TRUE with the system variable screen.

Procedure 8–9 Execution of ASCII save function

Condition
- The file input/output device is set correctly. (See Section 8.1.)
- When a program is to be saved to a floppy disk, the floppy disk drive is ready (Section 8.1), and a correct port setting is already made (Section 8.2.)

Step

1 If a printer is set on the port setting screen, set the port to No Use. (See Section 8.2.)

```
4 ALARM
5 I/O
6 SETUP
```

MENUS

```
Port Init
TYPE
```

F1

```
SETUP Port Init                    JOINT  30 %
                                          2/3
      Connector    Port    Comment
  1 RS-232-C       P2:  [Handy File      ]
  2 PORT B         P3:  [Printer         ]
  3 JD17 RS-232-C  P4:  [No Use          ]

[ TYPE ]         DETAIL
```

```
[ TYPE ]         DETAIL
```

F3

```
LIST   [CHOICE]
```

F4

```
                                   JOINT  30 %
  1 Handy File         5 Sensor
  2 FANUC Floppy       6 Host Comm
  3 PS-100/200 Disk    7 No Use
  4 Printer            8 ---next page---
SETUP Port Init

   1 Device                 [Printer       ]
   2 Speed(Baud rate)                [4800 ]
   3 Parity bit                      [None ]
   4 Stop bit                       [2bit  ]
   5 Time out value(sec)             [    0]

[ TYPE ]         LISE   [CHOICE]
```

2 Press the MENU key to display the screen menu.

3 Select Select on the next page. The program directory screen appears.

```
Select                             JOINT  30 %
              49828 bytes free       3/5
 No.  Program name        Comment
  1   SAMPLE1          [Sample program 1]
  2   SAMPLE2          [Sample program 2]
  3   SAMPLE3          [Sample program 3]
  4   PROG001          [Program001      ]
  5   PROG002          [Program002      ]

[ TYPE ] CREATE DELETE  MONITOR [ATTR ]>
```

4 Press PRINT on the next page. The program print screen appears.

```
LOAD    SAVE    PRINT  >
```

F5

```
Select                              JOINT  30 %
 1 Words
 2 Upper Case
 3 Lower Case
 4 Options                      --Insert--
Select

 ---  Print Teach Pendant Program  ---

 Program Name [SAMPLE3 ]

Enter program name
PRG     MAIN    SUB     TEST
```

5 Enter the name of the program to be saved with the ASCII save function, then press ENTER.

```
Select

 ---  Print Teach Pendant Program  ---

 Program Name [SAMPLE3 ]
```

ENTER

6 The specified program is saved with the ASCII save function. A file is saved with extension LS. In the same way, print data can be output as a file of ASCII format by print operation based on the auxiliary menu (Section 8.6.2).

8.5 Loading Files

Loading files is to load the files being saved in the floppy disk to the C–MOS RAM memory in the controller. The files can be loaded with the following screens on the teach pendant:

- Program selection screen –The specified program file is loaded from the floppy disk as the program.
- File screen –The specified program files and system files can be loaded. The following files can be loaded.
  - Program file (*.TP or *.MN)
  - Default logic instruction (*.DF)
  - System file (*.SV)
  - Data file (*.VR,*.IO)

NOTE Selecting F4, RESTOR on the file screen in the control start (not control start 2) enables batched read. Files stored in an external memory unit are read in the following order:

1 Files having the same names as those saved when System files is selected
2 Files having the same names as those saved when Application is selected
3 *.TP, *.DF, and *.MN files in the external memory unit

*.SV and *.VR files are automatically read by selecting Convert=YES.

CAUTION

If a program having the same name exists during a program read operation, the existing program is overwritten automatically.

8.5.1 Loading using the program selection screen

In the program selection screen,the specified program file can be loaded from a floppy disk.

Procedure 8–10 Loading a program file using the program selection screen

Condition

- The file input/output device is set correctly. (See Section 8.1.)
- When a program is to be saved to a floppy disk, the floppy disk drive is ready (Section 8.1), and a correct port setting is already made (Section 8.2.)

Step

1 Press MENUS key to display the screen menu.

2 Select ”0 — NEXT —” and select ”1 SELECT” from the next page.
Program selection screen is displayed.

```
1 SELECT
2 EDIT
```

MENUS

```
 Select                          JOINT  30 %
                49828 bytes free      3/5
  No.  Program name        Comment
   1  SAMPLE1      JB [SAMPLE PROGRAM 1]
   2  SAMPLE2      JB [SAMPLE PROGRAM 2]
   3  SAMPLE3      JB [SAMPLE PROGRAM 3]
   4  PROG001      PR [PROGRAM001      ]
   5  PROG002      PR [PROGRAM002      ]

 [ TYPE ] CREATE DELETE  MONITOR [ATTR ]>

 COPY    DETAIL  LOAD    SAVE    PRINT  >
```

3 Press ”NEXT”,>, and press F3,LOAD, on the next page. Program load screen is displayed.

```
LOAD   SAVE   PRINT >
```

F3

```
 1  Words
 2  Upper Case
 3  Lower Case
 4  Options                 ---Insert---
 Select

     ---Load Teach Pendant Program---
  Program Name  [               ]

 Enter program name
 PRG     MAIN     SUB     TEST
```

4 Enter the name of a program to be loaded, then press the ENTER key.

```
Program Name PROG001

Enter program name
PRG    MAIN
```

ENTER

NOTE Do not include a file extension in the program name.

A specified program is loaded from a floppy disk.

5 When the program having the same name as you want to load exists in the memory, the following message is displayed.

```
PRG1 already exists, select function

                OVERWRITE        CANCEL
```

– OVERWRITE Loads the new file and overwrites it.

8.5.2 Loading a specified program file using the file screen

In the file screen, the specified file is loaded from the floppy disk to the memory.

The following files can be read:

- Program file (*.TP or *.MN) –Program file having contents of the program can be loaded.
- Default logic file (*.DF) –Default logic file having the settings of the default logic instruction can be loaded. The method of loading is the same as the program file.
- Data file (*.VR,*.IO) –The following data file can be loaded.
  - Register data file (NUMREG.VR)
  - Position register data file (POSREG.VR)
  - I/O config data file (DIOCFGSV.IO)
- System file (*.SV) –The following system files can be loaded. However,system files can be loaded only at the controlled start.(See Section B.1.3, "Controlled start")
  - System variable file (SYSVARS.SV)
  - Servo parameter file (SYSSERVO.SV)
  - Mastering data file (SYSMAST.SV)
  - Macro data file (SYSMACRO.SV)
  - Frame setup data file(FRAMEVAR.SV)

Procedure 8–11 Loading a program file using the file screen

Condition

- The file input/output device is set correctly. (See Section 8.1.)
- When a program is to be saved to a floppy disk, the floppy disk drive is ready (Section 8.1), and a correct port setting is already made (Section 8.2.)

Step

1 Press the MENUS key to display the screen menu.

2 Select "7 FILE" to display the file screen.

```
6 SETUP
7 FILE
8
```

MENUS

```
 FILE                                JOINT 30%
FLPY:\*.*                                1/13
   1 *          *  (all files)
   2 *          KL (all KAREL source)
   3 *          CF (all command files)
   4 *          TX (all text files)
   5 *          LS (all KAREL listings)
   6 *          DT (all KAREL data files)
   7 *          PC (all KAREL p-code)
   8 *          TP (all TP programs)
   9 *          MN (all MN programs)
  10 *          VR (all variable files)
 Press DIR to generate directory
[ TYPE ] [ DIR ]  LOAD  [BACKUP][UTIL ]>

 DELETE   COPY  DISPLAY                >
```

Loading a program file

3 Press F2 “DIR.”

```
[ TYPE ] [ DIR ]  LOAD
```

F2

```
                                   JOINT  30 %
  1 *.*                   5 *.LS
  2 *.KL                  6 *.DT
  3 *.CF                  7 *.PC
  4 *.TX                  8 ---next page---
 FILE
```

```
                                   JOINT  30 %
  1 *.MN                  5 *.IO
  2 *.TP                  6 *.DF
  3 *.VR                  7 *.ASCII Files
  4 *.SV                  8 ---next page---
 FILE
```

4 Select “*.TP” (program file). The directory of program files stored on the floppy disk is displayed.

```
  1 *.MN                  5 *
  2 *.TP                  6 *
  3 *.VR                  7 *
  4 *.SV                  8 --
```

```
 FILE                              JOINT  30 %
                                        1/17
   1 PROGRAM1 TP       768
   2 PROGRAM2 TP       384
   3 TEST1    TP      6016
   4 TEST2    TP       704
   5 *        *   (all files)
   6 *        KL  (all KAREL source)
 [ TYPE ] [ DIR ]  LOAD  [BACKUP][UTIL ]>
```

5 Move the cursor to the program file you want to load and press F3,LOAD.

```
[ TYPE ] [ DIR ]  LOAD
```

F3

```
 Loading PROGRAM1.TP, Prev to exit.
```

Selected program is loaded from the floppy disk.

```
 Loaded PROGRAM1.TP
```

6 If a program with the same name already exists in the RAM, the following indication is provided:

```
 PROGRAM1.TP already exists
                 OVERWRITE  SKIP  CANCEL
```

– OVERWRITE Loads the new file and overwrites it.

– SKIP Skips to the next file.

```
 FILE                                 JOINT 30%
                                           8/13
   8 *          TP (all TP programs)
   9 *          VR (all variable files)
  10 *          SV (all system files)

 Press DIR to generate directory
  [TYPE]   [DIR]   LOAD   [BACKUP]   [UTIL]
```

7 If you want to load all program files, select ”*.TP” and press F3,LOAD. When the PREV key is pressed, the operation is interrupted after the current a file is loaded.

```
[ TYPE ] [ DIR ]  LOAD
```

F3

Loading a data file

8 Press F2, DIR. Sub–menu is displayed.

```
[TYPE]  [DIR]   LOAD
```

F2

```
Directory Subset                      JOINT 30%
1 *.TP                  5 *.IO
2 *.MN                  6 ASCII Files
3 *.VR                  7 Loadable Files
4 *.SV                  8 ---next page---
FILE
```

9 Select “*.VR” (variable data file). The directory of variable data files stored on the floppy disk is displayed. Select a program to be loaded. The selected program is loaded from the floppy disk.

```
Directory Subset
1 *.TP
2 *.MN
3 *.VR
4 *.SV
```

ENTER

```
FILE                              JOINT  30 %
                                          1/15
   1 NUMREG   VR      868
   2 POSREG   VR     1024
   3 *        *   (all files)
   4 *        KL  (all KAREL source)
   5 *        CF  (all command files)
   6 *        TX  (all text files)
[ TYPE ] [ DIR ]  LOAD  [BACKUP][UTIL ]>
```

10 Select a program file you want to load and press F3, LOAD.

```
[ TYPE ] [ DIR ]  LOAD
```

F3

```
Loading NUMREG.VR, Prev to exit.
```

The specified program is loaded from the floppy disk. Loaded data is set as the current data.

```
Loaded NUMREG.VR
```

11 If you want to load all the file which has the same extension,Select ”*.VR”,”*.IO”,etc and press F3,LOAD.

```
FILE                              JOINT  30 %
                                          9/13
   8 *        MN  (all MN programs)
   9 *        VR  (all variable files)
  10 *        SV  (all system files)
Press DIR to generate directory
[ TYPE ] [ DIR ]  LOAD  [BACKUP][UTIL ]>
```

Loading system variable files

Condition ■ Turn on the power by controlled start (See Section B.1.3, ”Controlled start”) The following simplified system starts.

```
SYSTEM Variables          CNTRL START MENU
                                    1/98
  1 $AP_MAXAX         536870912
  2 $AP_PLUGGED       4
  3 $AP_TOTALAX       16777216
  4 $AP_USENUM        [12] of Byte
  5 $AUTOINIT         2
  6 $BLT              19920216

[ TYPE ]
```

12 Press the MENUS key, then select “2 File.” The file screen appears.

```
 1 Variables
 2 File
 3
```

MENUS

```
FILE                        CNTRL START MENU
FLPY: *.*                            1/13
  1 *         *   (all files)
  2 *         KL  (all KAREL source)
  3 *         CF  (all command files)
  4 *         TX  (all text files)
  5 *         LS  (all KAREL listings)
  6 *         DT  (all KAREL data files)
Press DIR to generate directory

[ TYPE ] [ DIR ]  LOAD  [BACKUP][UTIL ]>
```

13 Press F2 “DIR” to display the submenu.

```
[TYPE]  [DIR]  LOAD
```

F2

```
Director Subset                JOINT 30%
1 *.TP                 5
2 *.MN                 6
3 *.VR                 7
4 *.SV                 8 ---next page---
```

14 Select “*.SV” (system variable data file). The list of the system files which are saved in the floppy disk is displayed.

```
FILE                        CNTRL START MENU
FLPY: *.*                            1/17
  1 SYSVARS  SV      768
  2 SYSSERVO SV      384
  3 SYSMAST  SV     6016
  4 SYSMACRO SV      704
  5 *        *   (all files)
  6 *        KL  (all KAREL source)

[ TYPE ] [ DIR ]  LOAD  [BACKUP][UTIL ]>
```

15 Select the file you want to load and press F3,LOAD.
When you press the PREV key while the system files are loaded by selecting ”*.SV”,loading is kept on until the file being loaded at pressing the PREV key is finished to be loaded.

```
[TYPE]  [DIR]  LOAD
```

F3

16 When a system file is read, it is necessary to specify whether conversion is to be performed to maintain compatibility with the old system. Normally, select YES.

```
Convert ?
                            YES     NO
```

F4

17 Turn off the power again. Then, select "1 START (COLD)" from the function menu. The system is cold started.

```
1 START (COLD)
2
```

FCTN

Batched read

Step

1 Select a file screen in the control start (not control start 2).

2 Select F4 RESTOR.

3 A message asking the user for confirmation appears on the prompt line.

```
TEST                LINE 0
FILE                CONTROLLED START MENUS
FLPY: *.*                              2/17
  1 *       *    (all files)
  2 *       KL   (all KAREL source)
  3 *       CF   (all command files)
  4 *       TX   (all text files)
  5 *       LS   (all KAREL listings)
  6 *       DT   (all KAREL data files)
  7 *       PC   (all KAREL p-code)
  8 *       TP   (all TP programs)
  9 *       MN   (all MN programs)
 10 *       VR   (all variable files)
Restore from Handy File(OVRWRT)?
                             YES     NO
```

4 Select F4 YES. Then, the read operation starts.
Processing is interrupted using the backward key. An interrupt occurs once the current file has been processed.

8.6 Printing Files

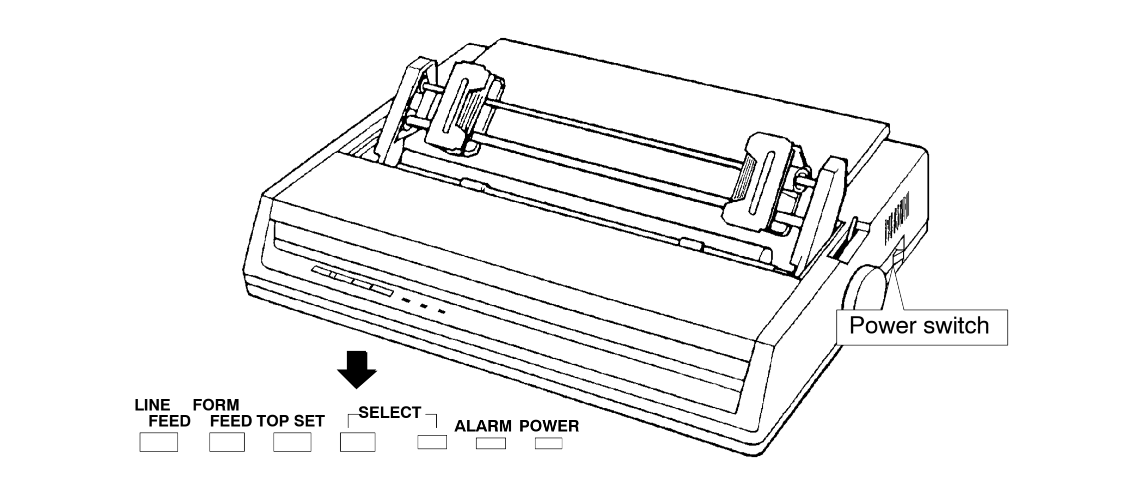
8.6.1 Printer

The printer prints out the contents of a program, data file, system variable, and so forth. A FANUC standard printer is available for connection with the Robot controller.

- FANUC Printer (A86L–0001–0103)

  The FANUC PRINTER is a serial, desktop dot–matrix printer which can print at high speed.

Figure 8–5. FANUC Printer



The Printer must be connected to a RS–232–C port. Normally, the printer is used by connecting it to port 2. (For communication setting, see Section 8.2.) Port 2 is located on the rear of the main CPU printed board.

Table 8–10. Standard Port Setting for the Printer

Device	Speed	Stop bit	Parity bit	Data code	Time–out value
Printer	4800 baud	1 bit	None	ISO	0 sec

Figure 8–6. Connection of Communication Cable to Controller

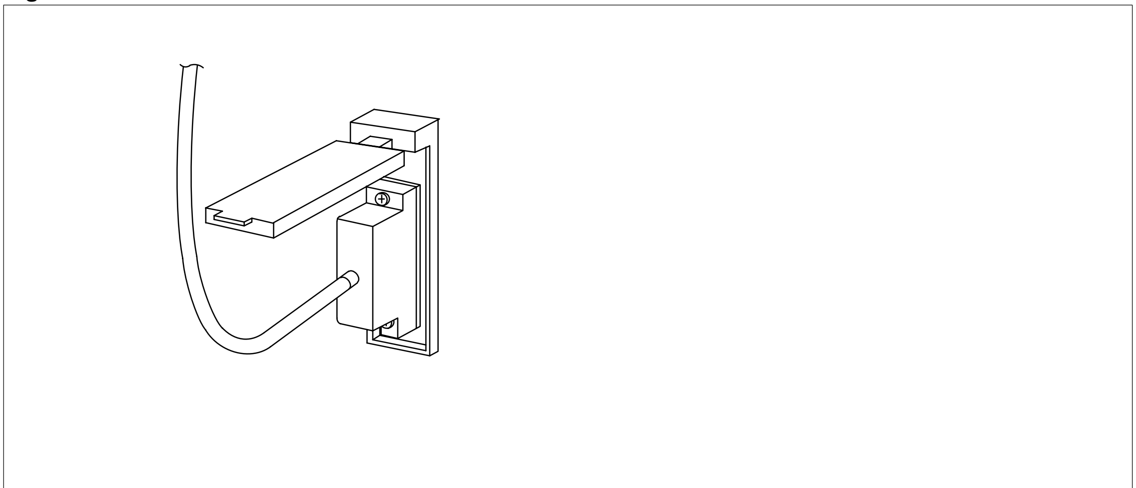


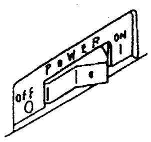
Table 8–11. Functions of LEDs and Switches

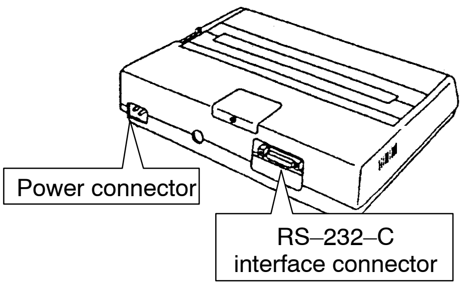
LED indication	Status
POWER (green)	Is lit when the power goes on.
ALARM (red)	Lights when a form is used up.
SELECT (green)	Is lit in the receive state (SELECT), and goes off in the local state. The SELECT switch is used to switch between the receive state and local state.
Switch	**Status**
POWER	Turns on and off the power.
SELECT	Switches between the receive state and local state.
TOP SET	Functions when the local state is set. The position of the first line is memorized. So position the form on the first line.
FORM FEED	Functions when the local state is set. This switch feeds the form to the first line of the next page.
LINE FEED	Functions when the local state is set. This switch advances the form one line.

For detailed information about the FANUC Printer (A86L–0001–0103), refer to the “FANUC Printer Operator’s Manual.”

Procedure 8–12 Operating the Printer

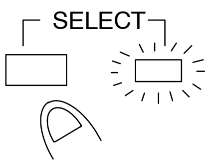
Step 1 Connect the Printer to the controller.

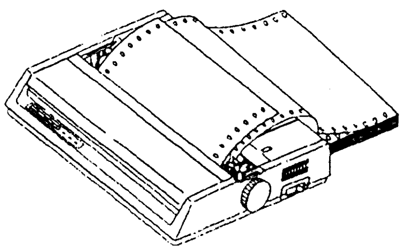




2 Turn on the power to the Printer.

3 Set an ink ribbon cartridge and form. Press the SELECT switch to set the receive state. The SELECT lamp lights.





8.6.2 Printing files

The contents of a file stored in the RAM can be printed out. The image being displayed on the teach pendant screen can also be printed out (print screen).

Printing files can be executed by the following screens.

- Program selection screen: Can print the program files.
- ”4 PRINT” on the second page of the FCTN menu: Can print the contents of the following screens:
  - Program edit screen: Program detail information and contents of program.
  - System variable screen: System variable data

NOTE If the control unit is not connected to a printer but to a PC or disk drive, printing creates a file TPSCRN.LS on the device.

Procedure 8–13 Printing files using program selection screen

Condition
- Communication port setting must be completed. (See Sections 8.2 and 8.5.1.)
- The Printer must be connected to the controller.

> **CAUTION**
> Before starting to print a file, the operator should check that the current printer is a serial printer. If not, the control unit or printer would be damaged.

Printing out a program file using the program selection screen

Step

1 Press the MENUS key to display the screen menu.

2 Select “1 SELECT” on the next page.
The program selection screen appears.

```
Select                                JOINT 30%
                56080 bytes free        3/5
1  PROG001      PR   [PROGRAM001        ]
2  PROG002      PR   [PROGRAM002        ]
3  SAMPLE1           [SAMPLE PROGRAM1   ]
4  SAMPLE2           [SAMPLE PROGRAM2   ]
5  SAMPLE3           [SAMPLE PROGRAM3   ]


[TYPE]  CREATE  DELETE  MONITOR  [ATTR] >
 COPY   DETAIL  LOAD    SAVE       PRINT >
```

3 Press F5 “PRINT” on the next page.
The program print screen appears.

```
LOAD   SAVE   PRINT >
               F5
```

```
                                      JOINT 30%
1  Words
2  Upper Case
3  Lower Case
4  Options                    ---Insert---
Select

    ---Print Teach Pendant Program---
 Program Name  [SAMPLE3      ]

Enter program name
PRG     MAIN     SUB     TEST
```

4 Enter the name of a program file to be printed out, then press the ENTER key.

```
Program Name SAMPLE3

Enter program name   ENTER
PRG    MAIN    SUB
```

5 The specified program file is printed out. To stop printing, press the PREV key.

Procedure 8–14 Printing using the miscellaneous function menu

Program printing

Condition ■ The program edit screen is displayed.

```
SAMPLE1                              JOINT  10%
                                            1/6
  1:J  P[1]   100% FINE
  2:J  P[2]   70% CNT50
  3:L  P[3]   1000mm/sec CNT30
  4:L  P[4]   500mm/sec FINE
  5:J  P[5]   100% FINE
 [End]

POINT   SINGLE   DUAL   BACKUP   TOUCHUP >
```

Step

1 Press the function key to display the miscellaneous function menu.

2 Press 0 NEXT, and select 4 PRINT.

```
9                  3 PRINT SCREEN
0 -- NEXT --       4 PRINT
                   5
```

FCTN

3 The currently displayed program is printed.
To interrupt printing, press PREV key.

System variable printing

Condition ■ The system variable screen is displayed.

```
SYSTEM Variables                     JOINT  10%
                                           1/98
  1   $AP_MAXAX                  536870912
  2   $AP_PLUGGED                4
  3   $AP_TOTALAX                16777216
  4   $AP_USENUM                 [12] of Byte
  5   $AUTOINIT                  2
  6   $BLT                       19920216

 [TYPE]
```

Step

1 Press the function key to display the miscellaneous function menu.

2 Press 0 — NEXT —, then select 4 PRINT.

```
9                  3 PRINT SCREEN
0 -- NEXT --       4 PRINT
                   5
```

FCTN

3 A list of system variables is printed.

NOTE It takes at least three hours to print all the system variables. To interrupt system variable printing, press the backward key.

4 To print only lower level system variables, for example, to print the system variables in $PARAM_GROUP, open the screen of the target level, and perform steps 1 and 2 above.

```
SYSTEM Variables

 47   $ORIENTTOL
 48   $OVRDSLCT
 49   $PARAM_GROUP
 50   $PASSWORD
```

ENTER

```
SYSTEM Variables                  JOINT  10%
$PARAM_GROUP                           49/98
  1   $BELT_ENABLE              FALSE
  2   $CART_ACCEL1              192
  3   $CART_ACCEL2              0
  4   $CIRC_RATE                1
  5   $CONTAXISNUM              0
  6   $EXP_ENBL                 TRUE

[TYPE]
```

Procedure 8–15 Printing the displayed screen (print screen)

Condition ■ The desired screen to be printed out is displayed.

Step 1 Press the FCTN key to display the auxiliary menu.

2 Press 0 —NEXT—, then select 3 PRINT SCREEN.

```
9
0 --NEXT--
```

```
2 SAVE
3 PRINT SCREEN
4 PRINT
```

FCTN

3 The displayed screen is printed out. ”¥” is printed as the part of the hlight display on the teach pendant. To stop printing, press the PREV key.

8.7 Automatic Backup

8.7.1 Overview of Automatic Backup

- Automatic Backup function performs the transaction of “all backup” in File menu automatically at the following timing.
  - The specified time (Up to 5 settings)
  - The specified DI is turned on.
  - Start up of the controller. (Interval can be specified.)
- The memory card (MC:) and the automatic backup area (FRA:) of F–ROM in the control unit can be specified as a backup copy destination. The FRA: is specified by default.
- Automatic Backup function can manage many versions of backup in one device. Even if you backup the wrong programs or settings, you can load the previous version of backup. The number of versions to keep can be set from 1 up to 99. (Default is 2.)
- A storage device to be used for automatic backup need be previously initialized for automatic backup. Automatic backup will not be performed for any external storage device that has not been initialized for automatic backup.
  Therefore, if an attempt is made to cause a backup copy to be automatically created on a memory card that has not be initialized for automatic backup, its content will not be lost.
  The FRA: need not be initialized, since it is previously initialized.
- If the control unit is turned off during automatic backup, or automatic backup is stopped immaturely, the latest backup copy is automatically restored into the system. No incomplete backup file is left in the storage unit, and the latest backup file can be read at any time.

NOTE This function automatically saves all files. If the storage device used for automatic backup becomes faulty, the data saved in it may be read. In case such an unforeseen accident takes place, it is necessary to save backups to another storage device such as a memory card as well.

8.7.2 Usable Memory Cards

The following table lists memory cards usable for automatic backup.

Type	Recommended product
Flash ATA memory card	PCMCIA Flash ATA Card manufactured by SanDisk and sold by I–O Data Device, Inc.
Compact flash memory card + PC card adapter	CompactFlash MEMORY CARD manufactured by SanDisk CompactFlash PC CARD ADAPTER manufactured by SanDisk
SRAM memory card	Available from FANUC. A87L–0001–0150#256K (with a capacity of 256 Kbytes) A87L–0001–0150#512K (with a capacity of 512 Kbytes) A87L–0001–0150#1M (with a capacity of 1 Mbyte) A87L–0001–0150#2M (with a capacity of 2 Mbytes)

NOTE 1 The SRAM card will lose its contents when the life of its built–in battery expires.
Neither the Flash ATA memory card nor the CompactFlash memory card need batteries.
It is recommended to use the Flash ATA or CompactFlash memory card for this function.

NOTE 2 The required storage capacity is “(program size + 200 Kbytes) × (number of backup copies + 1).”
If the size of a program is 500 Kbytes, 13 backup copy versions of it can be made on a 10–Mbyte memory card.

NOTE 3 If a memory card other then those recommended is used, a normal operation is not guaranteed, and a bad influence may occur on the control unit.

8.7.3 Setting of Automatic Backup

MENU→“7 FILE”→F1([TYPE])→“Auto Backup”. The following menu is displayed.

```
AUTO BACKUP JOINT 100%
                                   1/ 13
  1 Automatic Backup:             ENABLE
  2 Device:               Backup (FRA:)
      Status: Ready for auto backup
Backup Schedule------------------------
  3 Backup time 1:                 12:00
  4 Backup time 1:                 23:30
  5 Backup time 1:                 **:**
  6 Backup time 1:                 **:**
  7 Backup time 1:                 **:**
  8 Backup at DI rising:      DI[     0]
  9 Backup at Power up:          DISABLE
 10   Interval:                  7 Day
Status Output--------------------------
 11 Backup in progress:        DO[     0]
 12 Error occurs at backup:    DO[     0]
Version Management---------------------
 13 Maximum number of versions:        1
 14 Loadable version:     01/01/30 12:00

[TY PE] INIT DEV         CLEAR
```

Automatic Backup works only when it is ENABLE.

Device to save.
Default is “Backup (FRA:)”

Current status of the device is displayed.

Set the time to backup.
Up to 5 settings.
To clear setting, press F4 (CLEAR).

When the specified DI is turned on, backup is performed. (If index is 0, it is disabled.

If it is ENABLE, backup is performed at start up.
The interval can be set.

The specified DO is turned on when backup is performed, or when error occurs at backup.
(→8.7.4 Perform Automatic backup)

Settings to manage versions of backup.
(→8.7.5 Version management)
(→8.7.6 Restore the backup)

Power–on time backup

If “Backup at Power up” is enabled, a backup copy is made when the power is turned on.
If the date of the latest backup copy in the storage device is within a period range (specified in “Interval”) from the current date, no backup copy is made at power–on time.
The period range is 7 days by default. If the default value is left unchanged, a backup copy is made at power–on time once every 7 days provided that “Backup at Power up” is enabled.
The unit of interval can be selected from “Day,” “Time,” and “Minute.”
If the “Interval” is reset to 0, a back–up copy is made every time the power is turned on.

Initializing of the storage device *

To use Memory Card for Automatic Backup, the Memory Card must be initialized for Automatic Backup. It is to protect to write to the other Memory Card. The status of device is displayed in ”Status” line.

The FRA: need not be initialized, since it is previously initialized.

Ready for auto backup	Device is initialized for automatic backup
Device is not ready!	Device is not ready or device is not initialized for automatic backup

Device is initialized by the following operation.

(1) If the device is not formatted, please format the device in file menu.

(2) Press F2 (INIT_DEV)

(3) Message “Initialize the device for auto backup?” is displayed. Press F4 (YES).

(4) Message “Enter number of versions to keep:” is displayed. Please enter the number (1 to 99) of versions to keep. Pressing only the enter key sets the number of backup copy versions to 2.

NOTE INIT_DEV deletes all files in the device, and create the special files and directories.

NOTE INIT_DEV does not format the device.
Please format the device in file menu (F5 (UTIL)→“Format”)

8.7.4 Perform Automatic backup

When the specified condition is satisfied, automatic backup is performed.

```
AUTO BACKUP                       JOINT 100 %
                                        1/9

     Automatic backup in progress

          Please wait

          Device:  FRA:

   To stop backup, press [PREV] key


Saving MC:SYSVARS. SV (3/48)
```

Saving File

- While automatic backup is performed, the menu is displayed. When automatic backup is completed, the previous menu is displayed.
- If you press PREV key, backup is cancelled and the previous menu is displayed. Any key except PREV is not accepted while automatic backup is performed.
- Even if you are using Teach Pendant, when automatic backup is performed, this menu is displayed and any key except PREV is not accepted. Please wait for Automatic backup is completed.
- If an attempt is made to perform automatic backup during program execution, it is performed while the program is running. It is also possible to start a program from the outside during backup.
- If the backup–in–progress signal is set, the specified signal becomes on while this menu is displayed.

```
AUTO BACKUP                       JOINT 100 %
                                         1/9

   Error occurred at Automatic Backup!

          Check device (FRA:)


   To stop backup, press [PREV] key.



                               RETRY
```

Reason why backup is impossible

- This menu appears if backup is impossible, for example, because no memory card has been inserted.
- In this case, the robot will not enter an alarm state. If a program is already running, it continues running. Also in this case, it is possible to start a program from the outside.
- By pressing F5(RETRY), backup is performed again.
- Pressing the PREV key resumes the previous menu.
- If a backup error signal is set, the specified signal becomes on while this menu is displayed.

8.7.5 Version management

Automatic Backup function can keep many backups in one device. The number of versions to keep is set at initializing the device. And you can change the number of versions to keep by the item “Maximum number of versions” anytime. The number of versions exceeds the specified number, the oldest version is deleted automatically.

If the device is FRA: If the size of a free storage area in F–ROM in the control unit becomes smaller than 1 Mbyte, the oldest backup version is deleted automatically. In this case, the number of back versions actually held becomes smaller than “Maximum number of versions.” If the size of a free storage area in F–ROM is too small to hold an additional backup version, an error is detected during automatic backup execution.

If it is impossible to hold a specified number of backup versions on a memory card because of an insufficient storage capacity, an error is detected during automatic backup execution. Specify an appropriate number of backup versions by assuming the storage capacity required to hold one backup version is “program size + 200 Kbytes.”

If an error is detected because of an insufficient storage area during automatic backup, decrease the value specified in “Maximum number of versions.” This will causes an old backup version to be deleted, thus increasing a free area in the storage device.

Once a backup version is deleted by decreasing the value specified in “Maximum number of versions,” it cannot be restored by increasing the value.

Backup is stored in individual sub directories.

When automatic backup is performed, backup files are saved to the root directory, then these files are copied to the appropriate directory.

File menu can access the files only in root directory, so the latest version of backup can be loaded by file menu.

You can also load the older versions. (→ 8.7.6 Restore the backup)

When “all backup” is performed in file menu to the device that is initialized for Automatic Backup, the files are copied to the appropriate sub directory as same as automatic backup.

If the control unit is turned off during backup, or backup is stopped prematurely, all backup files created during the current backup session are deleted, and the last backup version selected is restored to the root directory.

8.7.6 Restore the backup

Backup files saved by Automatic Backup can be loaded by file menu. Pressing all of above on the file menu of the controlled start menu enables all files to be read simultaneously.

Usually the latest version of backup is in root directory and the version can be loaded by file menu.

You can load the previous version by the following operation.

(1) Press F4 (CHOICE) on the “Loadable version” item. The menu that contains the backup time of all versions in the device is displayed.

```
1 99/06/16 12:00      5 99/06/14 12:00
2 99/06/15 23:30      6 99/06/13 23:30
3 99/06/15 12:00      7 99/06/13 12:00
4 99/06/14 23:30      8 -- Next Page --
AUTO BACKUP                  JOINT 100 %
Version Management--------------------
 13 Maximum number of versions:        1
 14  Loadable version:    99/06/16 12:00


[ TYPE ]INIT_DEV         [CHOICE]
```

(2) Please select the version to load, then the item “ Loadable version ” shows the time of the selected version. At this time, the files of the selected version of backup are copied to root directory.

(3) You can load the files of the selected version in file menu.
When controlled start is performed, pressing all of above on the file menu of the controlled start menu enables all backup files to be read simultaneously.

9. UTILITY

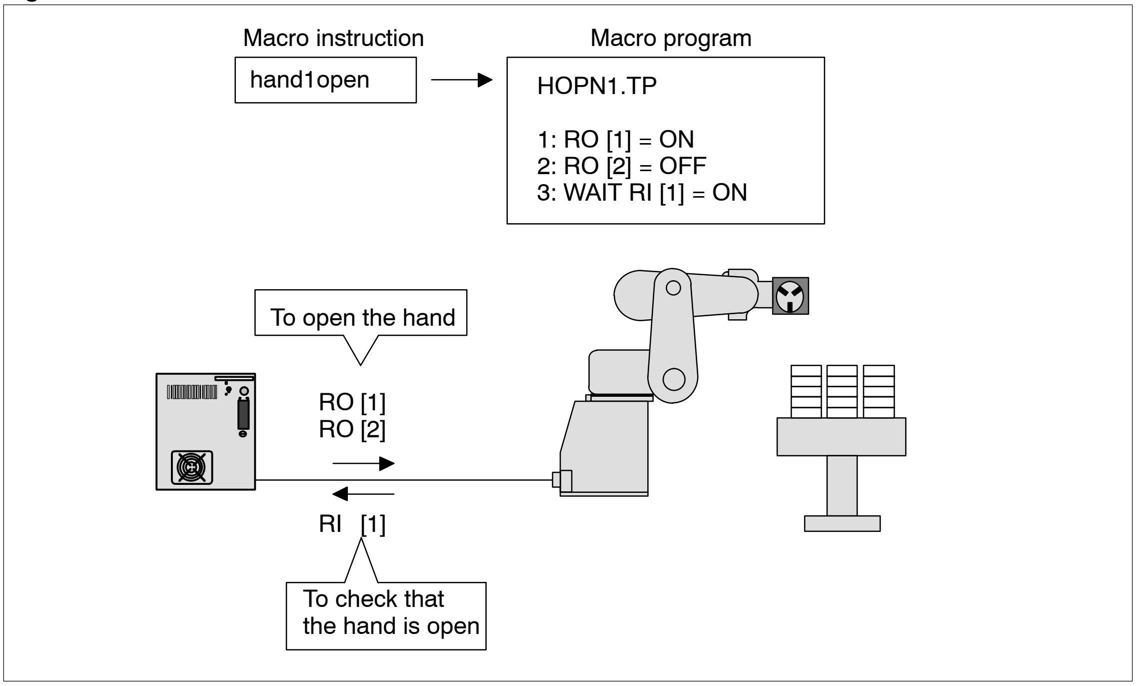
This chapter explains a macro instruction, which is a special function of the Robot controller.

☐ Contents of this chapter

9.1 Macro Instruction

A macro instruction is a function for registering a program consisting of a sequence of instructions as one instruction, and calling such a set of instructions for execution as required.

Figure 9–1. Macro Instructions



A macro instruction has the following capabilities:

- A macro instruction, when taught in a program, can be started as a program instruction.
- A macro instruction can be started using the manual operation screen on the teach pendant.
- A macro instruction can be started using a user key on the teach pendant.
- A macro instruction can be started using the user button on the operator's panel. (The operation box can not be used because it does not have a user key.)
- You can start the macro command using SDI, RDI or UI.

Existing programs can be registered as macro instructions. Up to 20 macro instructions can be registered. A macro instruction can be used according to the following procedure:

1. Create a program to be executed as a macro instruction.
2. Register the created macro program as a macro instruction and determine from which device the macro instruction is to be called.
3. Execute the macro instruction.

The macro instruction setting screen [6 SETUP. Macro] is used for setting a macro instruction.

9.1.1 Setting macro instructions

The setting of a macro instruction involves the following items:

- Macro program
- Name of a macro instruction
- Assignment of a device used to start the macro instruction

Macro program

A macro program is a program started by a macro instruction. A macro program can be taught and played back (when played back as a program) in the same way as an ordinary program, except for the following restrictions:

- The subtype of a program, when registered as a macro program, is changed to MR (macro). When the registration of the macro program is canceled, the subtype returns to the original one. (For information about the subtype s, see Section 4.1.3.)
- A macro program registered as a macro instruction cannot be deleted.
- A program not including a motion (group) can be started even when the motion enabled state is not set (even when an alarm is issued). (For the group mask, see Section 4.1.4.) For group mask setting, the program information screen is used. (See Section 5.3.1.)
- The macro command not having the motion instruction should be made as the program which does not contain the motion group.

Name of a macro instruction

The name of a macro instruction is used to call the macro program from within a program. A macro instruction name must consist of an alphanumeric character string not longer than 16 characters. The macro command can be started while robot is moving.

Assignment of a device

A macro instruction must be assigned to a key, screen item, etc. so it can be called. The item to which a macro instruction is assigned is called a device. The following devices are available:

- Items on the manual operation screen on the teach pendant (MF)
- User keys on the teach pendant (UK and SU)
- User buttons and other buttons on the operator's panel (not provided on the operator's box)
- SDI, RDI, UI

NOTE If a macro instruction is allocated to a key switch on the teach pendant, the function previously allocated to the key becomes unavailable.

> **CAUTION**
> The operator should check that no macro instructions are allocated to user keys of the teach pendant. If some instructions are allocated, a trouble would occur during execution.

Macro instructions can be assigned to the following devices:

- MF[1] to MF[99] : Items on the manual operation screen
- UK[1] to UK[7] : User keys 1 to 7 on the teach pendant
- SU[1] to SU[7] : User keys 1 to 7 + SHIFT key on the teach pendant
- SP[4] to SP[5] : User button 1 to 2 on the teach pendant
- DI[1] to DI[99] : SDI 1 to 99
- RI[1] to RI[24] : RDI 1 to 24
- UI[7] HOME signal

NOTE MF numbers from 1 to 99 can be used, but no more than 20 macro instructions can be assigned to MF items.

NOTE The total number of the assign to the DI and RI is up to 5.

NOTE The allocation of macros to UI signals other than the HOME signal can be enabled with system variable $MACRUOPENBL.

NOTE The number which can be actually used is only logical number allocated to the input signal line.

The macro instruction setting screen [6 SETUP. Macro] is used for setting a macro instruction.

> **WARNING**
> **Before a program set as a macro instruction is copied from a control unit onto another control unit, the macro setting screens of the two control units should be compared. It should be ensured that the lists of the two control units match. The program should be copied only when the lists match. Otherwise, an unpredictable result would be produced.**

Procedure 9–1 Setting macro instructions

Condition ■ A macro program is created.

```
HOPN1                                    JOINT 30%
                                              2/6
  1: RO[1]=ON
  2: RO[2]=OFF
  3: WAIT RI[1]=ON
 [END]

 POINT                                  TOUCHUP >
```

Condition ■ Macro program detail information is set.

NOTE For greater convenience, a group mask can be set for a program not including motion instructions.

NOTE If the program to be modified contains a motion instruction, the group mask cannot be set.

```
 Program detail                         JOINT 30%
                                              1/7
  1  Program name:                   [HOPN1   ]
  2  Sub Type:                   [None        ]
  3  Comment:                    [Open HAND1 ]
  4  Group Mask:            [ *   *   *   *   *  ]
  5  Write protect:                  [OFF     ]
  6  Ignore pause:                   [OFF     ]


 END      PREV     NEXT
```

Changing the motion group (setting a group mask)

Step
1. The program information screen is used to change the group mask.
2. Press the MENUS key to display the screen menu.
3. Select “1 SELECT” on the next page. The program selection screen appears.
4. Press F2 “DETAIL” on the next page. The program information screen appears.
5. Move the cursor to group 1 of “Group Mask.” Press F5 “*” to set (*,*,*,*,*).

1 *

F5

```
 Program detail                         JOINT 30%
                                              4/7
  4  Group Mask:            [ *   *   *   *   *  ]



 END      PREV     NEXT       1         *
```

NOTE If a motion instruction is already taught in a program to be modified, no group mask can be set.

Setting a macro instruction

Step

1 Press the MENUS key to display the screen menu.
2 Select “6 SETUP.”
3 Press F1 “TYPE” to display the screen change menu.
4 Select “Macro.” The macro instruction setting screen appears.

```
 Macro Command                         JOINT 30%
      Instruction name  Program    Assign
   1 [               ][         ]  --  [    ]
   2 [               ][         ]  --  [    ]
   3 [               ][         ]  --  [    ]
   4 [               ][         ]  --  [    ]
   5 [               ][         ]  --  [    ]

 [TYPE]   CLEAR
```

5 For macro instruction input, press the ENTER key to display the character string input screen, then enter characters using an F key.

```
 Macro Command
      Instruction name
   1 [              ] [
   2 [              ] [
```

ENTER

```
 Macro Command
      Instruction name
   1 [              ] [

ABCDEF   GHIJKL   MNOP
```

F1

```
                                       JOINT 30%
 1   Words
 2   Upper Case
 3   Lower Case
 4   Options                  ---Insert---
 Macro Command
      Instruction name  Program    Assign
   1 [hand           ][         ]  --  [    ]


 abcdef   ghijkl   mnopqr   stuvwx   yz-@*. >
```

Upon completion of input, press the ENTER key.

```
 Macro Command
      Instruction name
   1 [handlopen     ] [
```

ENTER

```
 Macro Command                         JOINT 30%
      Instruction name  Program    Assign
   1 [handlopen      ][         ]  --  [    ]


 [TYPE]   CLEAR              [CHOICE]
```

NOTE No duplicate macro instruction definition is allowed.

6 For macro program input, press F4 [CHOICE] to display a directory of programs, then choose a program from the directory. When the macro program name is entered without the macro name, the program name will be used as the macro name.

```
[CHOICE]
F4
```

```
                                          JOINT 30%
1 PROGRAM1                 5 SAMPLE1
2 PROGRAM2                 6 SAMPLE2
3 HOPN1                    7
4 HCLS1                    8 ---next page---
Macro Command
   Instruction name  Program    Assign
 1 [hand1open      ] [         ]  -- [    ]

[TYPE]  CLEAR
```

```
1 PROGRAM1          5
2 PROGRAM2          6
3 HOPN1
4 HCLS1           ENTER
```

```
Macro Command                             JOINT 30%
   Instruction name  Program    Assign
 1 [hand1open      ] [HOPN1    ]  -- [    ]

[TYPE]  CLEAR              [CHOICE]
```

7 For device assignment, press F4 “[CHOICE]” to display a directory of programs, then choose a program from the directory.

```
[CHOICE]
F4
```

```
                                          JOINT 30%
1 --                       5 SP
2 UK                       6 DI
3 SU                       7 RI
4 MF                       8 ---NEXT---
Macro Command
   Instruction name  Program    Assign
 1 [hand1open      ] [HOPN1    ]  -- [    ]

[TYPE]  CLEAR              [CHOICE]
```

NOTE On R–J3*i* Mate, SP (operator control panel) cannot be used for the device to be assigned.

```
1 --                5
2 UK                6
3 SU
4 MF              ENTER
```

```
Macro Command                             JOINT 30%
   Instruction name  Program    Assign
 1 [hand1open      ] [HOPN1    ]  MF [    ]

[TYPE]  CLEAR              [CHOICE]
```

8 Enter a desired device number.

```
         JOINT 30%
    Program  Assign
] [HOPN1 ]  MF[  ]
     1   ENTER
```

```
Macro Command                             JOINT 30%
   Instruction name  Program    Assign
 1 [hand1open      ] [HOPN1    ]  MF [ 1 ]

[TYPE]  CLEAR
```

> **CAUTION**
>
> After all macro instructions are set, the setting information should be saved in external storage (floppy disk, for example) in case the information needs to be re–loaded. Otherwise, the current setting information would be lost when it is changed.

9 For macro instruction deletion, move the cursor to a desired field, then press F2"CLEAR" while holding down the SHIFT key.

```
Macro Command
 Instruction name  Pros
 1 [hand1open  ] [HOPN1
 2 [hand1close ] [HCLS1

 [TYPE]  CLEAR
```

F2

```
Macro Command                      JOINT 30%
   Instruction name  Program   Assign
 1 [              ][HOPN1   ]  MF [  1 ]
 2 [hand1close    ][HCLS1   ]  MF [  2 ]
 3 [              ][        ]  -- [    ]
 4 [              ][        ]  -- [    ]

 [TYPE]  CLEAR
```

10 "Clear OK?" appears.

– To delete the macro instruction, press F4 "YES."

– To cancel deletion of the macro instruction, press F5 "NO."

```
      YES      NO
```

F4

```
Clear OK?
                              YES      NO
```

9.1.2 Executing macro instructions

A macro instruction can be executed by:

- Selecting an item on the manual operation screen on the teach pendant (with the SHIFT key held down)
- Pressing user keys on the teach pendant (without pressing the SHIFT key)
- Pressing user keys on the teach pendant (with the SHIFT key held down)
- SDI, RDI, UI
- Calling the macro instruction from the program

When a macro instruction is started, the macro program is executed in the same way as an ordinary program is executed, except for the following restrictions:

- The single step mode is disabled. The continuous operation mode is always used.
- The macro program is always aborted without the pausing status.
- The macro program is always executed starting from the first line.

When a macro program includes a motion instruction (uses a motion group), the motion enabled state must be set to execute the macro instruction. When no motion group is used, the motion enabled state need not be set.

The motion enabled state is set when:

- ENBL is on.
- SYSRDY output is on. (Servo power supply is on)

Table 9–1. Macro Instruction Execution Conditions

		Without a motion group	With a motion group
MF [1 to 99] SU [1 to 7]	TP enabled	Executable(*1)	Executable
UK [1 to 7]		Executable	–
SP [4 to 5] DI [1 to 999] RI [1 to 5] UI [1 to 8]	TP disenabled	Executable	Executable

NOTE (*1) Even when the teach pendant is disabled, a macro instruction that does not possess a motion group can be executed from an MF or SU by setting system variable $MACRTPDSBEXE = TRUE.

*) It is possible to supply an argument in a macro instruction call in a program and use it in a macro program. For details, see Section 4.7.5, "Arguments."

Procedure 9–2 Executing a macro instruction using the teach pendant (manual operation screen)

Condition ■ The teach pendant is enabled.

NOTE Even when the teach pendant is disabled, a macro instruction that does not possess a motion group can be executed from an MF or SU by setting system variable $MACRTPDSBEXE = TRUE.

■ A device from MF[1] to MF[99] is set using the macro instruction setting screen.

```
Macro Command                        JOINT 30%
   Instruction name  Program   Assign
 1 [hand1open      ][HOPN1   ]  MF [  1 ]
 2 [hand1close     ][HCLS1   ]  MF [  2 ]
 3 [               ][        ]  -- [    ]
 4 [               ][        ]  -- [    ]

[TYPE]
```

Step

1 Press the MENUS key to display the screen menu.

2 Select “3 MANUAL FCTNS.”

3 Press F1, [TYPE] to display the screen change menu.

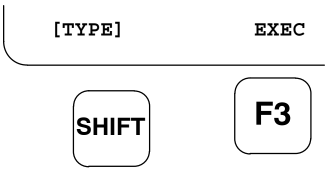
4 Select “Macros.” The manual operation screen appears.

```
MANUAL MACROS                        JOINT 30%
      Instruction                        1/3
 1  hand1open
 2  hand1close



[TYPE]            EXEC
```

WARNING

The macro program is started in the next step, causing the robot to make a motion. Before executing the operation, the operator should check that no persons and no unnecessary equipment are in the work area. Otherwise, injury or property damage could occur.

5 To start a desired macro instruction, press F3 “EXEC” while holding down the SHIFT key. The macro program is started.



Hold down the SHIFT key until the execution of the macro program is completed.

NOTE When the macro program contains a motion group, hold down the shift key until execution of the macro program terminates. If the shift key is released while the macro is being executed, the macro program is stopped. When the macro program does not contain a motion group, program execution continues even if the shift key is released.

CAUTION

If the SHIFT key is released during execution, the macro program is terminated forcibly. Note that when execution is interrupted and F3 “EXEC” is pressed again, the macro program is executed from the first line again.

Procedure 9–3 Executing a macro instruction using the teach pendant (using a user key)

Condition ■ The teach pendant is enabled.

NOTE Even when the teach pendant is disabled, a macro instruction that does not possess a motion group can be executed from an MF or SU by setting system variable $MACRTPDSBEXE = TRUE.

■ A device from UK[1] to UK[7] or SU[1] TO SU[7] is set on the macro instruction setting screen.

```
Macro Command                        JOINT 30%
     Instruction name  Program    Assign
  1 [hand1open      ][HOPN1    ]  SU [  1 ]
  2 [hand1close     ][HCLS1    ]  SU [  2 ]
  3 [               ][         ]  -- [    ]
  4 [               ][         ]  -- [    ]

[ TYPE ] CLEAR
```

Step 1 To start a macro instruction on the teach pendant, use the assigned user key on the teach pendant.

> **WARNING**
> **The macro program is started in the next step, causing the robot to make a motion. Before executing the operation, the operator should check that no persons and no unnecessary equipment is in the work area. Otherwise, injury or property damage would occur.**

2 When a user key from UK[1] to UK[7] is assigned to the macro instruction, press the assigned user key to start the macro instruction.

NOTE A macro instruction that possesses a motion group cannot be executed using a device from UK[1] to UK[7]. A device from SU[1] to SU[7] must be assigned to such a macro instruction.

3 When a device from SU[1] to SU[7] is assigned to the macro instruction, press the user key while holding down the SHIFT key.

NOTE When the macro program contains a motion group, hold down the shift key until execution of the macro program terminates. If the shift key is released while the macro is being executed, the macro program is stopped. When the macro program does not contain a motion group, program execution continues even if the shift key is released.

> **CAUTION**
> If the SHIFT key is released during execution, the macro program is terminated forcibly. Note that when execution is interrupted and F3 "EXEC" is pressed again, the macro program is executed from the first line again.

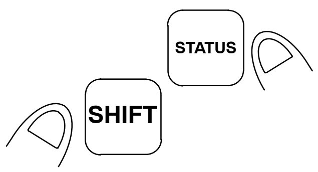
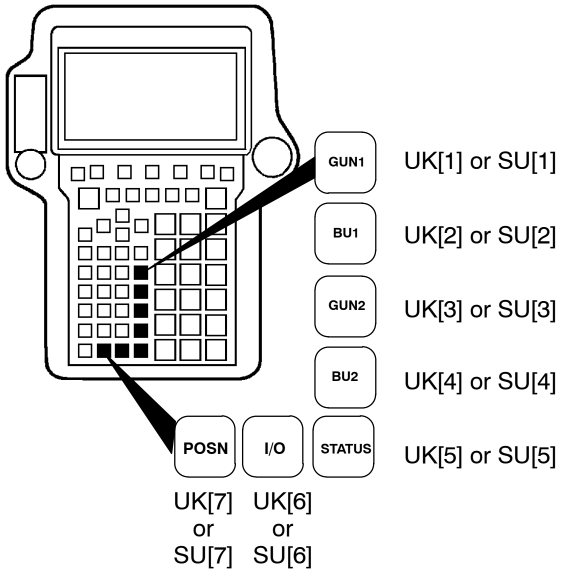


Figure 9–2. User Keys on Teach Pendant



CAUTION

When a key on the teach pendant is assigned to a macro instruction, it becomes that macro instructions device, and the key can no longer be used for its original function.

Procedure 9–4 Execution of macro command using SDI,RDI and UI

Condition
- The teach pendant must be disabled.
- DI[1 to 99], RI[1 to 24] or UI[7] is specified as the device in the macro instruction setting screen.

```
Macro Command                      JOINT  30 %
     Instruction name   Program  Assign
  1  [RETURN TO REFPOS][REFPOS   ]UI[ 7]
  2  [WORK1 CLAMP     ][CLAMP1   ]DI[ 2]
  3  [PROCESSING PREP ][PREP     ]RI[ 3]
  4  [                ][         ]--[  ]

 [ TYPE ] CLEAR
```

Step
1. To start the macro command using SDI, RDI or UI, input the digital signal from the external device or directly input these signals in the I/O screen on the teach pendant.
2. When SDI or RDI or UI which is set in the macro instruction setting screen is input, the macro command which is assigned to the signal will be started.

NOTE Moreover, $MACROUOPENBL can be changed in the system variable screen displayed at controlled start.

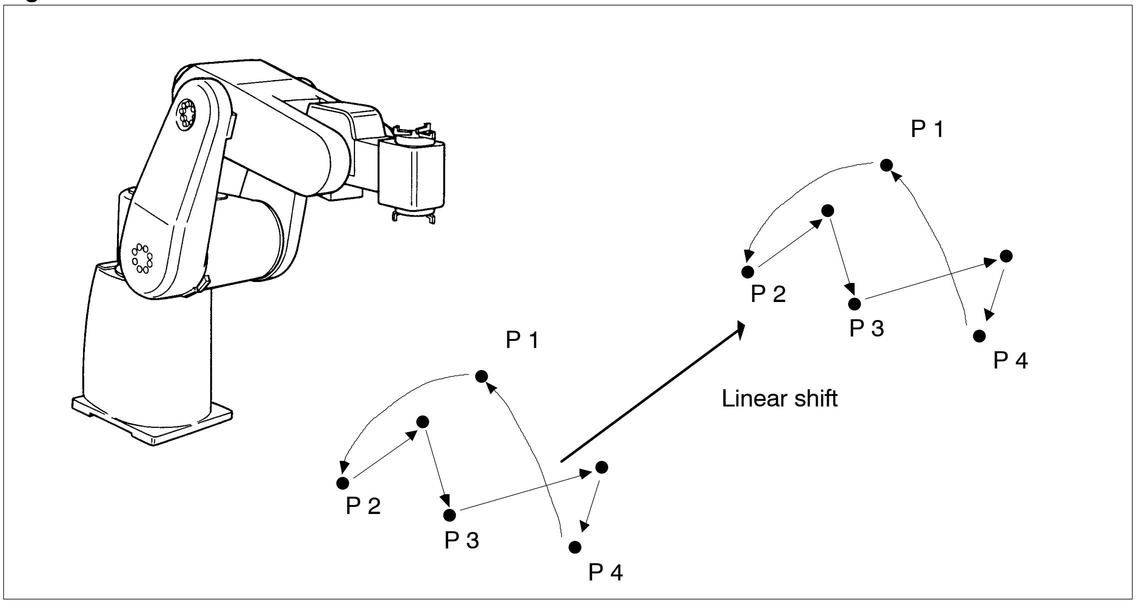
9.2 Shift Functions

The shift functions shift the specified positions for the operation instructions within a certain range of a previously taught program to other locations.

The shift functions perform the following:

- Shift the position data for the operation instructions within the entire range or within a certain range of an existing program.
- Insert the shift results into a new or existing program.
- Repeat the same shift on another program.

Figure 9–3. Shift



The following rules apply to converted position data:

Rules governing position data:

- Position data having Cartesian coordinates is converted to Cartesian coordinates. Position data with joint coordinates is converted to joint coordinates.
- If converted joint coordinate position data falls outside the variable axis area, it is stored as unspecified. Converted Cartesian coordinate position data is stored as is even if it falls outside the variable axis area.
- Position data in the position registers is not converted.
- The position data with joint coordinates for operation instructions involving incremental instructions is stored as unspecified.

Rules governing the Cartesian coordinate system number (UT, UF) in position data having Cartesian coordinates:

- The Cartesian coordinate system number is not changed due to conversion.
- During conversion (on the shift information input screen), a user coordinate system number (UF) of 0 is used. Position data is converted to data in the Cartesian coordinate system with a UF of 0 (world coordinate system) and displayed.

Rules governing the configuration (joint placement and turn number) of position data having Cartesian coordinates:

- The configuration is not changed as a result of the conversion.
- For the turn number, if the conversion causes rotation about the wrist axis by 180° or greater, the turn number for the axis is optimized, and a message appears so that the user can determine whether to accept it.

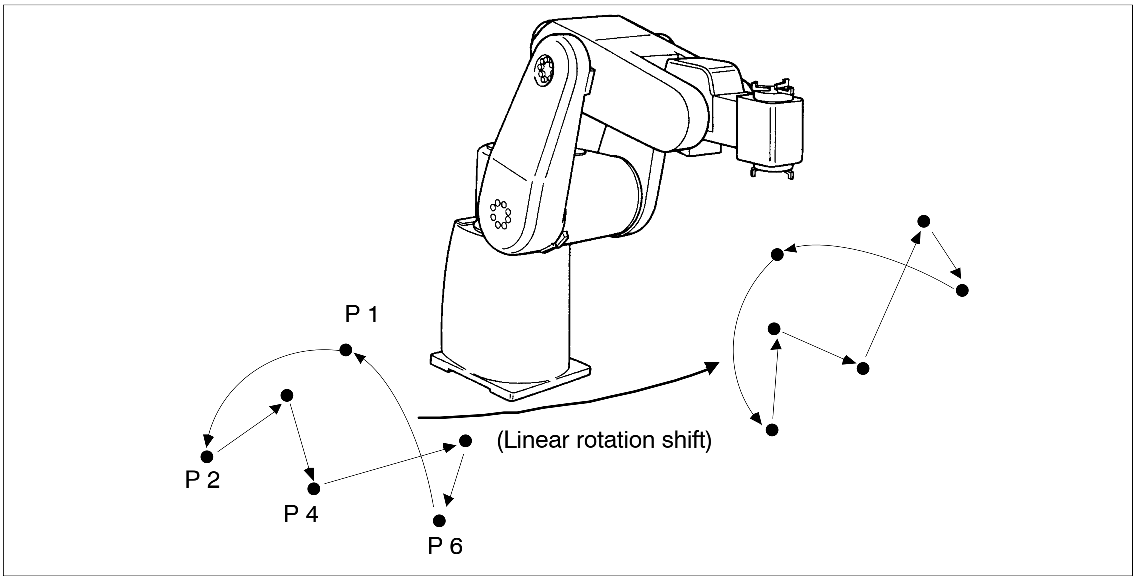
The following shift functions are available:

- Program shift : Performs a 3–dimensional linear shift or linear rotation shift.
- Mirror shift : Performs a 3–dimensional symmetrical shift about a specified mirror plane.
- Angle–input shift : Performs a rotation shift about a specified rotation axis.

9.2.1 Program shift function

The program shift function performs a linear shift or linear rotation shift on the specified positions for the operation instructions within a certain range of a previously taught program.

Figure 9–4. Linear Rotation Shift



The program shift function requires the following setup:

Program name setting

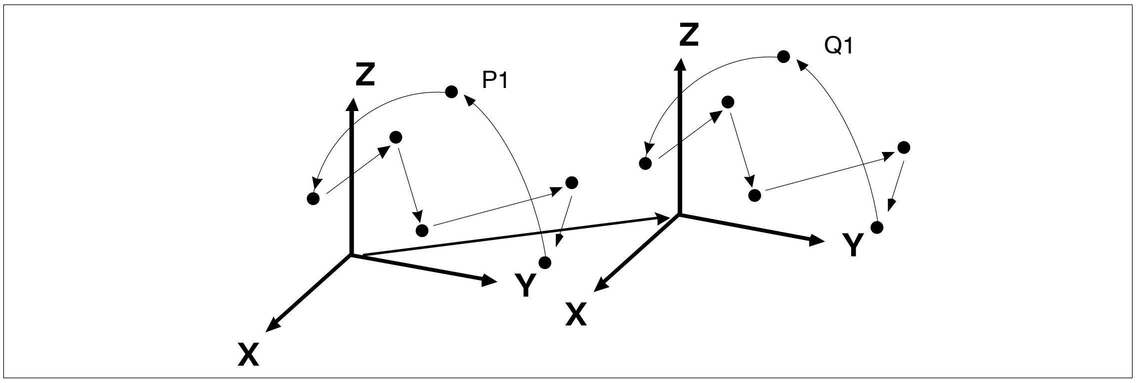
Program name setting specifies the name of the source program, the range of lines on which the shift is to be performed, as well as the name of the program into which the shift results are to be inserted and the line at which they are to be inserted.

Shift information input

Shift information input specifies the direction and amount of the program shift function. Two types of shift are supported: linear shift and linear rotation shift. The shift direction and amount can be specified in either of two ways: representative point specification and direct specification.

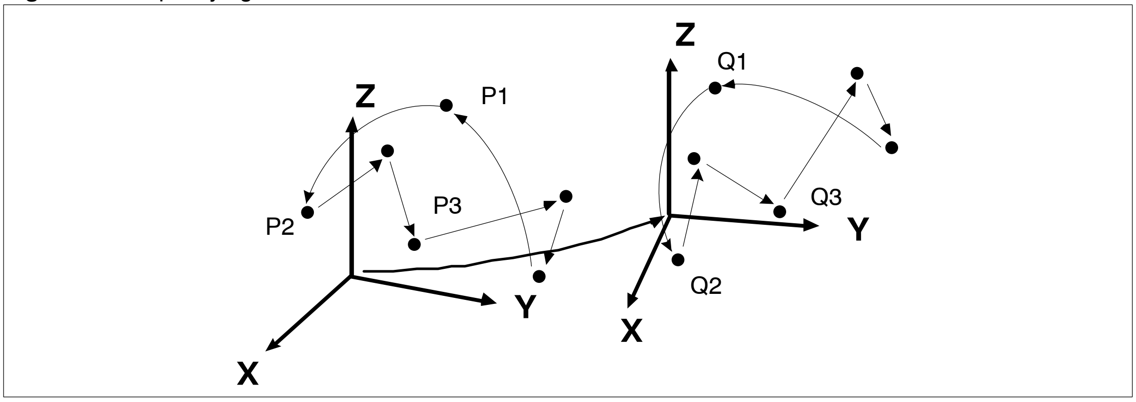
- In representative point specification, the user indicates (specifies) representative source and destination points to determine the shift direction and amount.
  For a linear shift, one source point (P1) and one destination point (Q1) must be indicated (specified).

Figure 9–5. Specifying a Linear Shift



For a linear rotation shift, three source points (P1, P2, and P3) and three destination points (Q1, Q2, and Q3) must be indicated (specified).

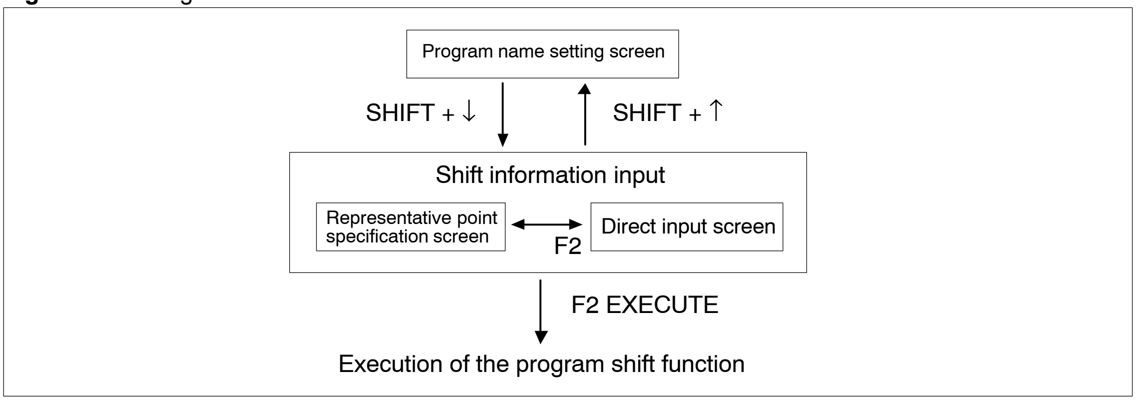
Figure 9–6. Specifying a Linear Rotation Shift



- In direct specification, the user directly specifies the direction and amount (X, Y, Z) of linear shift. In direct specification, linear rotation shift cannot be specified.

To execute the program shift function, use the program shift screen PROGRAM SHIFT. The figure below shows how to navigate through the program shift screen.

Figure 9–7. Program Shift Screen



The program name input screen contains the following items:

Table 9–2. Contents of the Program Name Input Screen

Item	Description
Original Program	Specifies the name of the source program.
RANGE	Specifies the type of the desired range of the source program. ● WHOLE = Performs shift on the entire program. ● PART = Performs shift on part of the program.
Start line	Specifies the start line of the desired range of the source program. If WHOLE is set to all, this item cannot be specified.
End line	Specifies the end line of the desired range of the source program. If WHOLE is set to all, this item cannot be specified.
New Program	Specifies the program into which the shift results are to be inserted. If a new program name is specified, a new program is created with that name. If the name of an existing program is specified, the results are inserted into that program.
Insert line	Specifies the line at which the shift results are to be inserted, if insertion of the results are to be into an existing program is specified. If the program is a new one, this item cannot be specified.

The representative point specification screen contains the following items:

Table 9–3. Contents of the Representative Point Specification Screen

Item	Description
Position data	Indicates the position of the point where the cursor is currently located. The position is always represented by coordinates in the world coordinate system.
Rotation	Specifies whether rotation is to be performed.
Source position	Specifies the position of a representative source point.
Destination position	Specifies the position of a representative destination point.
REFER	F4 REFER allows the use of a position variable or position register in the source program as the position of a representative point.

Procedure 9–5 Executing the program shift function

Condition ■ The program on which the shift is to be performed exists.

```
TEST2                                   JOINT  30%
                                               1/6
   1:J  P[1]   100% FINE
   2:J  P[2]   70% CNT50
   3:L  P[3]   1000cm/min CNT30
   4:L  P[4]   500m/sec FINE
   5:J  P[1]   100% FINE
[End]

POINT                                   TOUCHUP >
```

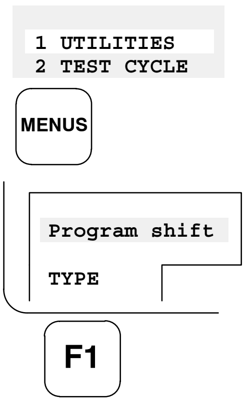
Step

1 Press the screen selection key. The screen menu appears.

2 Select 1, UTILITIES.

3 Press F1, [TYPE]. The screen switching menu appears.

4 Select Program shift. The program name input screen appears.




```
PROGRAM SHIFT                           JOINT  10%
Program                                        1/6
  1   Original Program:          [Test1     ]
  2   Range:                         WHOLE
  3   Start line: (not used)         ***
  4   End line:   (not used)         ***
  5   New Program:               [Test1     ]
  6   Insert line:(not used)         ***

Use shifted up, down arrows for next page

 [TYPE]                                      >
CLEAR                                        >
```

5 Specify the necessary items.

6 After specifying the items, go to the next screen with SHIFT + ↓. The representative point specification screen appears. To return to the previous screen, use SHIFT + ↑.

SHIFT ⇩

```
PROGRAM SHIFT                      JOINT  10%
Shift amount/Teach
 Position data    P1
X    *****      Y    *****      Z    *****

1    Rotation                      OFF

2    Source position         P1


3    Destination position    Q1



[TYPE]   EXECUTE             ON      OFF    >
```

7 For a shift with rotation, set "Rotation" to ON.

```
PROGRAM SHIFT                      JOINT  10%
Shift amout/Teach
Position data
X    *****      Y    *****      Z    *****

1    Rotation                      ON

2    Source position         P1
                             P2
                             P3
3    Destination position    Q1
                             Q2
                             Q3

[TYPE]   EXECUTE             ON      OFF    >
```

8 Specify representative source and destination points.

REFER RECORD >

SHIFT F5

```
PROGRAM SHIFT                      JOINT  10%
Shift amount/Teach
Position data    Q1
X    1234.4     Y    100.0      Z    120.0

1    Rotation                        ON

2    Source position         P1      Recorded
                             P2      Recorded
                             P3      Recorded
3    Destination position    Q1      Recorded
                             Q2
                             Q3

[TYPE]   EXECUTE          REFER     RECORD  >
```

9 For reference point input, press F4 REFER. Select F4 P[] or F5 PR[] to enter arguments.

REFER RECORD >

F4

P[] PR[] >

F4

```
3    Destination position    Q1      Recorded
                             Q2      P[5]
                             Q3

[TYPE]   EXECUTE             P[]    PR[]   >
```

10 After setting shift information, press F2, EXECUTE and then F4, YES. The conversion results are written into the program.

```
[TYPE]  EXECUTE
```

F2

```
Execute transform?
          YES    NO
```

F4

```
TEST2                              JOINT  30%
                                          1/6
   1:J  P[1]   100% FINE
   2:J  P[2]   70% CNT50
   3:L  P[3]   1000cm/min CNT30
   4:L  P[4]   500m/sec FINE
   5:J  P[1]   100% FINE
[End]

POINT                              TOUCHUP >
```

11 The direct input screen appears with F2, DIRECT on the next page. Specify the shift amount directly.

```
PROGRAM SHIFT                     JOINT  10%
Shift amount/Direct entry
  1   X  (mm)                   1888.92
  2   Y  (mm)                    239.87
  3   Z  (mm)                     50.52


[TYPE]   EXECUTE                            >

CLEAR     TEACH                             >
```

NOTE Set the shift amount using coordinates in the world coordinate system.

12 After setting the shift amount, press F2, EXECUTE to execute the shift.

13 If the turn number is changed due to the shift, the user is notified and asked which to select.

```
Select P[3]:J5 angle   (183°)
183°    -177°    uninit           QUIT  >
```

14 F1 indicates the axial angle associated with the changed turn number.
F2 indicates the axial angle associated with the original turn number.
F3 uninit causes the data to become unspecified data.
F5 QUIT interrupts the conversion.

15 To erase all the shift information, press F1, CLEAR on the next page. Then, the currently selected program is specified as the source program.

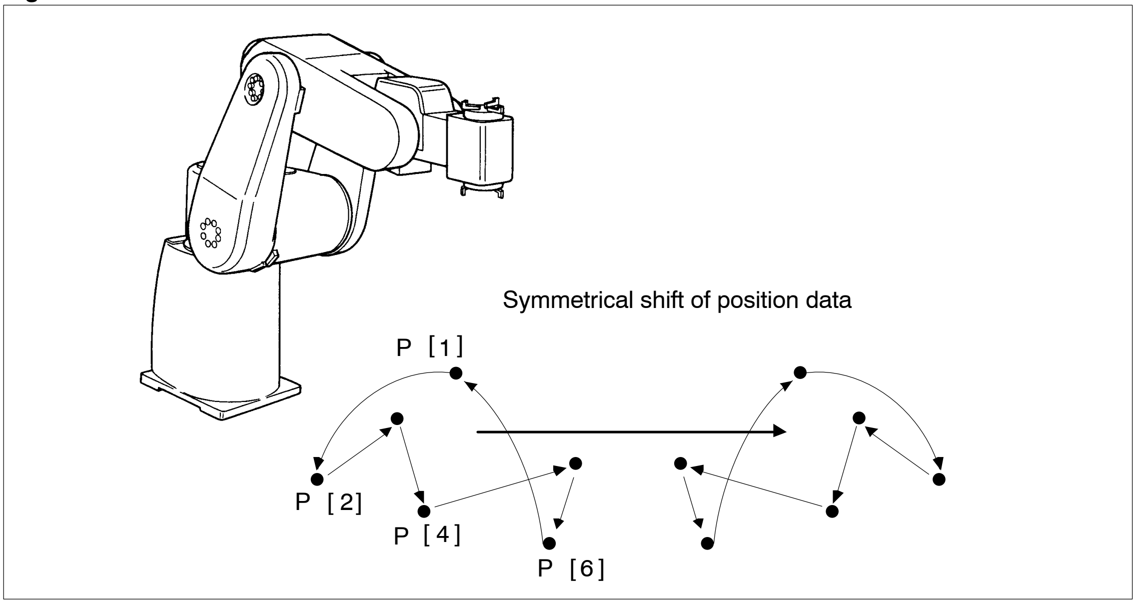
```
CLEAR   TEACH
```

F1

9.2.2 Mirror shift function

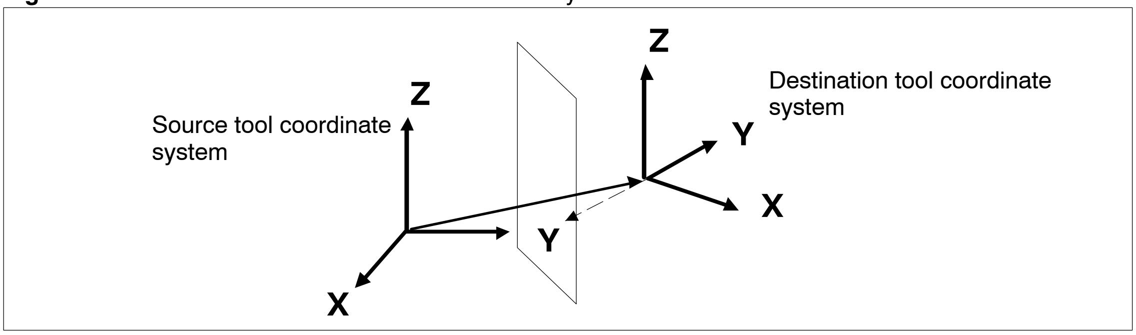
The mirror shift function shifts the specified positions for the operation instructions in a certain range of an already taught program symmetrically about a plane.

Figure 9–8. Mirror Shift Function



Theoretically, the mirror shift function converts the attitude of the tool from right–handed coordinates to left–handed coordinates. In reality, however, the attitude is returned to the right–handed coordinate system by inverting the Y–axis because no left–handed coordinates exist.
The mirror shift function, therefore, performs conversion most naturally when the plane of symmetry is parallel to the XZ plane of the tool coordinate system.

Figure 9–9. Conversion from One Tool Coordinate System to Another with the Mirror Shift Function



> **⚠ CAUTION**
> The tool coordinate system must be established accurately. The mirror shift function requires that the Z–axis match the tool direction.

> **⚠ CAUTION**
> The tool center point (TCP) must be set accurately to ensure correct operation with the points resulting from a symmetrical shift. Otherwise, the points resulting from the shift will contain offset values.

The mirror shift function requires the following setup:

Program name setting

Program name setting specifies the name of the source program, the range of lines on which the shift is to be performed, as well as the name of the program into which the shift results are to be inserted and the line at which they are to be inserted.

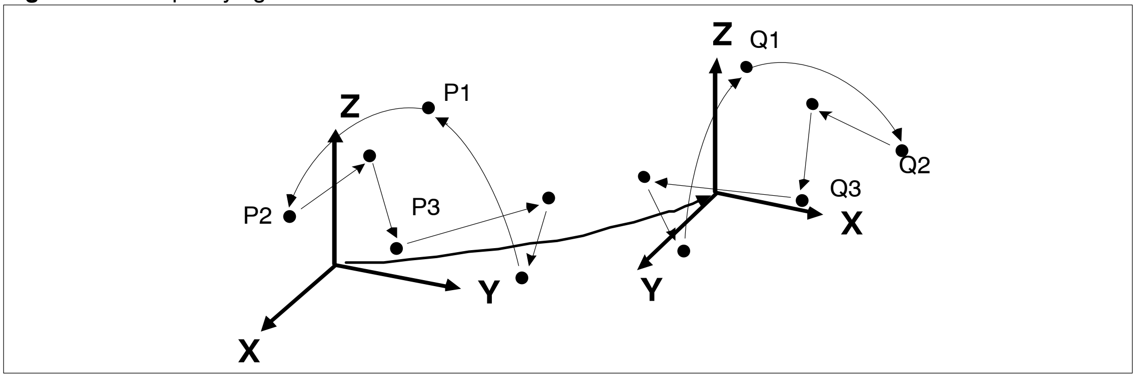
Shift information input

Shift information input specifies the direction and amount of the mirror shift. Two types of shift are supported: symmetrical shift and symmetrical rotation shift.

- In representative point specification, the user indicates (specifies) representative source and destination points to determine the shift direction and amount.
  For a symmetrical shift, one source point (P1) and one destination points (Q1), two points in total, must be indicated (specified).

  For a symmetrical rotation shift, three source points (P1, P2, and P3) and three destination points (Q1, Q2, and Q3), six points in total, must be indicated (specified).

Figure 9–10. Specifying the Mirror Shift Function



To execute the mirror shift function, use the mirror screen MIRROR IMAGE SHIFT. The explanation of the program shift screen also applies to the mirror screen.

Procedure 9–6 Executing the mirror shift function

Condition ■ The program on which the shift is to be performed exists.

```
TEST2                                    JOINT  30%
                                                1/6
   1:J  P[1]    100% FINE
   2:J  P[2]    70% CNT50
   3:L  P[3]    1000cm/min CNT30
   4:L  P[4]    500m/sec FINE
   5:J  P[1]    100% FINE
[End]

POINT                                   TOUCHUP  >
```

Step

1. Press the screen selection key. The screen menu appears.
2. Select 1, UTILITIES.
3. Press F1, [TYPE]. The screen switching menu appears.
4. Select Mirror Image. The program name input screen appears.

```
1 UTILITIES
2 TEST CYCLE
```

MENUS

```
Mirror Image

TYPE
```

F1

```
MIRROR IMAGE SHIFT                       JOINT  10%
Program                                         1/6
  1   Original Program:             [Test1     ]
  2   Range:                             WHOLE
  3   Start line: (not used)             ***
  4   End line:   (not used)             ***
  5   New Program                   [Test1     ]
  6   Insert line:(not used)             ***

Use shifted up, down arrows for next page

[TYPE]                                          >

CLEAR                                           >
```

NOTE The program selected last with the list screen is automatically selected as the source program.

5. Specify the necessary items.
6. After specifying the items, go to the next screen with SHIFT + ↓. The representative point specification screen appears. To return to the previous screen, use SHIFT + ↑.

SHIFT ⇩

```
MIRROR IMAGE SHIFT                       JOINT  10%
Shift amount/Teach
 Position data
X    *****      Y    *****      Z    *****

1   Rotation                          OFF

2   Source position          P1


3   Destination position     Q1



 [TYPE]   EXECUTE              ON      OFF    >

CLEAR     DIRECT                              >
```

7 For shift with rotation, set ”Rotation” to Yes.

```
MIRROR IMAGE SHIFT                JOINT  10%
Shift amount/Teach
 Position data
X   *****     Y   *****     Z   *****

1   Rotation                    ON

2   Source position          P1
                             P2
                             P3
3   Destination position     Q1
                             Q2
                             Q3

[TYPE]   EXECUTE             ON     OFF   >
```

8 Specify representative source and destination points.
For details, see the explanation of the program shift function.

9 After setting the shift amount, press F2 EXECUTE to execute the shift.

```
[TYPE]  EXECUTE
```

F2

```
Execute transform?
          YES    NO
```

F4

> **WARNING**
> **Avoid moving the robot to a position that is not correctly shifted. Check the shift results before moving the robot. Otherwise, serious problems can occur.**

10 To erase all shift information, press F1 CLEAR on the next page.

```
CLEAR   TEACH
```

F1

9.2.3 Angle–input shift function

The angle–input shift function allows the user to perform a program shift by directly entering three or four representative points and an angular displacement. It also allows the user to perform multiple shifts at equal intervals on the same circumference at one time by specifying the iteration.

If many locations on the same circumference are subject to the same machining, such as the holes on a car wheel, this function allows the user to create position data for all the locations to be machined by specifying only a single location.

The angle–input shift function requires the following setup:

Program name setting

Program name setting specifies the name of the source program, the range of lines on which the shift is to be performed, as well as the name of the program into which the shift results are to be inserted and the line at which they are to be inserted.

Shift information input

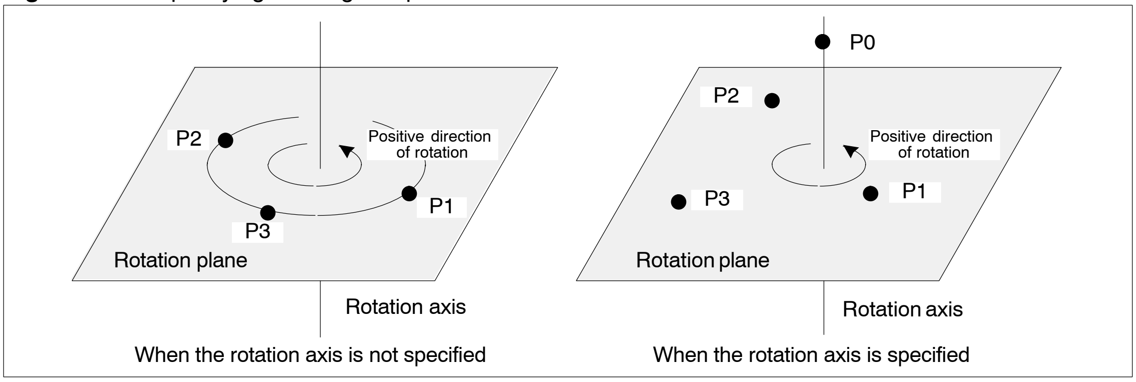
Shift information input specifies the representative points for determining the rotation axis for the angle–input shift function and sets the angular displacement and shift iteration. The representative points can be specified in either of two ways: one in which the rotation axis is specified and one in which it is not specified.

- If the rotation axis is not specified, three representative points (P1, P2, and P3) on the same circumference must be specified. With these three points, the rotation plane and axis are automatically calculated. The intersection of the rotation plane and axis (rotation center) is set as representative point P0.
  Rotation center P0, which is set automatically, can be changed directly later. From the second conversion on, the position of the rotation center can be compensated for by enabling the rotation axis.
- If the rotation axis is specified, a point on the rotation axis must be specified for representative point P0 and any three points on the rotation plane must be specified for representative points P1, P2, and P3. (P1, P2, and P3 need not be on the same circumference.) The rotation plane is determined with representative points P1, P2, and P3. The axis that is vertical to the rotation plane and which passes through representative point P0 is determined as the rotation axis.

In either way, the more distant the representative points P1, P2, and P3, the more precise the conversion.

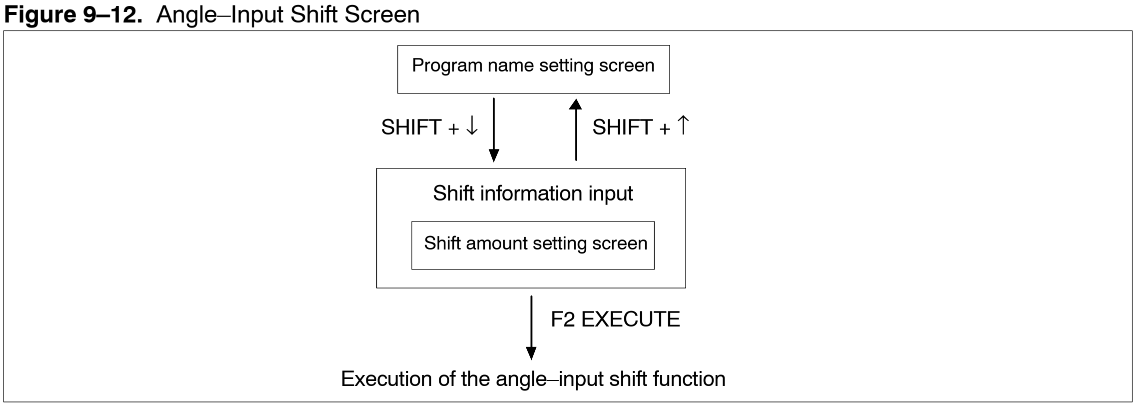
The direction of rotation is regarded as being positive when the rotation is from representative point P1 to P2.

Figure 9–11. Specifying the Angle–Input Shift Function



To execute the angle–input shift function, use the angle–input shift screen ANGLE ENTRY SHIFT. The figure below shows how to navigate through the angle–input shift screen.

Figure 9–12. Angle–Input Shift Screen



The items on the program name setting screen are the same as those on corresponding screen for the program shift function.

The shift amount setting screen contains the following items:

Table 9–4. Contents of the Shift Amount Setting Screen

Item	Description
Rotation plane	Specifies the positions of the representative points for determining the rotation plane. If the rotation axis is not specified, these points must be on the same circumference so that the rotation center can be calculated. If the rotation axis is specified, the representative points need not necessarily be on the same circumference. The positions must be specified with coordinates in the world coordinate system.
Rotation axis enable	Specifies how the rotation axis is to be determined from the representative points. The representative points must be specified differently depending on the setting made for this item.
Rotation axis	Specifies the position of representative point P0 for determining the rotation axis. This item is available only when Rotation axis enable is set to TRUE. Only representative point P0 can be specified directly with position data (numeric values) in any coordinate system. To specify P0 directly, position the cursor to this item and press the Enter key. The rotation axis direct specification screen appears.
Angle	Specifies the angular displacement (in degrees) by which the shift is to be performed with the rotation axis and plane determined with the representative points. Enter an unsigned real number directly. (The plus sign need not be entered.) The direction of rotation is regarded as being positive when the rotation is from representative point P1 to P2.
Repeating times	Specifies the conversion iteration. If the locations to be machined are arranged at equal intervals on the same circumference, specifying the iteration allows the user to machine all the locations by specifying a single location. If the iteration is 2 or greater, a comment line is automatically inserted at the beginning of the program resulting from the shift. Consider the following example: Source program: Program A `1:J  P[1]   100% FINE` `2:L  P[2]   1500mm/sec FINE` If conversion is performed with the ”angular displacement” set to 20o, ”iteration” set to 3, and ”destination program” set to program B, program B will be as follows: Destination program: Program B `1:!Angle entry shift 1 (deg 20.00)` `2:J  P[1]   100% FINE` `3:L  P[2]   1500mm/sec FINE` `4:!Angle entry shift 2 (deg 40.00)` `5:J  P[3]   100% FINE` `6:L  P[4]   1500mm/sec FINE` `7:!Angle entry shift 3 (deg 60.00)` `8:J  P[5]   100% FINE` `9:L  P[6]   1500mm/sec FINE` The position data in program B is as follows: P[1]: Position resulting from rotating P[1] in program A by 20° P[2]: Position resulting from rotating P[2] in program A by 20° P[3]: Position resulting from rotating P[1] in program A by 40° P[4]: Position resulting from rotating P[2] in program A by 40° P[5]: Position resulting from rotating P[1] in program A by 60° P[6]: Position resulting from rotating P[2] in program A by 60°
REFER	F4, REFER allows the use of a position variable or position register in the source program as the position of a representative point.

Procedure 9–7 Executing the angle–input shift function

Condition ■ The program on which the shift is to be performed exists.

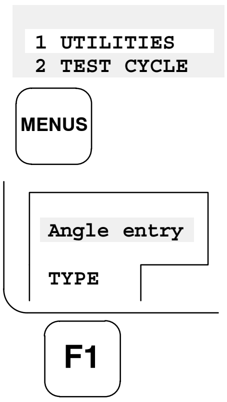
Step

1 Press MENUS. The screen menu appears.

2 Select 1, UTILITIES.

3 Press F1, [TYPE]. The screen switching menu appears.

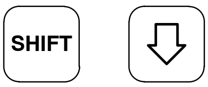
4 Select Angle entry. The program name input screen appears.




```
ANGLE ENTRY SHIFT                    JOINT  10%
Program
   1  Original Program:       [          ]
   2  Range:                        WHOLE
   3  Start line:  (not used)       ****
   4  End line:    (not used)       ****
   5  New Program:            [          ]
   6  Insert line:(not used)        ****

Use shifted up, down arrows for next page

[TYPE]                                        >
CLEAR                                         >
```

5 Specify the necessary items.

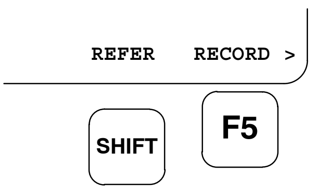
6 After specifying the items, go to the next screen with SHIFT + ↓. The shift amount setting screen appears. To return to the previous screen, use SHIFT + ↑.




```
ANGLE ENTRY SHIFT                   JOINT  10%
Shift amount
 Position data of P1
X:*****.**     Y:*****.**     Z:*****.**


1   Rotation plane          P1:
2                           P2:
3                           P3:
4   Rotation axis enable:        FALSE
5   Rotation axis           P0:Not used
6   Angle(deg):                   0.00
7   Repeating times:                 1


[TYPE]   EXECUTE        REFER    RECORD >
CLEAR   >
```

7 For shift with the rotation axis specified, set ”Rotation axis specification” to TRUE. If required, specify ”Iteration.”

8 Specify the representative points.




```
1   Rotation plane          P1:Recorded
2                           P2:
3                           P3:
```

9 For reference point input, press F4, REFER. Select F4 [P] or F5 [PR] to enter arguments.

```
      REFER    RECORD >
```

F4

```
        P[]    PR[]   >
```

F4

```
1   Rotation plane        P1:Recorded
2                         P2:P[5]
3                         P3:
```

10 Enter the angular displacement.

11 After setting the shift information, press F2 EXECUTE to execute the shift.

```
  [TYPE]  EXECUTE
```

F2

12 If the turn number is changed due to the conversion, the user is notified and prompted to make a selection.

```
Execute transform?
          YES    NO
```

F4

```
Repeat3:Select P[1]:J6   (183°)
183°    -177°    uninit           QUIT  >
```

13 F1 indicates the axial angle associated with the changed (optimized) turn number.
F2 indicates the axial angle associated with the original turn number.
F3 uninit causes the data to become unspecified data.
F5 QUIT interrupts the conversion.
Select one of the above keys.

14 To directly enter the position data for representative point P0, position the cursor to the P0 line and press the Enter key. The rotation axis direct specification screen appears.

```
ANGLE ENTRY SHIFT               JOINT  10%
Shift amount
Rotation center axis direct entry
  1   Frame                    USER FRAME1
  2   X:                              0.00
  3   Y:                              0.00
  4   Z:                              0.00

 [TYPE]   EXECUTE         [CHOICE]      >

CLEAR                                   >
```

15 To specify the position of representative point P0 with numeric values in any coordinate system, position the cursor to line Frame and press F4, [CHOICE]. From the menu that appears, select the desired coordinate system.

```
      [CHOICE]       >
```

F4

16 Provide the other necessary shift information has been set, press F2, EXECUTE to execute the shift.

```
[TYPE]  EXECUTE
```

F2

```
Execute transform?
         YES    NO
```

F4

17 To erase all the shift information, press F1, CLEAR on the next page.

```
CLEAR   TEACH
```

F1

9.3 Coordinate System Change Shift Functions

The coordinate system change shift functions change the tool coordinate system (tool) or user coordinate system for the operation instructions within a certain range of an already taught program, and convert the position data so that the TCP is located at the same position, considering the shift amount resulting from the change from the old to the new coordinate system.

NOTE The coordinate system change shift functions allow the user to specify that the position data not be converted.

Coordinate system change shift functions

The coordinate system change shift functions perform the following:

- Change the tool coordinate system or user coordinate system number in the position data (Cartesian coordinates) for the operation instructions within the entire range or within a certain range of an existing program.
- If the position data is joint coordinates, convert the coordinates considering the shift amount resulting from the tool change or user coordinate system change.
- Insert the shift results into a new or existing program.
- Execute the same shift on another program.

Position data conversion

The following rules apply to converted position data:

Rules for positions and attitudes:

- Position data with Cartesian coordinates is converted to Cartesian coordinates. Position data with joint coordinates is converted to joint coordinates.
- If converted joint coordinate position data falls outside the variable axis area, it is stored as unspecified. Converted Cartesian coordinate position data is stored as is even if it falls outside the variable axis area.
- Position data in the position registers is not converted.
- Position data with joint coordinates for operation instructions involving incremental instructions is stored as unspecified.

Rules for the configuration (joint placement and turn number) of position data with Cartesian coordinates:

- The configuration is not changed due to conversion.
- For the turn number, if the conversion causes rotation about the wrist axis by 180° or more, the turn number for the axis is optimized, and a message appears so that the user can decide whether to accept it.

For the tool change shift functions, select the desired position data conversion method from the following:

- TCP fixed: The original position of the tool end point is preserved in the converted data.

  For example, TCP fixed is useful if the previously used hand was damaged and replaced by a new one. By setting the tool coordinate system number of the old hand for Old UTOOL number and the tool coordinate system number of the new hand for New UTOOL number and using a tool change shift function with TCP fixed, the TCP of the new tool is moved to the original specified point correctly.
- Robot fixed: The original attitude of the robot (joint positions) is preserved in the converted data.

  For example, Robot fixed is useful if the program was taught in a tool coordinate system different from that used by the actually mounted hand and the correct tool coordinates are set later. By setting the tool number used when the program was taught for Old UTOOL number and the correct tool coordinate system number for New UTOOL number, and using a tool change shift function with Robot fixed, the program can operate in the correct tool coordinate system, with the same positions as the originals.

The coordinate change shift functions allow the user to specify whether to convert position data.

- Perform conversion: Position data is converted so that the TCP is located at the same position.
- Do not perform conversion: Position data is not converted even if the coordinate system number is changed.

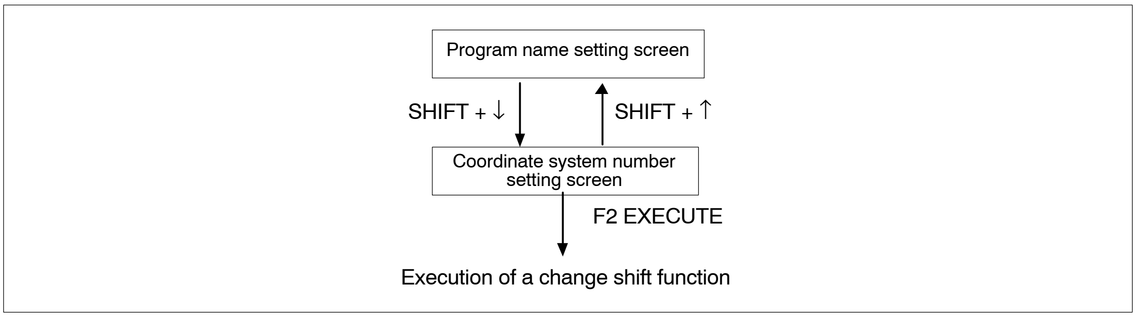
Types of coordinate system change shift functions

The following coordinate system change shift functions are supported:

- Tool change shift function: Changes the tool coordinate system number in the position data.
- Coordinate change shift function: Changes the user coordinate system number in the position data.

To execute the coordinate system change shift functions, use the change shift screen TOOL OFFSET (UFRAME OFFSET). The figure below shows how to navigate through the change shift screen.

Figure 9–13. Coordinate System Shift Screen



Procedure 9–8 Executing the tool change shift function

Condition ■ The program on which the shift is to be performed exists.

```
TEST1                                JOINT  30%
                                            1/6
   1:J  P[1]    100% FINE
   2:J  P[2]    70% CNT50
   3:L  P[3]    1000cm/min CNT30
   4:L  P[4]    500m/sec FINE
   5:J  P[1]    100% FINE
[End]

POINT                                TOUCHUP >
```

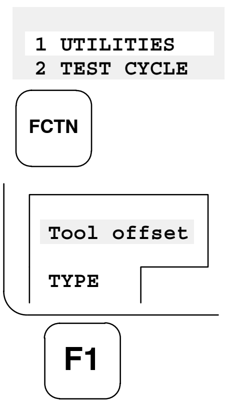
Step

1 Press MENUS. The screen menu appears.

2 Select 1 UTILITIES.

3 Press F1 TYPE. The screen switching menu appears.

4 Select Tool offset. The program name input screen appears.




```
TOOL OFFSET                          JOINT  10%
Program                                     1/6
  1  Original Program:          [Test1     ]
  2  Range:                        WHOLE
  3  Start line: (not used)        ***
  4  End line:   (not used)        ***
  5  New Program:               [Test2     ]
  6  Insert line:(not used)        ***


Use shifted up, down arrows for next page
[TYPE]                                      >
CLEAR                                       >
```

5 Specify the necessary items.

6 After specifying the items, go to the next screen with SHIFT + ↓. The coordinate system number setting screen appears. To return to the previous screen, use SHIFT + ↑.

SHIFT ⇩

```
TOOL OFFSET                       JOINT  10%
UTOOL number                             1/6

  1   Old UTOOL number                  1
  2   New UTOOL number                  2
  3   Convert type               TCP fixed




[TYPE]    EXECUTE                          >
```

```
CLEAR                                      >
```

7 Enter the current and new tool coordinate system numbers.
To set F as the new tool coordinate system number, enter 15.

8 Press F2 EXECUTE to execute the shift.

```
[TYPE]   EXECUTE
```

F2

9 If the turn number is changed as a result of the conversion, the user is notified and prompted to make a selection.

```
Select P[3]:J5 angle   (183°)
183°    -177°    uninit            QUIT  >
```

10 F1 indicates the axial angle associated with the optimized turn number.
F2 indicates the axial angle associated with the original turn number.
F3 uninit causes the data to become unspecified data.
F5 QUIT interrupts the conversion.

11 To erase all the shift information, press F–>”>” and press F1 1 CLEAR on the next page.

```
CLEAR
```

F1

> **CAUTION**
> When the tool change shift function is performed, the tool coordinate system number selected by the system is changed to the new tool number.

Procedure 9–9 Executing the coordinate change shift function

Condition ■ The program on which the shift is to be performed exists.

```
TEST1                                  JOINT   30%
                                               1/6
   1:J  P[1]    100% FINE
   2:J  P[2]    70% CNT50
   3:L  P[3]    1000cm/min CNT30
   4:L  P[4]    500m/sec FINE
   5:J  P[1]    100% FINE
[End]

POINT                                  TOUCHUP  >
```

Step

1 Press MENUS. The screen menu appears.

2 Select 1, UTILITIES.

3 Press F1, [TYPE]. The screen switching menu appears.

4 Select Frame offset. The program name input screen appears.

```
1 UTILITIES
2 TEST CYCLE
```

FCTN

```
Frame offset

TYPE
```

F1

```
UFRAME OFFSET                          JOINT   10%
Program                                        1/6
  1   Original Program:          [Test1      ]
  2   Range:                        WHOLE
  3   Start line: (not used)        ***
  4   Ene line:   (not used)        ***
  5   New Program:               [Test2      ]
  6   Insert line:(not used)        ***


Use shifted up, down arrows for next page
[TYPE]                                          >

CLEAR                                           >
```

5 Specify the necessary items.

6 After specifying the items, go to the next screen with SHIFT + ↓. The coordinate system number setting screen appears. To return to the previous screen, use SHIFT + ↑.

SHIFT ⇩

```
UFRAME OFFSET                          JOINT   10%
UFRAME number                                  1/3

  1   Old UFRAME number:                  1
  2   New UFRAME number:                  2
  3   Convert Position data (Y/N):     YES




 [TYPE]    EXECUTE                              >

CLEAR                                           >
```

7 Enter the current and new user coordinate system numbers.
To set F as the new user coordinate system number, enter 15.

8 Press F2, EXECUTE to execute the shift.

```
[TYPE]   EXECUTE
```

F2

9 If the turn number is changed as a result of the conversion, the user is notified and prompted to make a selection.

```
Select P[3]:J5 angle   (183°)
183°    -177°    uninit           QUIT  >
```

10 F1 indicates the axial angle associated with the optimized turn number.
F2 indicates the axial angle associated with the original turn number.
F3 uninit causes the data to become unspecified data.
F5 QUIT interrupts the conversion.

11 To erase all the shift information, press F–>">" and then press F1 CLEAR on the next page.

```
CLEAR
```

F1

CAUTION

When the coordinate change shift function is executed, the user coordinate system number selected by the system is changed to the specified new user coordinate system number.

9.4 Soft Float Function

Usually, the robot moves accurately toward the goal specified using the teach pendant (taught point).

When the robot is used to mount workpieces on a machine tool, variances in workpiece precision may result in a shift in the workpiece position relative to the tool, thus possibly causing interference between the workpiece and tool.

A soft float function has been added which is effective in mounting workpieces with variances in precision onto a machine tool.

The soft float function is also very effective if the synchronization speed is unstable as in the extraction of workpieces in sync with hydraulic extrusion, and if workpieces that the robot cannot grip accurately, such as rough–machined workpieces, are to be handled.

Function

The soft float function works as follows:

- Two types of soft float are supported: joint soft float for specifying the softness related to the direction of rotation of each arm of the robot, and Cartesian soft float for specifying the softnesses on the Cartesian axes.
- The function is enabled/disabled using an instruction in the program. Its conditions are also specified using the instruction.
- ”Servo flexibility” can be specified for each axis. The term servo flexibility indicates how strongly the axis resists external forces. It is specified between 0% and 100%. A servo flexibility of 100% corresponds to being the most flexible. The servo flexibility is specified using a condition table that contains a set of data for one group (for nine axes).
- If an external force above a certain level (so high as to overcome a static frictional force) is applied to a robot, the axis of the robot is pressed and moved.
- An external force applied to a robot may prevent it from reaching the taught point. The distance between the taught point and the point the robot can reach is nearly proportional to the magnitude of the external force.
- If static load is applied to a robot, the robot controls force to maintain its attitude even if the soft float function is enabled.

The detailed descriptions of the soft float function follow.

Program instruction

The following three program instructions related to the soft float function are supported.

SOFTFLOAT[n]

The soft float function is enabled using condition n.
* The setting of soft float condition is explained in ”Condition setting menu”.

SOFTFLOAT END

The soft float function is disabled.

FOLLOW UP

When an external force is removed from a robot, it usually tries to go back to the taught point. However, this instruction causes the robot to assume that the current position is the taught point, and prevents it from going back to the taught point.

Soft float function effective range

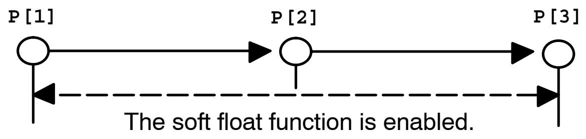
The SOFTFLOAT[n] instruction can be used in two modes; in one mode it is used solely in a program and in the other mode it is used as an auxiliary motion instruction after a motion statement.The range in which the soft float function is effective for robot operation is determined according in to which mode this instruction is used.

Sole instruction

The soft float function is enabled after the end of the motion specified on the line preceding the solely specified SOFTFLOAT[n] instruction.

In the following example, the soft float function is enabled after the motion specified on line 1 ends, and disabled by SOFTFLOAT END on line 5.

```
1: J P[1] 100% FINE
2:  SOFTFLOAT[1]
3: L P[2] 100mm/sec FINE
4: L P[3] 100mm/sec FINE
5:  SOFTFLOAT END
```



Auxiliary motion instruction

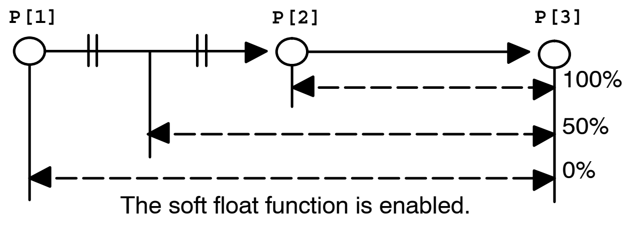
The soft float function becomes enabled during execution of a motion statement attached with a SOFTFLOAT [n] instruction.

The point at which the soft float function becomes enabled is determined by a soft float condition ”Exec Start Ratio.”

Auxiliary motion instruction is specified as the ratio (from 0% to 100% in 1% steps) of a distance to be traveled before the robot reaches the taught point corresponding to a motion statement attached with a SOFTFLOAT[n].

In the following example, the soft float function is effective between P[1] taught using a motion statement on line 1 and P[2] taught using a motion statement on line 2 attached with the SOFTFLOAT[n] instruction.

```
1: J P[1] 100% FINE
2: L P[2] 100mm/sec FINE SOFTFLOAT[1]
3: L P[3] 100mm/sec FINE
4: SOFTFLOAT END
```



NOTE The auxiliary motion instruction is not supported by Cartesian soft float.

Condition setting menu

The soft float conditions are specified on the [SETUP.Softfloat] menu, which consists of the following two menus.

- List menu
- Detail menu

A function key is used to select either menu.

– Pressing the F3 (DETAIL) key on the list menu selects the detail menu.

– Pressing the F3 (LIST) key on the detail menu selects the list menu.

Up to 10 conditions can usually be specified for the soft float function.

List menu

```
SETUP/SOFTFLOAT                JOINT 30%
Group 1
 No     Start (%)        Comment
  1          0      [                ]
  2          0      [                ]
  3          0      [                ]
  4          0      [                ]
  5          0      [                ]
  6          0      [                ]
  7          0      [                ]
  8          0      [                ]
  9          0      [                ]
 10          0      [                ]

[ TYPE ] GROUP  DETAIL
```

F3, DETAIL

F3, LIST

Joint soft float details screen

```
SOFTFLOAT (JOINT)              JOINT 30%

    Group 1
  1 Schedule No[   1]:[****************]
  2 Exec Start Ratio :   0 %
  3 Axis1 Soft Ratio :   0 %   DISABLE
  4 Axis2 Soft Ratio :   0 %   DISABLE
  5 Axis3 Soft Ratio :   0 %   DISABLE
  6 Axis4 Soft Ratio :   0 %   DISABLE
  7 Axis5 Soft Ratio :   0 %   DISABLE
  8 Axis6 Soft Ratio :   0 %   DISABLE
  9 Axis7 Soft Ratio :   0 %   DISABLE
 10 Axis8 Soft Ratio :   0 %   DISABLE
 11 Axis9 Soft Ratio :   0 %   DISABLE

 [TYPE]   NUMBER   LIST   JOINT   CART  >
          GROUP    LIST                 >
```

F5, CART

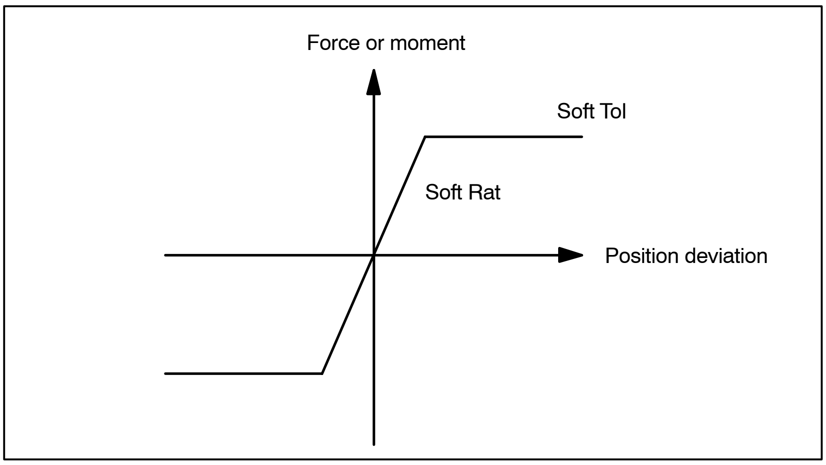
F4, JOINT

Cartesian soft float screen

```
SOFTFLOAT (CARTES1A)                    JOINT

    Group 1
  1 Schedule No[   1]:[                 ]
  2 Enable/Disable:[DISABLE]
  3 Coordinate:    [WORLD]
                     Soft Rat   Soft Tol
  4 X direction      [0]%       [0]%
  5 Y direction      [0]%       [0]%
  6 Z direction      [0]%       [0]%
  7 X rotation       [0]%       [0]%
  8 Y rotation       [0]%       [0]%
  9 Z rotation       [0]%       [0]%

 [TYPE]   NUMBER   LIST   JOINT   CART  >
          GROUP    LIST                 >
```

The following data can be specified on the detail menu.

Table 9–5. Setting items of Soft float detail menu

ITEMS	DESCRIPTIONS
Comment	Soft float condition number. By default, ten numbers can be set. Pressing the input key with the cursor on line 1 enables entering a comment. The comment text can be specified in the same way as on other menus.
Soft float start ratio	Line 2 specifies the point where the soft float function is enabled if the SOFTFLOAT [n] is used as an auxiliary motion instruction. See ”Soft float function effective range” for the soft float start ratio.
Servo flexibility	Servo flexibility for each axis can be specified on line 3 and the subsequent lines. The servo flexibility indicates how strongly the axis resist external forces. It is specified between 0% and 100%. A flexibility of 100% corresponds to being the most flexible. Whether the soft float function is enabled/disabled can be specified for each axis on line 3 and the subsequent lines. Setting the cursor at the rightmost end (enabled/disabled setting position) of each line causes the F4 (ENABLE) and F5 (DISABLE) keys to appear. Use these keys to specify whether to enable/disable the soft float function. **NOTE** Pressing the F2 (NUMBER) key selects another page of the detail menu for other conditions. `10 Axis8 Soft Ratio :   0 %   DISABLE` `11 Axis9 Soft Ratio :   0 %   DISABLE` `[ TYPE ] NUMBER   LIST                >` ↓↑ `10 Axis8 Soft Ratio :   0 %   DISABLE` `11 Axis9 Soft Ratio :   0 %   DISABLE` `[ TYPE ] NUMBER   LIST  ENABLE  DISABLE>`
Enable/Disable	When this item is set to DISABLE, soft float cannot be executed.
Coordinate	Select one of WORLD, USER, and TOOL. **NOTE** If the remote TP is used, USER indicates the coordinate system on the remote TCP.
X direction	Set the softnesses on or around the X–, Y–, and Z–axes. If Soft Rat increases, the spring constant decreases, allowing the robot to move with less force. If Soft Tol increases, the maximum force and moment applied by the robot in that direction decreases, allowing the robot to move with less force. The difference between Soft Rat and Soft Tol is illustrated below. 

Cautions/restrictions

When using the soft float function, observe the following cautions/restrictions.

- Restrictions imposed when the soft float function is enabled
  - It is not guaranteed that the robot always follows the taught path.
  - The taught route changes according to override.
  - The required operation time may be prolonged compared with normal operation.
- The soft float function is disabled automatically when:
  - Program execution starts.
  - Program execution ends.
  - The program stops due to an alarm that turns off the servo.
  - Jog feed is performed with the program at pause
  - The program is restarted after the cursor is moved manually with the program at pause.
  - Backward execution is performed.
  - Power is applied
- If the program is caused to pause, then restarted, the states of the soft float function (such as enabled/disabled and the soft float start ratio) are set to the conditions which exist before the program is caused to pause, except for the cases in the above operation, where the soft float function is disabled.
- The soft float function cannot be enabled by any method other than the SOFTFLOAT instruction.
- When the soft float function is enabled, the robot moves in the CNT 0 mode (no position check is made) even if FINE has been specified as motion statement positioning mode.
- When the soft float function is enabled, if an external force causes the robot to move beyond a certain distance, the following servo alarms occur.
  - If the robot is at rest : [SRVO–023 Stop error excess(G:i A:j)]
  - If the robot is operating : [SRVO–024 Move error excess(G:i A:j)]
- If an attempt is made to enable the soft float function with a brake applied, the brake is released automatically before the function is enabled.
- When the soft float function is enabled, brake control is ineffective.
- If the motion group mask in a program is [*,*,*,*,*] (there is no motion group), when the program issues instructions with the soft float function, the following alarm occurs:
  [INTP–216 (program name, line number) Invalid value for group number]
- The range of motion with the soft float function enabled should be minimized. A weight balance may vary depending on the soft float ratio and travel distance, thus shifting the vertical axis upward or downward. The range of motion with an auxiliary motion instruction issued should also be minimized for the same reason. In addition, the speed of motion should be kept low.
- When the soft float function is enabled, if follow–up processing requires more time than specified in system variable $SFLT_FUPTIM, the servo alarm or program pause alarm occurs. System variable $SFLT_ERRTP specifies which alarm to occur.

  $SFLT_FUPTIM Default value: 1000 (ms)
  This value varies from one system to another. The large value that does not cause an alarm during normal operation should be used.

  $SFLT_ERRTYP Default value: 0
  - If 0, servo alarm ”SRVO–111 Softfloat time out” occurs.
  - If 1, Program pause alarm ”SRVO–112 Softfloat time out” occurs. (The alarm number is different between the alarms.)

  The default value should be used unless turning the servo off invites as an alarm any inconvenience in the system.
- When the soft float function is enabled, follow–up processing is normally performed for individual motion instructions.
  This processing is enabled or disabled according to system variable $SFLT_DISFUP.

  $SFLT_DISFUP Default value: FALSE
  - If FALSE, follow–up is performed at the start of each motion instruction in the program.
  - If TRUE, follow–up is not performed for individual motion instructions in the program.
- This function cannot be used with arc tools.

NOTE Follow–up
With the soft float function, external forces are applied to the robot so that it operates at positions slightly different from those specified. When the external force is removed after the completion of the operation, the robot usually attempts to move back to a specified point abruptly. Follow–up prevents this abrupt movement.

9.5 Continuous Rotation Function

The continuous rotation function allows continuous and limitless rotation about the final axis or an additional rotation axis of the robot in one direction.

NOTE For example, the ”final axis” refers to the J6 axis of a robot having six axes.

For example, this function is useful for rotating those devices that require continuous rotation, such as conveyers, pumps, and grinders, about a robot axis or additional rotation axis.

To specify the items for this function, such as disable/enable, use the SETUP Continuous T screen (new). The start and stop of continuous rotation are directed from a program.

Before this function can be used, the setup necessary for continuous rotation must be performed.

Only a single continuous rotation axis can be allocated for each operation group. The axis must satisfy the following conditions:

- Final axis of the robot
- Final axis of the built–in additional rotation axes
- Any of the normal additional rotation axes
- Final axis of the independent additional axes

The continuous rotation axis must satisfy the following mechanical conditions:

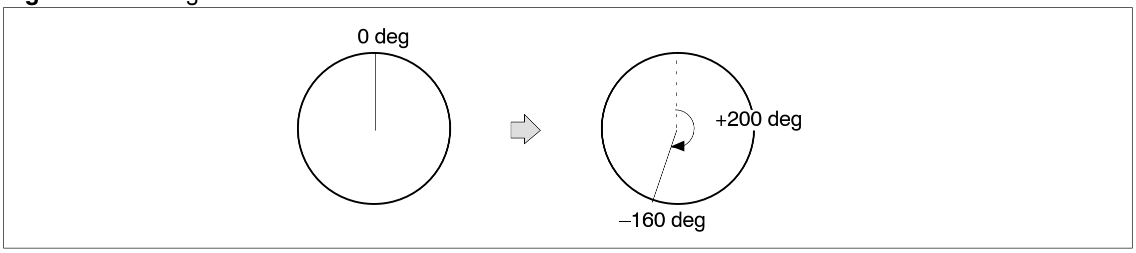
- The mechanism must allow continuous operation (must be free of obstacles such as stoppers).
- The gear reduction ratio (value of Numerator of Gear Ratio/Denominator of Gear Ratio on the setting screen, the speed of the motor required for one rotation about the axis) must be 4000 or less.

To use this function, an option (continuous rotation function) is required.

Function

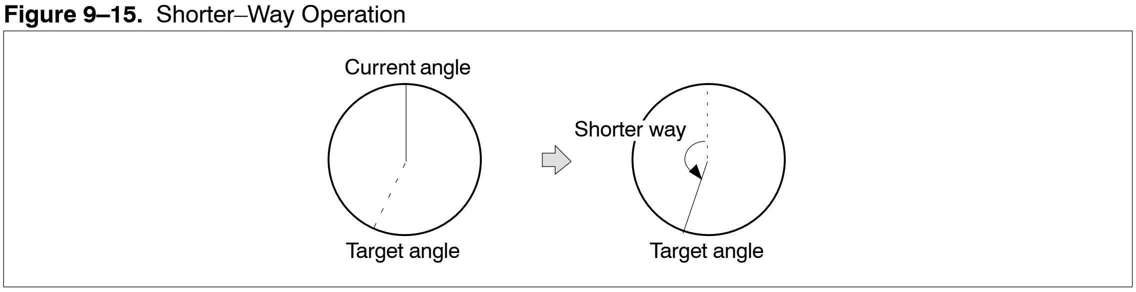
When this function is enabled, the axis allocated as a continuous rotation axis allows limitless rotation. The angle on the axis is, therefore, represented by a relative degree within +180°, not by an absolute one. For example, the figure below shows rotation from 0° to 200° in the positive direction. The angle on the axis after the rotation is –160°, not 200°.

Figure 9–14. Angle on the Continuous Rotation Axis



When this function is enabled but continuous rotation is not performed (see the next page for an explanation of how to use continuous rotation), rotation is performed about the continuous rotation axis from the current angle to the target angle in whichever direction incurs the least amount of motion. (Usually, the direction of rotation about the axis is determined with the relationship between the current and target angles.) This ”shorter–way operation” is effective in reducing the cycle time.

Figure 9–15. Shorter–Way Operation



Setup

To use the function,

- Perform setup on the SETUP Continuous T screen and
- Specify the start/stop of continuous rotation with the operation add instruction, ”continuous rotation speed instruction.”

Procedure 9–10 Setting up the continuous rotation function

Step

1 Press MENUS. The screen menu appears.

2 Select SETUP.

3 Press F1, [TYPE]. The screen switching menu appears.

4 Select Cont Turn. The continuous rotation setup screen appears.

```
5 I/O
6 SETUP
7 FILE
```

MENUS

```
Cont Turn

TYPE
```

F1

```
SETUP Continuous T            JOINT  10%

  1   Group:1
  2   Continuous Turn Axis Num    :   0
  3   Numerator of Gear Ratio     :   0
  4   Denominator of Gear Ratio   :   0

[TYPE]                      DONE
```

5 Specify the necessary items using the numeric and other keys.

- To disable the continuous rotation function, set ”0” for Continuous Turn Axis Num.
- The maximum value for Numerator of Gear Ratio and Denominator of Gear Ratio is 32766.
- Set the operation group number for Group. If a different number (number of the operation group to be viewed) is entered in this field, the other settings are changed to those of the operation group.

6 After specifying the items, press F4 DONE. The following message appears.

```
DONE
```

F4

7 Turn off the power, then turn it back on with a cold start.
The items on the continuous rotation setup screen are described below.

Table 9–6. Contents of the Continuous Rotation Setup Screen

Item	Description
Group	Set the operation group number.
Continuous Turn Axis Num	Set the number of the continuous rotation axis. If ”0” is set, this function is disabled for the operation group.
Numerator of Gear Ratio Denominator of Gear Ratio	Set the gear reduction ratio for the continuous rotation axis set for the above item. A value from 0 to 32766 can be set for each item. The items must, however, satisfy the following: Numerator of Gear Ratio ÷ Denominator of Gear Ratio ≦ 4000

Using the function

After setting up the continuous rotation axis, specify the start point of continuous rotation using the operation add instruction, ”continuous rotation speed instruction.”

The following ”continuous rotation speed instruction” is supported. The ”continuous rotation speed instruction” must be specified as an operation add instruction.

* The specification method is the same as that for other operation add instructions, and is therefore omitted. (→ Section 5.3.4, ”Specifying operation add instructions.”)

- Continuous rotation speed instruction CTV
  * where i = –100 to 100, which is the ratio of the rotation axis speed to the maximum axis speed (%)

Starting continuous rotation

Continuous rotation is started as soon as an operation statement with a continuous rotation speed instruction added is started.

Stopping continuous rotation

Continuous rotation is stopped when the first operation statement with no continuous rotation speed instruction added is started since a continuous rotation speed instruction was started.
When continuous rotation is stopped, the operation on the other axes for the same operation group also terminates. The robot, therefore, decelerates even if the positioning format for the previous operation is CNT.
The robot starts decelerating to stop on the continuous rotation axis after it has completely stopped on the other axes. At this time, the robot is not necessarily at the specified position on the continuous operation axis. Thus, the synchronization of the operation on the continuous rotation axis with the operation on the other axes (including those for other operation groups) is lost.
If an operation statement is specified next, the robot rotates in the same direction as the previous continuous rotation direction to move to the specified position.

Notes

- Continuous rotation continues even if logic instructions (instructions other than those in operation statements) are executed.
- During program playback, the turn number for the continuous rotation axis is ignored, and is always assumed to be "0."
- The turn number for the continuous rotation axis at a point specified when this function is enabled is always stored as "0."
- If the rotation axis speed for a continuous rotation speed instruction is specified as "0," continuous operation is not performed. If an operation statement is specified next, shorter–way operation is performed on the continuous rotation axis. This feature is useful if continuous rotation about the continuous rotation axis is to be stopped temporarily but temporary stop of the robot due to the end of the continuous rotation is to be avoided. (See the next section, "Example of use.")
- In single–step execution (both forward and backward), continuous rotation is not performed even if a continuous rotation speed instruction is added; shorter–way operation is performed.
- Continuous rotation stops due to a hold. If program execution is subsequently restarted, if the target position has already been reached on axes other than the continuous rotation axis, continuous rotation is not performed. If the target position has not been reached on axes other than the continuous rotation axis, continuous rotation is restarted.
- Continuous rotation about the continuous rotation axis is possible from jog feed.

Example of use

The following shows an example of using the continuous rotation speed instruction.

```
1:J  P[1]   100% FINE
2:J  P[2]   100% CNT100 CTV100
3:J  P[3]   100% FINE
4:J  P[4]   100% CNT100 CTV100
5:J  P[5]   100% FINE CTV100
6:J  P[6]   100% FINE
7:J  P[7]   100% FINE CTV100
8:WAIT 100.0sec
9:J  P[8]   100% FINE
```

- Description of lines 1 to 3:
  During operation from P[1] to P[2], continuous operation is performed. Although the positioning format specified on line 2 is "Smooth," the robot decelerates (stops temporarily on all axes at the start of the operation on line 3) because a continuous rotation speed instruction is not added to the next line, line 3.
- Description of lines 4 to 6:
  Continuous rotation starts as soon as the execution of line 4 starts. Because the rotation axis speed specified with the continuous rotation speed instruction on line 5 is 0, continuous rotation stops temporarily at the start of the execution of line 5. Because continuous rotation continues, the positioning format CNT100 on line 4 is valid and the robot does not decelerate.
  When line 6 is executed, shorter–way operation is performed on the continuous rotation axis.

- Description of lines 7 to 9:
  Continuous rotation starts at the start of operation on line 7. Continuous rotation continues during the execution of the wait instruction (logic instruction) on line 8.
  The robot stops temporarily on all axes at the start of operation on line 9, and continuous rotation stops.

Notes/restrictions

Note the following when using this function:

- When continuous rotation is to be performed on a robot axis or built–in additional axis, The X and Y components of the tool coordinate system must both be 0. (Only the Z–axis component can have a value other than 0.)
  If this condition is not satisfied, the path of linear or arc operation cannot be guaranteed in normal operation other than continuous rotation.
- This function cannot be used together with the following functions:
  - Asynchronous addition axis speed instruction. (The synchronous additional axis speed instruction can be used.)
  - Arc sensor
  - Weaving
  - TCP speed estimation function (sealing flow rate control)
- This function automatically updates the mastering data (for the continuous rotation axis only) according to the amount of rotation about the continuous rotation axis. Thus, previously recorded mastering data may not match the current mastering data.
  After this function is disabled, it is not necessary to perform mastering.
- When this function is disabled, the current position on the continuous rotation axis may fall outside the stroke limits. If this occurs, move the position on continuous rotation axis within the stroke limits using jog feed or a program.
- If, on a multigroup system, the settings on the SETUP Continuous T screen are changed and the F4 DONE key is pressed, it is necessary to set system variable $PARAM_GROUP[group].$SV_OFF_ENBL[i] (where i is an axis number) to FALSE to disable break control for all the axes for all operation groups before turning the power back on with a cold start.
- On a multigroup system, even if there are multiple continuous rotation axes, separate continuous rotation speeds cannot be specified for them.
- At the end of continuous rotation, one or more rotations about the continuous rotation may be performed to ensure smooth deceleration and stop. (The amount of rotation differs depending on the acceleration/deceleration constant.)
- Even during backward execution (single–step execution), shorter–way operation is performed on the continuous rotation axis. If, therefore, forward step execution and backward execution are performed sequentially in an operation statement with the movement angle being very close to 180o, rotation may be performed about the continuous rotation axis in the same direction during the forward and backward executions.

9.6 Position Register Look–Ahead Execution Function

While the robot is executing a program, it reads the lines ahead of the line currently being executed (look–ahead execution).

Conventionally, look–ahead execution was performed for motion statements having normal position data (not using position registers). Look–ahead execution could not be performed for motion statements that used position registers for their position data.

Motion statements using position registers could not be read in advance because the values in the position registers could be changed by the program, data transfer function, and so forth.

* If the robot reads a motion statement using a position register prior to its execution, the value of the position register may yet be changed by a program or another function (such as data transfer). Such a change is not reflected in the motion statement that has already been read by the robot. Consequently, the robot's operation may be unpredictable.

Motion statements that use position registers can be classified into two types:

- Motion statements with the target position specified by a position register
- Motion statements with an offset instruction where an offset is given by a position register

Even when a target position or offset is calculated during program execution, and a position register holding this calculation result is used with a motion statement, look–ahead execution was not performed for the statement, for the reason explained above.

The position register look–ahead execution function enables look–ahead execution for position registers. For this purpose, an instruction to lock position registers and an instruction to unlock the registers are newly provided. By means of these instructions, the user can explicitly specify a program portion. Then, for the specified program portion, even when it contains motion statements that use position registers, look–ahead execution can be performed.

Function

The position registers can be locked to prevent their contents from being changed after they are read. When an attempt is made to execute an instruction to change a locked position register (for example, an assign instruction for the position register, or an application instruction to set data in the position register), the following alarm message is issued:
[INTP–128 Pos reg is locked]

When a function (such as the data transfer function) other than the program attempts to change the value of a locked position register, the following alarm message is issued, and the attempt fails:
[VARS–053 Pos reg is locked]

Position registers are generally locked and unlocked with instructions taught in a program. When a program that has locked the position registers terminates, the position registers are unlocked automatically.

All position registers are locked simultaneously. While the position registers are locked, access to any position register is disabled, even in a different motion group.

NOTE Before using position register instructions, lock position registers. When position register instructions are used with the position registers unlocked, operation may become tight.

Operation

The following program instructions have been added:

LOCK PREG

Locks all position registers. This instruction prevents any change being made to any position register.

UNLOCK PREG

Unlocks the position registers.

These are control instructions (not motion instructions). They can be taught in the same way as other control instructions (See Section 5.3.5, "Teaching a control instruction").

Example

The following shows how to use the LOCK PREG and UNLOCK PREG instructions in a program:

```
 1: J P[1] 100% FINE
 2:    PR[1]=PR[2]
 3:    PR[2]=PR[3]
 4: LOCK PREG
 5: L P[2] 100mm/sec Cnt100
 6: L P[3] 100mm/sec Cnt100
 7: L PR[1] 100mm/sec Cnt100
 8: L P[4] 100mm/sec Cnt100 offset, PR[2]
 9: L P[5] 100mm/sec FINE
10: UNLOCK PREG
```

When line 4 of this sample program has been executed, the position registers are locked. They are unlocked when line 10 has been executed. Therefore, the motion statements with position registers in lines 7 and 8, which are executed with the position registers locked, are subject to look–ahead execution.

If the program is terminated between lines 4 and 10, the locked position registers are unlocked automatically.

If the program is paused between lines 4 and 10, the cursor is moved manually, then the program is restarted, the locked position registers are unlocked. In this case, look–ahead execution is not performed for the statements in lines 7 and 8.

NOTE When back execution is performed, then normal execution is restarted, the position registers are unlocked. For example, suppose that program execution is paused during the execution of line 6, back program execution is performed up to line 5, then forward program execution is restarted. In this case, the position registers are unlocked. So, look–ahead execution is not performed for lines 7 and 8.

When program execution is started from a line located after line 4, the position registers are not locked. So, look–ahead execution is not performed for lines 7 and 8.

A LOCK PREG instruction can be executed even when the position registers are already locked. (Nothing occurs, however, when the LOCK PREG instruction is executed for a second time.) Similarly, the UNLOCK PREG instruction can be executed even when the position registers are not locked. (Nothing occurs, however, when the UNLOCK PREG instruction is executed for a second time.)

Notes

Note the following when using this function:

- The LOCK PREG and UNLOCK PREG instructions are not executed in backward program execution mode.
- Look–ahead execution is not performed for the LOCK PREG and UNLOCK PREG instructions. This means that when one of these instructions is encountered, look–ahead execution is stopped temporarily; after the instruction is executed, look–ahead execution is again enabled.

9.7 Operation Group DO Output Function

The operation group DO output function outputs information about the operation groups that are capable of jog feed, and about the operation groups of the programs being executed/temporarily stopped, to an external device with a digital output signal (SDO) or robot output signal (RD0). This allows devices other than the teach pendant to recognize the currently effective operation groups, thus improving safety. This function is effective when the multigroup option is used.

Function

This function allows the allocation of two DOs (jog signal and program signal) to a single operation group. For DOs, any digital output signals or robot output signals of the robot can be used.

Each allocated DO signal turns on/off under the following conditions:

Jog signals

When the teach pendant is disabled, all signals turn off.
When the teach pendant is enabled, the signal for the currently selected operation group on the teach pendant turns on, while the other signals turn off.

Program signals

Regardless of whether the teach pendant is enabled or disabled, the signal for the operation group of the program currently being executed/temporarily stopped turns on. (The signal does not turn on when the program is merely selected.)
If other programs are being executed/temporarily stopped with the multitask option, the signals for the operation groups of these programs also turn on.

Setup

To set up the operation group DO output function, use the [Set up operation group DO] screen.

To change the signal number for an operation group, move the cursor to the signal number and enter a new value.

```
Motion group DO                       JOINT  10%
GROUP NO.   PROGRAM   JOG

   1         RO[1]     RO[2]
   2         DO[3]     DO[3]
   3         RO[0]     RO[0]

[TYPE]                                 RO     DO
```

To change the type of a signal, position the cursor to the signal number and press function key F4 ”RDO” or F5 ”SDO.”

To disable a signal, set the number of the signal to ”0.”

The same signal can be set for both the program and jog signals for the same operation group. In this case, the output signal is the OR of the two signals. That is, the signal turns on if either the program or jog signal turns on. (The signal turns off only if both the program and jog signals turn off.)

Example of using this function with the multitask option

This section explains the operation of this function when a subprogram call or the multitask option is used.

The output program signal is the OR of the signals for all the operation groups of the program currently being executed or temporarily stopped.

If a program without an operation group calls a program having an operation group by using a subprogram call, the signal for the operation group of the subprogram turns on only while the subprogram is being executed. (The signal does not turn on when the main program without an operation group is merely selected/executed.)

If the execution instruction of the multitask function is to start another program that operates the robot (the main program that has the execution instruction does not have an operation group), the signal for the operation group of the program started by the execution instruction does not turn on when the main program is merely selected/executed. The program signal turns on when the program that operates the robot is actually started.

Consider the following three example programs:

```
PROGRAM MAIN : Operation group[*,*,*,*,*]
1:RUN PRG A
2:RUN PRG B
:

PROGRAM PRG A : Operation group[1,*,*,*,*]
1:J  P[1]   100% FINE
:

PROGRAM PRG B : Operation group[*,1,*,*,*]
1:L  P[1]   500mm/sec CNT100
:
```

Program MAIN, which does not have an operation group, starts PRG A and PRG B having operation groups by using execution instructions. PRG A uses operation group 1 and PRG B uses operation group 2.

- The program signals for the groups do not turn on when program MAIN is merely selected.
- When line 1 of MAIN is executed, PRG A is started and the signal for operation group 1 turns on.
- When line 2 of MAIN is executed, PRG B is started and the signal for operation group 2 turns on.
- When PRG A and PRG B terminates, the respective signals for operation groups 1 and 2 turn off.

Notes

Note the following when using this function:

- The same signal cannot be defined for different operation groups.
- While a program is being executed/temporarily stopped, the type (SDO or RDO) and number of the program signal cannot be changed.

9.8 Pre–Execution Instruction Function

This function calls a subprogram and output a signal before or after the specified time at which robot operation is to terminate.

For example, if a signal output instruction is specified in a subprogram, this function allows a signal to be output during robot operation. It can also eliminate the wait time associated with the transfer of data to and from peripheral devices, thus reducing the cycle time.

Function

This function allows a main program to call and execute a subprogram before or after the specified operation termination time. Signal output is executed before or after the specified operation termination time.

Using an instruction in a program, specify the time at which a subprogram is to be called (in seconds). (This specified time is called the execution start time.) The time at which operation terminates is assumed to be 0 seconds, which differs depending on the positioning type FINE CNT.

Using an instruction in a program, specify the name of the subprogram to be called.

The pre– (or post–) instruction is an operation add instruction. Both the subprogram name and execution start time must be specified with the operation add instruction.

Instruction statement

Specify the execution start time and subprogram after an operation statement.

Figure 9–16. Pre–Execution Instruction (Operation Add Instruction)

Operation statement — TIME BEFORE / TIME AFTER — execution–start–time CALL subprogram–name

TIME BEFORE	Executes the subprogram before operation termination.
TIME AFTER	Executes the subprogram after operation termination.

Example

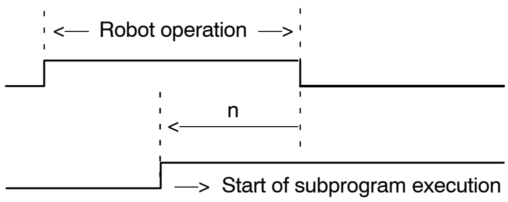
```
1:J  P[1]   100% FINE
:TIME BEFORE   1.0sec CALL OPEN HAND

1:J  P[1]   100% FINE
:TIME AFTER   1.0sec CALL OPEN HAND
```

Description of execution start time

According to the specified execution start time, the subprogram is executed at the following time:

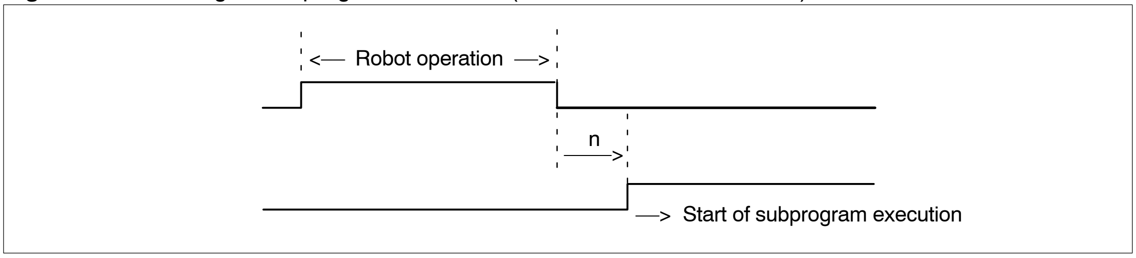
If execution start time, ”n” seconds, is specified with a pre–execution instruction, the subprogram is executed n seconds before operation termination.

Figure 9–17. Timing of Subprogram Execution (Pre–Execution Instruction)



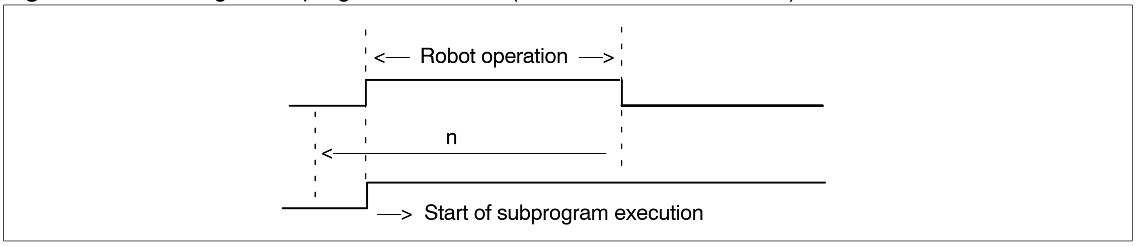
If execution start time, ”n” seconds, is specified with a post–execution instruction, the subprogram is executed n seconds after operation termination.

Figure 9–18. Timing of Subprogram Execution (Post–Execution Instruction)



If the execution start time specified with a pre–execution instruction exceeds the operation time, the subprogram is executed as soon as operation starts.

Figure 9–19. Timing of Subprogram Execution (Pre–Execution Instruction)



The execution start time that can be specified in a program is

- 0 to 30 seconds for a pre–execution instruction
- 0 to 0.5 seconds for a post–execution instruction

CAUTION

Even if the robot operation time is changed due to a change in the override, the time at which subprogram execution is to start depends on the execution start time. The execution start position of the subprogram is, therefore, changed due to a change in the override.

If 0 seconds is specified, the subprogram is executed at almost the same time as when the pre–execution instruction is not specified.

Search/replace functions

Search function

By selecting CALL program for search item CALL, the search function searches for the call instructions of pre–execution instructions.

Replace function

- By selecting replace item TIME BEFORE/AFTER, TIME BEFORE/AFTER replacement and execution start time replacement can be performed.
- By selecting CALL program of replace item CALL, the subprogram names for pre–execution instructions can be replaced.

Single step

When an operation statement with an execution start time adjustment instruction specified is executed in single–step mode, operation stops temporarily at the time when the subprogram is called. Subsequently, the rest of the operation is executed in sync with single–step execution of the subprogram.

Power failure handling

If power failure handling is enabled and the power is removed during subprogram execution, execution starts with the remaining instructions of the subprogram due to a restart after the power is turned on again. In this case, the subprogram is executed with the position the robot was located when the power was removed. Thus, the subprogram is executed with timing different from the usual timing. Great care must be taken regarding this point.

WARNING

Be sure you know the point where the robot is going to begin robot motion when power is restored after a power failure. Otherwise, you could injure personnel or damage equipment.

Procedure 9–11 Specifying the pre–execution instruction

Step

1 Position the cursor to the operation add instruction specification area (space following an operation instruction).

```
PNS0001                                   JOINT  10%
                                                 1/2
   1:J  P[1]    100% FINE
[End]




                                [CHOICE]
```

2 Press function key F4 CHOICE. A list of operation add instructions appears.

```
Motion Modify                             JOINT  10%
  1                   5
  2                   6
  3   TIME BEFORE     7
  4   TIME AFTER      8
PNS0001
                                                 1/2
  1:J  P[1]    100% FINE
[End]

                                [CHOICE]
```

3 Select item TIME BEFORE.

```
PNS0001                                   JOINT  10%
                                                 1/2




  1:J   P[1]    100% FINE
   :TIME BEFORE    sec ...
[End]

Enter value
                                [CHOICE]
```

4 Specify the time and press the Enter key. Example: 2 seconds.

```
TIME statement                            JOINT  10%
  1   CALL program     5  GO [ ]=
  2   CALL program[ ]  6  AO [ ]=
  3   SDO [ ]=         7
  4   RDO [ ]=         8
PNS0001
                                                 1/2
  1:J  P[1]    100% FINE
   :TIME BEFORE 2.0sec
[End]

Select item
                                [CHOICE]
```

5 Select item CALL program.
Select item CALL program to use AR.

```
PROGRAM list                         JOINT  10%
  1   Open hand      5
  2   Close hand     6
  3                  7
  4                  8
PNS0001
                                            1/2
  1:J  P[1]   100% FINE
   :TIME BEFORE 2.0sec, CALL
[End]

Select item
                         STRINGS
```

6 Select item Open hand.

```
PNS0001                              JOINT  10%
                                            1/2




  1:J  P[1]   100% FINE
   :TIME BEFORE 2.0sec, CALL Open hand
[End]


                          [CHOICE]
```

Program example

Main program: PNS0001

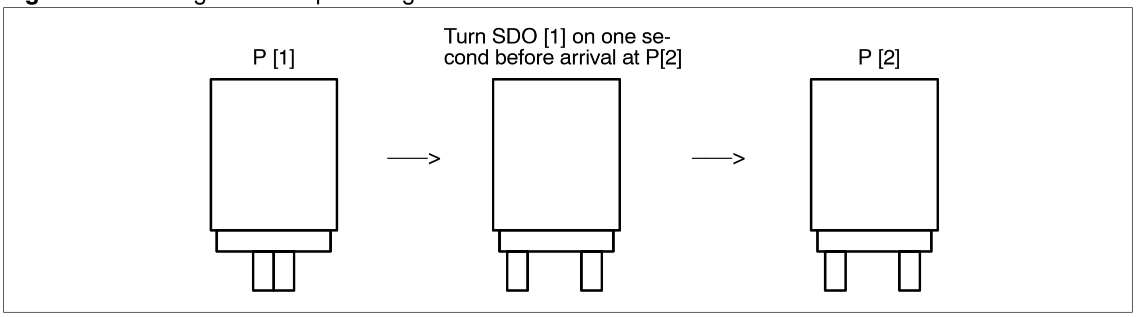
```
1:J  P[1]   100% FINE
2:J  P[2]   100% CNT100
 :TIME BEFORE 1.0sec CALL Open hand
3:CALL Close hand
```

Subprogram: Hand open

```
1:DO[1]=On
```

Operation performed when the main program is executed

Figure 9–20. Program Example Using a Pre–Execution Instruction



Notes/restrictions

In the subprogram specified for Call, operation statements cannot be specified. (The operation group in the subprogram must be [*, *, *, *, *].)

Until the execution of the called subprogram terminates, the next instruction in the main program is not executed.

No limit is imposed on the number of instructions that can be specified in a subprogram.

The TIME BEFORE/AFTER add instructions can be used together with other operation add instructions (except application instructions such as spot [] and skip instructions).

If the positioning specification for an operation statement is Smooth, the time of operation termination changes depending on the degree of Smooth. The time at which the subprogram is called changes accordingly. Depending on the situation, even if the execution time is set to 0 seconds with a pre–execution instruction, the subprogram may be executed too quickly. If this occurs, use a post–execution instruction.

If a pre–execution instruction is specified on the last line of a main program, the execution of the main program may terminate before the subprogram is called, in which case, the subprogram is not called. Do not, therefore, specify a pre–execution instruction on the last line of a program.
For direct specification of signal output, only SDO, RDO, GO, and AO are supported.

9.9 Distance before operations

9.9.1 Overview

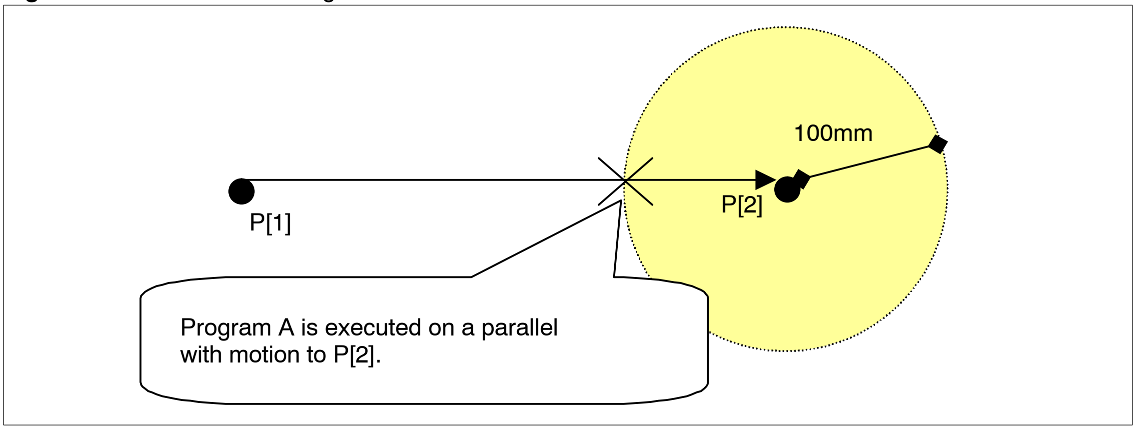
This function calls program or outputs signal when TCP is going into a region which is within specified distance from destination point. This program call and signal output is done on a parallel with main program execution.

Example

1 J P[1] 100% FINE

2 L P[2] 1000mm/sec FINE DB 100mm,CALL A

Figure 9.9.1 Execution timing of Distance Before



9.9.2 Specification

Item	Specification	Limitation
Distance value	0.0 to 999.9[mm]	Distance value and actual execution timing is different. The error depends on speed of TCP.
Trigger condition (*1)	TCP goes into a region, which is within specified distance from destination point. Please refer to Chapter 4 for details.	Distance value and actual execution timing is different. The error depends on speed of TCP.
Available instructions	● Signal output (ex. DO[1] = ON) ● CALL program	Program to be called cannot use motion group. Only logic instruction is available.

NOTE (*1) This is condition to precess instruction part.

9.9.3 Configuration

Before using Distance Before, set following system variable.

$SCR_GRP[1].$M_POS_ENB = TRUE

9.9.4 Instruction

1 Format

Distance Before is taught in following format.

`Motion statement + DB distance value, instruction part`

Example

`L P[2] 1000mm/sec FINE DB 100mm, CALL A`

Instruction part (Please referr to 3.)

distance value (Please refer to 2.)

> **CAUTION**
> Distance Before is a motion option. You cannot use DB as a standard instruction.

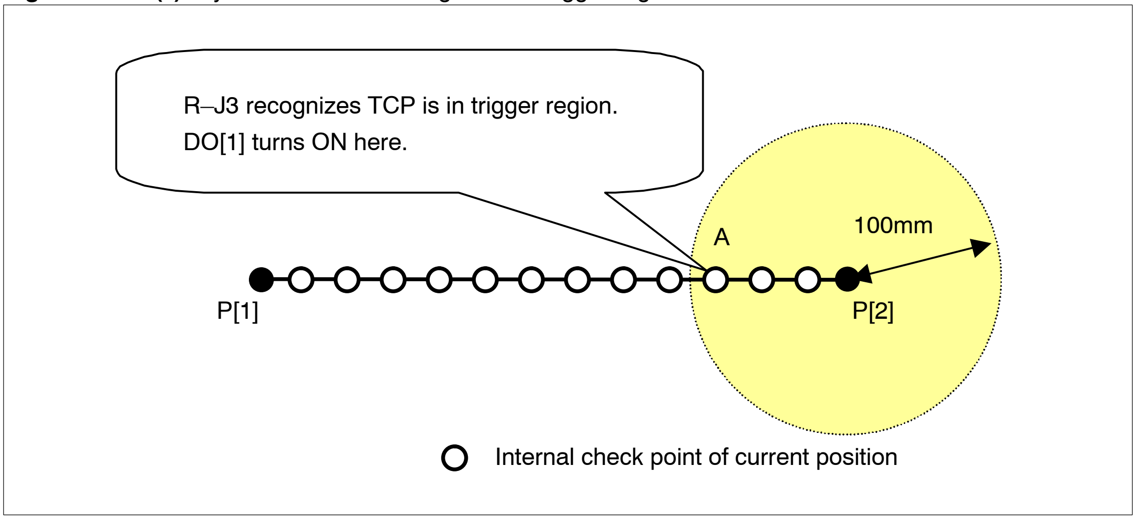
2 Distance value

(i) Distance value

Distance Before executes instruction part when TCP goes into a spherical region whose center is destination point. Distance value decides the radius of this sphere. Distance value is taught in millimeter. Distance value is from 0 to 999.9mm. This sphere is referred as trigger region hereafter.

```
1:  L P[1] 2000mm/sec FINE DB 100.0mm  DO[1] = ON
2:  L P[2] 2000mm/sec FINE DB 100.0mm  DO[1] = ON
```

Figure 9.9.4 (a) Cyclical checks if TCP goes into trigger region.



Internally, Robot controller calculates current position to judge if TCP is in trigger region or not. Instruction part is executed when this calculated position is in trigger region.

CAUTION

Execution timing of instruction part is decided by distance (in millimeter). Because judgement to trigger is done by calculating distance between current position and destination point, actual execution timing is different from distance value.

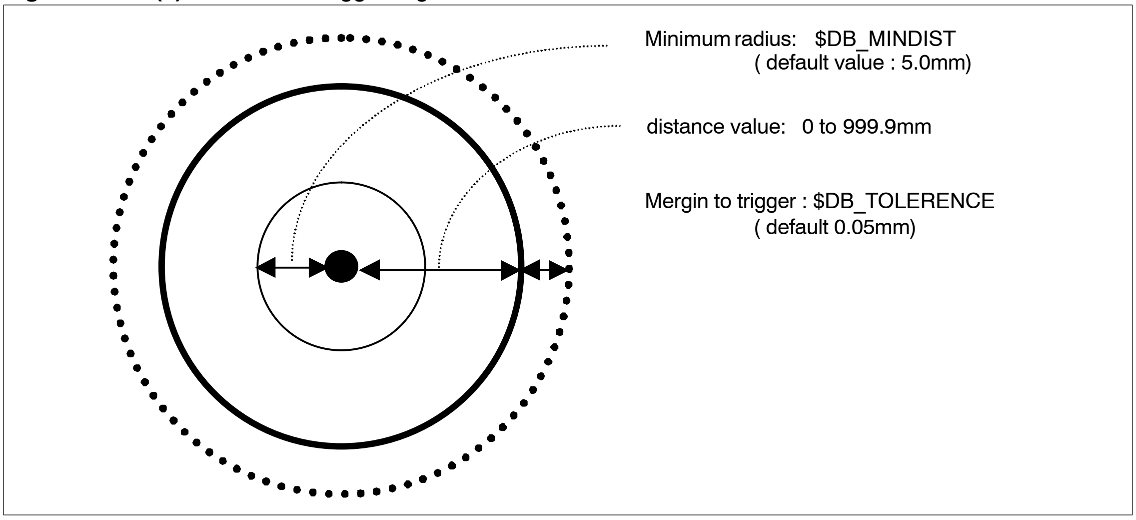
(Error in case of 2000mm/sec is estimated around 16mm)

(ii) Radius of trigger region.

Radius of trigger region is as follows.

Radius = (distance value or $DB_MINDIST)+$DB_TOLERENCE

Figure 9.9.4 (b) The size of trigger region



If distance value is less than $DB_MINDIST, $DB_MINDIST is used as distance value.

Example Suppose following motion statement is taught with $DB_MINDIST = 5.0

```
L P[1] 2000mm/secFINE DB 0.0mm DO[1]=ON
```

In this case, Robot controller interprets it as DB 5.0mm.Then $DB_TOLERENCE is added to decide radius of trigger region. Consequently, radius of trigger region is 5.05mm with default system variables.

3 Instruction part

This part shows what is done when TCP goes into trigger region. DB can do following action.

- CALL program
- Signal output

(i) DB Call program

Specified program is executed when condition is triggered.
Program to be called cannot use motion group.
(Change group mask to [*,*,*,*,*] in program header information screen.)
You can use arguments to call program.
Example)

```
L P[2] 1000mm/sec FINE DB 100mm, CALL A (1,2)
```

(ii) DB signal output

You can teach following signal output.
You can use one signal output for one DB.

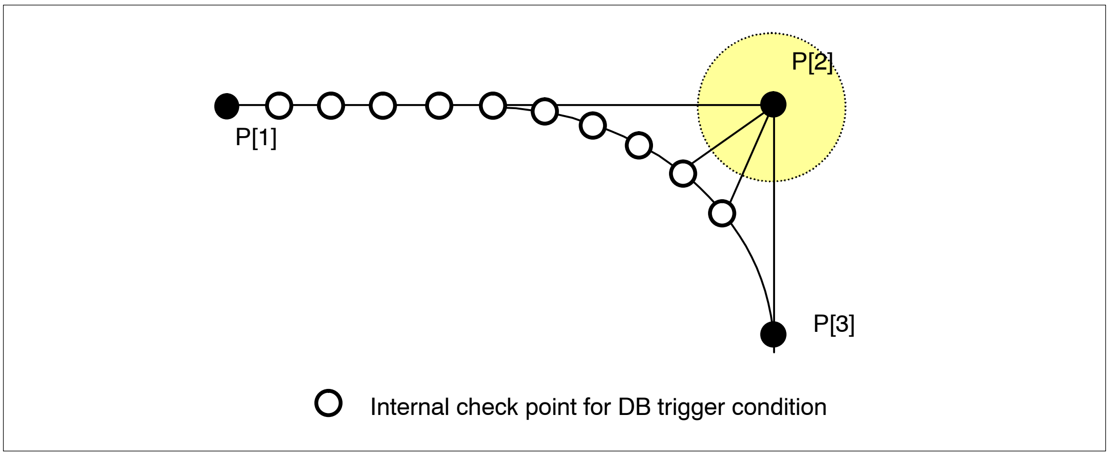
DO[] RO[]	=	ON OFF R[] pulse
GO[] AO[]	=	Constant R[] AR[ ]

You can also output signal by calling program which use signal output instruction. But to output only one signal with one DB, this direct signal output is better. It's easier to read and maintain.

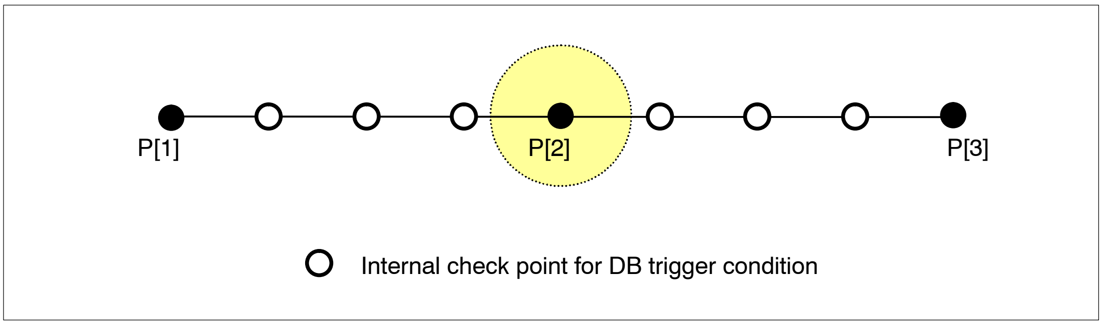
4 Changing trigger condition

Instruction part is executed when Robot controller recognizes that TCP is in trigger region.
But in some cases like following "going away" and "penetrate", robot controller doesn't recognize that TCP is in trigger region. These cases are described in this section.

Case 1 Trajectory of CNT motion doesn't go through trigger region.("going away")



Case 2 Trigger region is too small for controller to check current position in time.(“penetrate”)



For these case, the condition for instruction part to be executed (referred as DB condition) is changed by $DB_CONDTYP.

$DB_CONDTYP	DB condition	When alarm is posted.
0	TCP is in trigger region. (“region trigger”) + end of motion (*2)	“going away” +“penetrate” +end of motion
1 (default value)	“region trigger” +“going away” +“penetration” +end of motion	end of motion +(“going away”) (*1)
2	“region trigger ”+“penetration” +end of motion	“going away” end of motion

“going away” and “penetration ” is defined in (i), (ii) and (iii) respectively.

Distance Before executes instruction part when DB condition is satisfied. Otherwise, posts alarm.

There are two alarms for not–triggered DB. They are INTP–293 and INTP–295. $DBCONDTRIG decides which alarm is posted. Message is same but severity is different. Please refer to 5 for details.

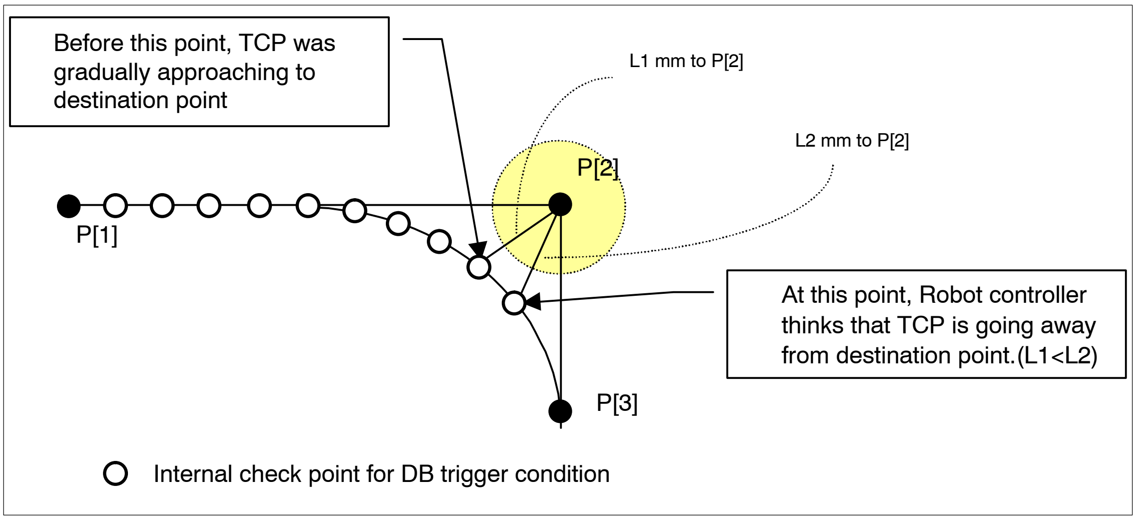
NOTE (*1) When Distance Before is triggered by “going away” in case of $DB_CONDTYP = 1, you can post alarm in addition to exectuion of instruction part. Please refer to 4 (i) for details.

(*2) By default configuration, if motion statement with Distance Before completes and robot stops before neither “region trigger” nor “going away” nor “penetration” trigger happens, Distance Before executes instruction part and post alarm. Please refer to 4 (iii).

(i) In case of going away.

If termination type is CNT and distance value is small, TCP may not go into trigger region.

Figure 9.9.4 (c) TCP doesn’t go into trigger region.



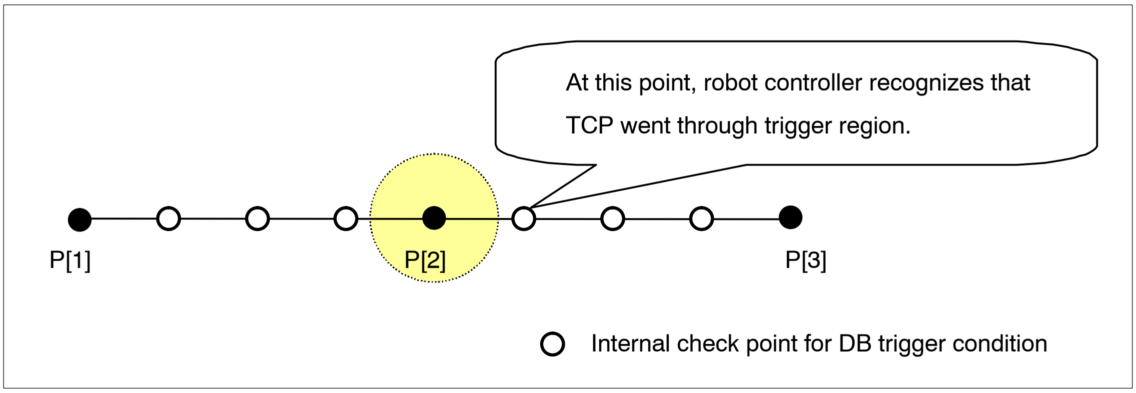
In case of Fig. 9.9.4 (c), TCP doesn't go into trigger region. TCP starts to go away from destination point (P[2]). Robot controller cyclically judges if TCP is going away from destination point or not in addition to DB condition. Robot controller recognizes that TCP is going away when calculated distance between current position and destination point is greater than previous one by more than ($DB_AWAY_TRIG) millimeter. This case is referred as "going away" in this manual.

- To post alarm in addition to execution of instruction part only when the DB is triggered by "going away" trigger, set $DB_AWAY_ALM to TRUE. DB executes instruction part and post following alarm.
  INTP–295 (program name, line number) DB too small (away)(%dmm)
  This is warning.

(ii) Penetration

This function cyclically checks if DB condition is triggered or not. Because of this cyclical check, CNT motion with high–speed may cause for Robot controller to omit cyclical check in small trigger region. See Fig.9.9.4 (d).

Figure 9.9.4 (d) Penetration



In this case, TCP moves too fast for R–J3 to check DB condition in small trigger region. Because cyclical check is done outside of trigger region, the fact TCP is in trigger region is not recognized by the robot controller.This case is referred as " penetration" in this manual.
To handle cases like Fig. 9.9.4 (d), Distance Before checks if TCP went through trigger region or not. If trajectory of TCP penetrated trigger region (penetration), instruction part is executed by default configuration.
But in this case, execution of instruction part is done after TCP passed away destination point.

- Motion with termination type FINE doesn't cause trigger by "penetration".

(iii) End of motion

If motion statement with DB completes and robot stops before "region","going away" and "penetration" is satisfied, DB executes instruction part and post following alarm.
INTP–297 (program name, line number) DB too small (done) (mm).
This alarm is not posted by FINE motion.
If you don't want this trigger, set $DB_MOTNEND to FALSE (default value:TRUE).
Distance displayed by this alarm is distance to destination.

> **CAUTION**
> If you stop your robot by E–stop when motion statement is about to complete, Distance Before may be trigger just after resume of the program.

> **CAUTION**
> If you halt a program when motion statement with DB is near its completion, DB may not be triggered. In this case, Distance Before executes its instruction part after resume of program.

5 Alarms for not –triggered Distance Before

Distance Before posts alarm if condition is not triggered. What is posted depends on $DBCONDTRIG.

$DBCONDTRIG	Alarm to be posted
0 (default value)	INTP–295 WARN (Program name, line number)DB condition was not triggered.
1	INTP–293 PAUSE.L (Program name, line number)DB condition was not triggered.

By default configuration, INTP–295 is posted. Because severity of this alarm is WARN, execution of program doesn't stop.
If you want to halt program when condition was not triggered, set $DB_CONDTRIG to 1.INTP–293 is posted when condition was not triggered. Program is halted for severity of this alarm is PAULSE.L .Robot decelerates to stop. Displayed distance is recommended value for the DB to be triggered by region trigger.

6 Step execution

If Distance Before CALL program is executed by step execution, program is halted at the timing sub program is called. The rest of motion statement is done by next step execution that executes sub program step by step.
Step execution of motion statement with DB signal output is just same as motion statement with out DB except signal output is done.

CAUTION

If distance value is small, program may be halted before completion of motion and before DB conditions are satisfied. In this case, Distance Before is not triggered by step execution of the line it is taught. The DB is triggered by execution of next line.

7 Halt and resume

Halt and resume of motion statement with DB changes its radius of trigger region. After resume, radius of trigger region is changed to minimum radius ($DB_MINDIST +$DB_TORELENCE).

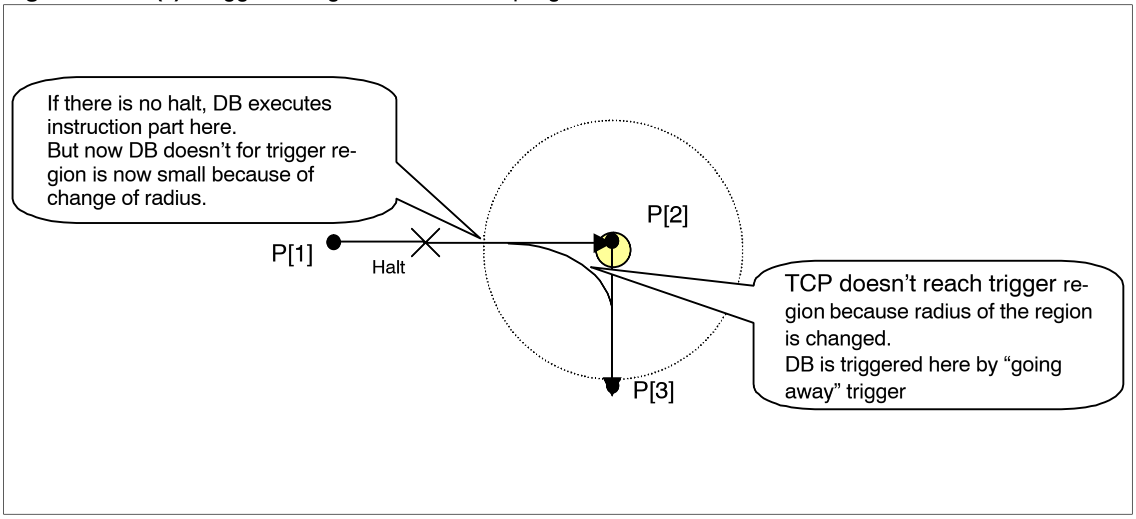
The purpose of this process is to execute instruction part after TCP reaches to its destination point.This prevents earier trigger because of halt and resume.
This means that halt and resume of program changes trigger timing of Distance Before. Not to change radius of trigger region, set $DISTBF_TTS to 0 (default value: 1).

Example Default configuration

Suppose following program is executed.

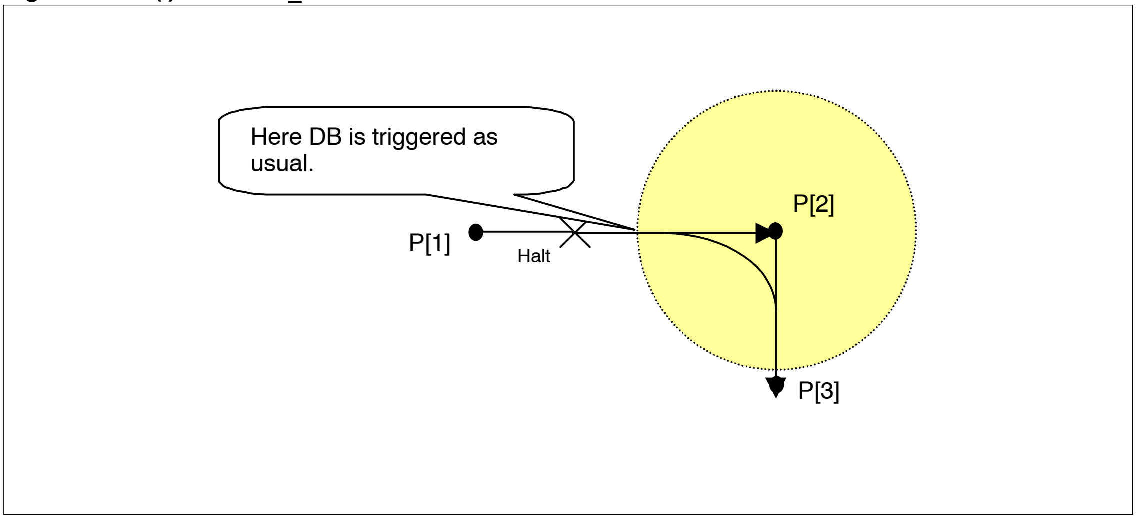
```
1: L P[1] 2000mm/sec FINE
2: L P[2] 2000mm/sec CNT100 DB 100.0mm CALL SUB
3: L P[3] 2000mm/sec CNT100
```

Figure 9.9.4 (e) Trigger timing after resume of program.



Example Resume with $DISTBF_TTS = 0

Figure 9.9.4 (f) $DISTBF_TTS = 0



8 Resume after JOG

If you halt motion statement with DB, JOG robot and resume program, execution timing depends on TCP position at the instant of program resume.
Because this procedure is accompanied by program halt, execution timing depends on $DISTBF_TTS, too.

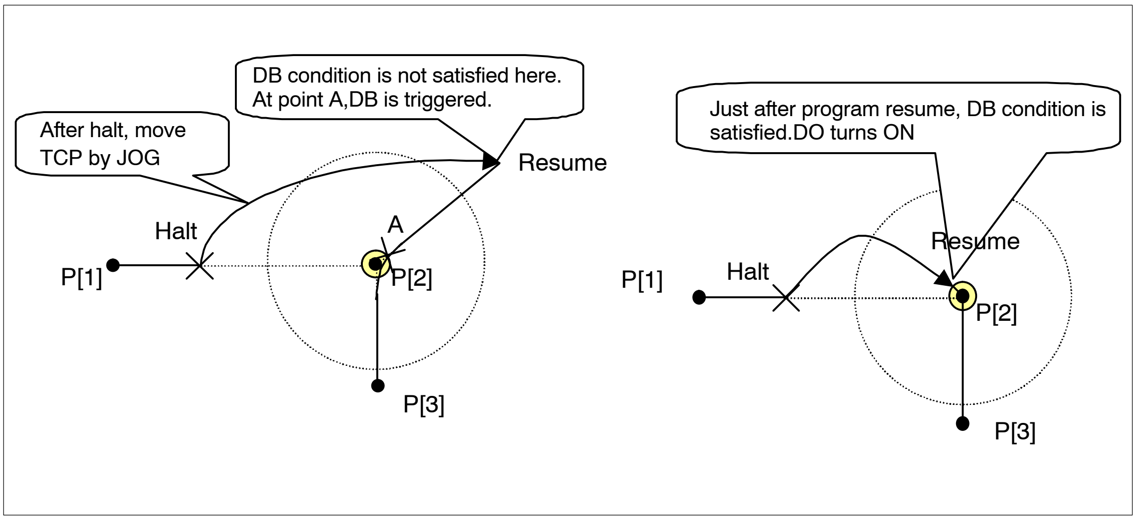
(i) Default configuration ($DISTBF_TTS = 1)

After resume of program, radius of trigger region changed to minimum value ($DB_MINDIST +$DB_TOLERENCE). If TCP is in new (diminished) trigger region, DB is triggered just after resume of program. If not, DB is triggered when DB condition is satisfied.

Example Suppose following program is executed and halted on line two. DB condition is not triggered yet.

```
1: L P[1] 2000mm/sec FINE
2: L P[2] 2000mm/sec CNT100 DB 100.0mm DO[1] = ON
3: L P[3] 2000mm/sec CNT100
```

Figure 9.9.4 (g) Resume after JOG



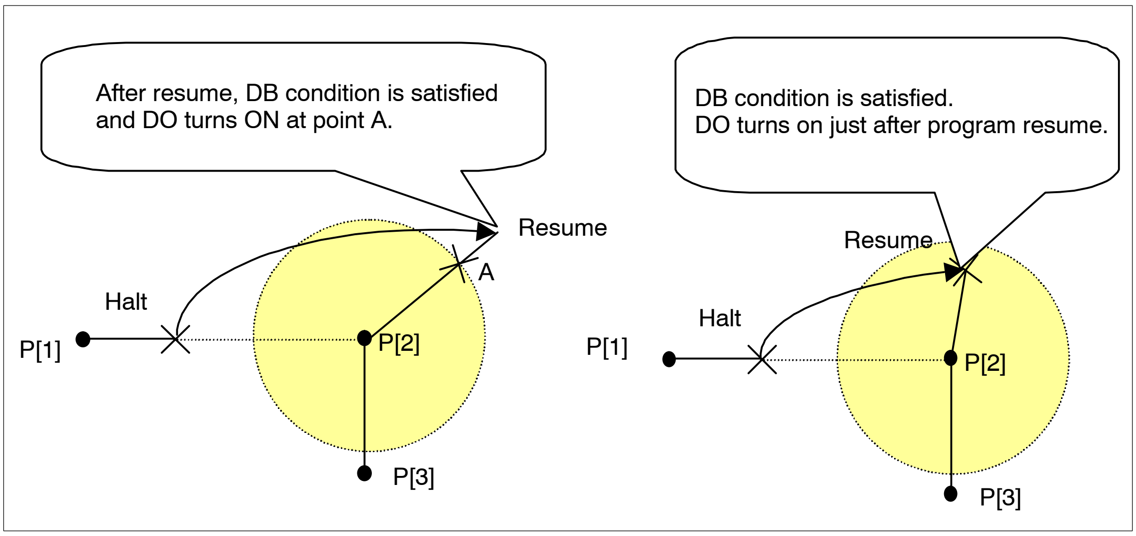
(ii) $DISTBF_TTS = 0

Radius of trigger region is not changed.
If TCP is in trigger region, DB is triggered just after resume of program. If not, DB is triggered when DB condition is satisfied.
Example) Suppose following program is executed and halted on line two.
DB condition was not satisfied yet.

```
1: L P[1] 2000mm/sec FINE
2: L P[2] 2000mm/sec CNT100 DB 100.0mm DO[1] = ON
3: L P[3] 2000mm/sec CNT100
```

If TCP is distant from P[2] enough not to trigger(more than 100mm away), DO[1] turns ON when DB condition is triggered by motion after resume, at point A in left diagram in Fig.9.9.4 (h).
If TCP is in trigger region when you resume program, DO[1] turns ON just after resume. (right diagram in Fig. 9.9.4 (h).

Figure 9.9.4 (h) Resume after JOG($DISTBF_TTS = 0)



9 Power failure recovery

If power is turned down during sub program execution and power failure recovery is enabled, resume after power failure recovery executes the rest of sub program. In this case, sub program is executed where TCP was at power failure. Execution timing is different from usual one.

9.9.5 Entering Distance Before

1 DB call program

1) Move cursor to motion option area.

```
PNS0001      LINE 1              ABORTED
PNS0001                         JOINT 10 %
                                     1/2


   1:J P[1] 100% FINE
[End]



                        [CHOICE]
```

2) Press F4.List of motion option is displayed.

```
Motion Modify                   JOINT 10 %
 1 TIME BEFORE      5
 2 TIME AFTER       6
 3 DISTANCE BEFORE  7
 4                  8 ---next page---
PNS0001
                                    1/2
   1:J P[1] 100% FINE
[End]


Select item
                        [CHOICE]
```

3) Select DISTANCE BEFORE.DB is added to program.

4) Input distance value and press Enter. Submenu to select instruction part is displayed.

```
TIME statement                 JOINT  10 %
 1 CALL program      5 GO[  ]=...
 2 CALL program( )   6 AO[  ]=...
 3 DO[  ]=           7
 4 RO[  ]=           8
PNS0001
                                     1/2
   1: J P[1] 100% FINE
      DB 100.0mm ...

[End]
```

5) To use argument, select CALL program( ).If you don't, select CALL program. Program list is displayed anyway.

```
PROGRAM list                   JOINT  10 %
 1 HAND_OPEN         5
 2 HAND_CLOSE        6
 3                   7
 4                   8
PNS0001
                                     1/2
  1: J P[1] 100% FINE
      DB  100.0mm CALL ...( )
```

6) Select program to call.

```
Parameter select                 JOINT  10 %
 1 R[ ]                    5 <None>
 2 Constant                6 <Insert>
 3 String                  7
 4 AR[ ]                   8
PNS0001
                                       1/2
   1: J P[1] 100% FINE
      DB100.0mm CALL HAND_OPEN(     )

[End]
```

To specify argument, following procedure is needed.

7) Select argument type. Screen displayed below is example to use Constant.

```
PNS0001          LINE 1         ABORTED
PNS0001                          JOINT  10 %
                                         1/2


   1:J P[1] 100% FINE
     DB 100.0mm CALL HAND_OPEN(Constant)
[End]

                          [CHOICE]
```

8) Input value of argument.

- To use more than 2 arguments, move cursor to ")" and press F4[CHOICE]. Submenu to select argument type is displayed. Teach argument by procedure 7) and 8) described above.
- To delete argument, move cursor to argument you want to delete and press F4. Then select <Insert>.

To add argument to CALL without argument, following procedure is needed.

1 Move cursor to program name.

```
    1:J P[1] 100% FINE
     DB 100.0mm CALL A
```

2 Press prev key 2 times. Following submenu is displayed.

```
TIME statement                   JOINT  10 %
 1 CALL program            5 GO[  ]=...
 2 CALL program( )         6 AO[  ]=...
 3 DO[  ]=                 7
 4 RO[  ]=                 8
PNS0001
```

3 Select CALL program ().

```
PROGRAM list                     JOINT  10 %
 1 HAND_OPEN               5
 2 HAND_CLOSE              6
 3                         7
 4                         8
PNS0001
                                       1/2
  1: J P[1] 100% FINE
      DB  100.0mm CALL ...( )
```

4 Select program to call and teach argument.

2 DB Signal output

1) Do just same procedure 1)–4) for DB CALL program. Submenu to select instruction is displayed.

```
TIME statement                JOINT  10 %
 1 CALL program      5 GO[  ]=...
 2 CALL program( )   6 AO[  ]=...
 3 DO[  ]=           7
 4 RO[  ]=           8
PNS0001
                                    2/2
   1: J P[1] 100% FINE
      DB 100.0mm ...

[End]
```

2) Select signal output instruction.

```
PNS0001
                                    2/2
   1: J P[1] 100% FINE
      DB 100.0mm DO[...]=...

[End]


Select item
                       [CHOICE]
```

3) Input index and output value just as you do for normal I/O instruction.

```
PNS0001          LINE 1         ABORTED
PNS0001                       JOINT  10 %
                                      1/2


   1:J P[1] 100% FINE
     DB  100.0mm DO[1]= ON
```

3 Finding/Replacing Instructions

- Finding Instructions

You can find program which is used for DB by “find” on F5 pull–up menu. By selecting “CALL” then “Call program” to find program used in DB.
You can find signal output instruction by this function, too. Select item “I/O” on submenu.

- Replacing Instructions

Distance Before can be replaced to TIME BEFORE/AFTER by “replace” on F5 pull–up menu. Select “TIME BEFORE/AFTER” on replace item submenu.
You can also replace CALL and signal output in instruction part just as you do when you replace usual CALL and DO etc.

9.9.6 Caution and limitations

- Distance Before cannot be used with TIME BEFORE/AFTER.
- More than 6 motion statement with Distance Before cannot be processed at the same time.
- Distance Before calculates distance between current position and destination point cyclically. Because trigger condition is judged by this cyclical check, actual execution timing of instruction part is different from distance value. Instruction part may be executed inside of trigger region. This means the point where instruction is executed is closer than distance value. Degree of error depends on speed of robot. The slower TCP moves, the more accurate execution timing.
- Distance Before is not recovered by power failure recovery if it was attached to CNT motion statement and power is down when the motion is about to complete.
- Distance Before cannot be used with INC, skip and quick skip in a motion statement.

- Multi group is not supported.
- Robots that don't have Cartesian coordination are not supported.
- Position data in matrix form is not supported.
- Integrated axis is not supported.
- FANUC Robot F–200*i* is not supported.
- Line tracking is not supported.
- If CJP or ACCUPATH is used, use this function with $DB_MOTNEND = TRUE.
- If program ends before DB condition triggers, execution instruction is not processed even if DB condition is triggered after program execution completed.
- During deceleration due to program halt, "going away" trigger may not work.
  In this case. DB is triggered after program resume.
- After E–stop, DB doesn't work. If TCP passes by destination point, DB is triggered after resume of program.
- After E–stop and resume or program, DB may be triggered just after resume.
- Single step execution of DB of small distance value may fail for program is paused before motion statement completes and DB condition satisfied. The DB is triggered by execution of next line.
- If DB condition is satisfied after pause of program, DB is not triggered by step execution of the line. In this case, the DB is triggered by execution of next line.

9.9.7 System Variables

system variable	role	default value
$DISTBF_VER	This system variable set execution timing of line which is just after motion statement with DB. 1(default) : Execution of next line doesn't wait completion of instruction part of DB. 2: Execution of next line waits for completion execution of instruction part. **Example** Suppose following program is executed. 1: L P[1] 2000mm/sec FINE 2: L P[2] 2000mm/sec CNT100 DB 1.0mm DO[1] = ON 3: L P[3] 2000mm/sec FINE With $DISTBF_VER=2, execution of line 3 doesn't start until DO[1] is turned to ON. With $DISTBF_VER = 1, line 3 is executed just as if there were no DB.	1
$DB_AWAYTRIG	Distance Before calculate distance between current position and destination cyclically.robot controller recognize TCP is "going away" from destination point if this calculated distance is greater previous value by $DB_AWAYTRIG millimeters. Please refer to 9.9.4, 4 (i) for details.	0.08(mm)
$DB_AWAY_ALM	This system variable decides whether INTP–295 is posted or not when DB is triggered by "going away" with $DB_CONDTYP = 1. Please refer to 9.9.4, 4 (i) for details.	FALSE
$DB_TOLERENCE	The radius of trigger region is distance value +$DB_TOLERENCE. (If distance value <$DB_MINDIST, radius is $DB_MINDIST +$DB_TOLERENCE) Please refer to 9.9.4, 2 (ii) for details.	0.05(mm)
$DB_CONDTYP	This system variable defines DB trigger condition. 0:When TCP goes into a region which is within distance value ("region trigger") 1:In addition to "region trigger", in case of "going away" 2:In addition to "region trigger", in case of "penetrate". Please refer to 9.9.4, 2 and 9.9.4, 4 for details.	1

system variable	role	default value
$DBCONDTRIG	This system variable decides alarm that is posted when DB condition was not triggered. 0:INTP–295 WARN "(program name, line number) DB condition was not triggered." is posted. 1: INTP–293 PAUSE.L "(program name, line number) DB condition was not triggered." is posted. Please refer to 9.9.4, 5 for details.	0
$DB_MINDIST	Internal minimum value of distance value. If distance value is smaller than this value by $DB_MINDIST or more, $DB_MINDIST is used as distance value instead of distance value user taught. Please refer to 9.9.4, 2 (ii) for details.	5.0(mm)
$DB_MOTNEND	This system variable decides if motion completion trigger DB or not. Please refer to 9.9.4, 4 (iii).	TRUE
$DISTBF_TTS	This system variable decides execution timing of instruction part after motion statement with DB is halted. Please refer to 9.9.4, 7 for details.	1

9.9.8 Error Codes

Following alarms are related to Distance Before.

INTP–292 PAUSE.L

More than 6 motion with DB executed.

[Cause] more than 6 Distance Before were processed at the same time.

Example

```
1:  L P[1] 2000mm/sec CNT100 DB 10mm DO[1] = ON
2:  L P[2] 2000mm/sec CNT 100 DB 10mm DO[2] = ON
3:  L P[3] 2000mm/sec CNT 100 DB 10mm DO[3] = ON
4:  L P[4] 2000mm/sec CNT 100 DB 10mm DO[4] = ON
5:  L P[5] 2000mm/sec CNT 100 DB 10mm DO[5] = ON
6:  L P[6] 2000mm/sec CNT 100
7:  L P[7] 2000mm/sec CNT 100 DB 10mm DO[7] = ON
8:  L P[8] 2000mm/sec CNT 100 DB 10mm DO[8] = ON
9:  L P[9] 2000mm/sec CNT 100 DB 10mm DO[9] = ON
10: L P[10] 2000mm/sec CNT 100 DB 10mm DO[10] = ON
11: L P[11] 2000mm/sec CNT 100 DB 10mm DO[11] = ON
```

If CNT motion statement with DB frequently like this example, more than 6 calculation for Distance Before may be done at the same time.

[Remedy] Change termination type from CNT to FINE. Otherwise, change structure of program not to execute DB frequently.

INTP–293 PAUSE.L

(program name, line number) DB too small (away) (distance mm)

[Cause] Condition of Distance Before was not triggered.

[Remedy] Change program for TCP to move into trigger region.

INTP–295 WARN

(program name, line number) DB too small (away)(distance mm)

[Cause] Condition of Distance Before was not triggered.

[Remedy] Change program for TCP to move into trigger region.

INTP–296 WARN

(program name, line number) $SCR_GRP[1].$M_POS_ENB is FALSE.

[Cause] $SCR_GRP[1].$M_POS_ENB is FALSE.

[Remedy] Change $SCR_GRP[1].$M_POS_ENB to TRUE

INTP–297 WARN

(program name, line number) DB too small(done) (distance mm)

[Cause] DB is triggered by completion of motion statement to which it is attached.

[Remedy] Use greater distance value. This is best solution.
If you do not want this trigger, set $DB_MOTNEND to FALSE.

9.10 State Monitoring Function

This function accepts, as conditions, the values of the input/output signals, alarms, and registers of the robot controller (referred to simply as the controller), and executes the specified programs if the conditions are satisfied. The controller itself monitors these conditions.

This function consists of the following instructions and programs:

- Monitor start instruction
  Specifies the condition program to be monitored and the start of monitoring.
  Example:
  ```
  1:MONITOR WRK FALL
  ```
  Condition program name

- Monitor stop instruction
  Specifies the condition program to terminate.
  ```
  9:MONITOR END WRK FALL
  ```
  Condition program name

- Condition program
  Describes the condition to be monitored and specifies the program to be executed if the condition is satisfied.
  Program example:
  ```
  1:WHEN DI[2]=Off, CALL STP RBT
  ```
  *1 *2
  This condition program states that when RDI[2] turns off, STP RBT is to be called.
  *1 Describe the desired monitoring condition by following instruction WHEN. The types of monitoring condition are explained in the WHEN section.
  *2 Specify the program to be executed if the condition described in *1 is satisfied. The action program can be created and named in the same way as a normal program.

- Action program
  Called if the condition is satisfied. The same instructions as those used in normal programs can be used.
  Program example:
  ```
  1:DO[2]=On        ! Notification to a peripheral device
  2:R[8]=R[8]+1     ! Drop count
  3:UALM[1]         ! Alarm and robot stop
                $UALRM_MSG[1]=WORK HAS FALLEN
  ```

With the following program example, if the robot performing handling drops a workpiece, the user is alerted with an error message and the robot is stopped.

Sample. TP (program for handling operation)

```
 1:MONITOR WRK FALL
 2:J   P[1]   100% FINE
    :
    :
    :
    :
 8:J   P[7]   100% FINE
 9:MONITOR END WRK FALL
10:Open hand
```

Handling operation (lines 2–8)
State monitoring (lines 1–9)

Workpiece drop. condition (condition program)
```
1:WHEN DI[2]=Off, CALL STP RBT
```

Robot stop. TP (action program)
```
 1:DO[2]=On        ! Notification to a peripheral device
 2:R[8]=R[8]+1     ! Drop count
 3:UALM[1]         ! Alarm and robot stop
[End]
```

Monitor types

There are two main types of monitors: the program monitor and system monitor.

- The program monitor starts/stops from a mnemonic program (referred to simply as a program). When the program terminates, monitoring also terminates.
- The system monitor is started/stopped from the dedicated screen. It performs monitoring constantly regardless of the execution state of the program. (Monitoring continues even after the program terminates.)

Program monitor

This type of monitor depends on the execution state of the program. It is suitable for state monitoring within a separate program.

Monitoring starts with an instruction (monitor start instruction) in the program. Monitoring terminates with a monitor stop instruction or program termination.

The program monitor can be switched between two settings: setting 1 in which the monitor stops when the program stops temporarily, and setting 2 in which the monitor continues monitoring.

NOTE Settings 1 and 2 cannot be used at the same time.

System monitor

This type of monitor does not depend on the execution state of the program. It is suitable for monitoring the state of the entire system.

The monitor is started and stopped from the state screen. It cannot be operated with instructions in the program.

The system monitor can be switched between two settings: setting 1 in which the monitor stops after a cold start, and setting 2 in which the monitor continues monitoring.

NOTE The program monitor and the system monitor can be used at the same time.

The monitors can be switched between the settings using the following system variables:

$TPP_MON.$LOCAL_MT = 1D Switches the program monitor to setting 1 (default).
$TPP_MON.$LOCAL_MT = 2D Switches the program monitor to setting 2
(same specification as that for KAREL)
$TPP_MON.$GLOBAL_MT= 0D Enables the system monitor (default).
$TPP_MON.$GLOBAL_MT= 1D Switches the system monitor to setting 1.
$TPP_MON.$GLOBAL_MT= 2D Switches the system monitor to setting 2.

Monitor state transition

The states of the monitors assumed when each operation is performed are listed in the table below:

Operation	Program monitor		System monitor	
	Setting 1	Setting 2	Setting 1	Setting 2
MONITOR instruction	●	●	–	–
RESTART (state screen)	○	○		
START (state screen)			●	●
Program Stop	△	– *1)	–	–
Program End/Enforced End	×	×	–	–
MONITOR END	×	×	–	–
PAUSE (state screen)	△	△	×	×
END (state screen)	×	×		
RESUME	○	○	–	–
Power failure handling Power off with monitoring state	△	–	–	–
Power failure handling Power off without monitoring state	–	–	–	–
START (COLD)	×	×	×	–
CONTROLLED START	×	×	×	×
Other operation	–	–	–	–

Meanings of symbols

● : State monitoring is started.
○ : State monitoring is restarted if it is stopped.
△ : State monitoring is stopped.
× : State monitoring is deleted. (Cannot be restarted)
– : The state of state monitoring does not change due to the operation.
*1 : Monitoring continues, but the action program will pause even if the conditions are satisfied.

Operation–by–operation description

Operation	State
MONITOR instruction	When a monitor start instruction in the program is executed, monitoring with the specified program monitor starts.
RESTART (state screen)	When function key RESTART is pressed on the Program monitor screen of the state screen, monitoring with the program monitor specified with the cursor restarts.
START (state screen)	When function key START is pressed on the System monitor screen of the state screen, monitoring with the system monitor specified with the cursor starts.
Program Stop	When the temporary stop key is pressed or if the program stops temporarily due to the occurrence of an alarm, state monitoring with the program monitor previously started by the temporarily stopped program stops, if the program monitor is set to 1.
Program End/Enforced End	When the program terminates due to program termination, forced termination, or the occurrence of an alarm, the program monitor previously started by the terminated program is deleted. The deleted program monitor does not start unless a monitor start instruction is executed.
MONITOR END	When a monitor stop instruction in the program is executed, the specified program monitor is terminated. The terminated program monitor does not start unless a monitor start instruction is executed.
PAUSE (state screen)	When function key PAUSE is pressed on the Program monitor screen of the state screen, monitoring with the program monitor specified with the cursor stops. The stopped monitor restarts when the “Restart” key is pressed or the program restarts. When function key PAUSE is pressed on the System monitor screen of the state screen, monitoring with the system monitor specified with the cursor stops.
END (state screen)	When function key END is pressed on the Program monitor screen of the state screen, the program monitor specified with the cursor stops.
RESUME	When the temporarily stopped program restarts, the stopped program monitor restarts.
Power failure handling	If power failure handling is enabled and the monitor is monitoring, the following occurs when the power is turned OFF/ON. ● State monitoring stops if the program monitor is set to setting 1. ● State monitoring continues if the program monitor is set to setting 2. (The program stops temporarily, but state monitoring is performed.) ● The system monitor continues state monitoring. If the monitor is stopped, it remains in the stopped state when the power is turned OFF/ON.
Cold start	If power failure handling is disabled and the power is turned OFF/ON, all monitors terminate except the system monitor of setting 2. The system monitor of setting 2 maintains the state assumed before the power was removed.
Other	For operations other than the above, the monitor state is preserved.

Instruction statements

State monitoring is performed in the section enclosed by the following instructions:

- MONITOR <conditional–program–name>
  Monitoring starts under the condition described in the condition program.
- MONITOR END <conditional–program–name>
  Monitoring performed under the condition described in the condition program stops.

Condition program

The monitoring condition program, which has the subtype called WHEN, can specify condition instructions only.

- WHEN <conditional–expression>, CALL <program–name>

The following conditions can be used:

Figure 9–21. Register/System Variable Condition Compare Instruction

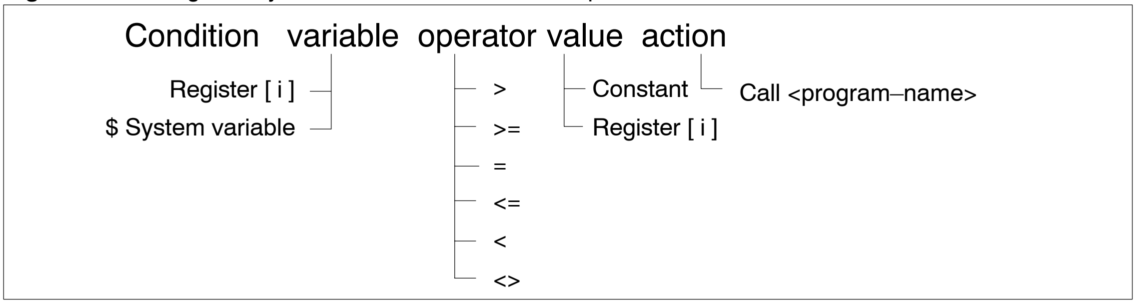


Figure 9–22. I/O Condition Compare Instruction 1

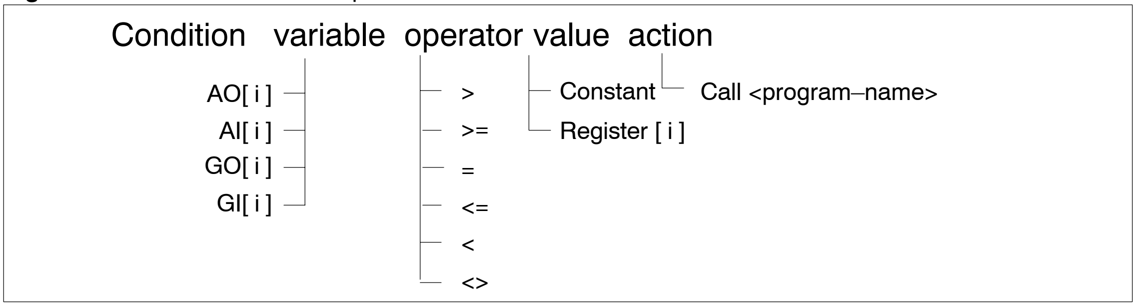
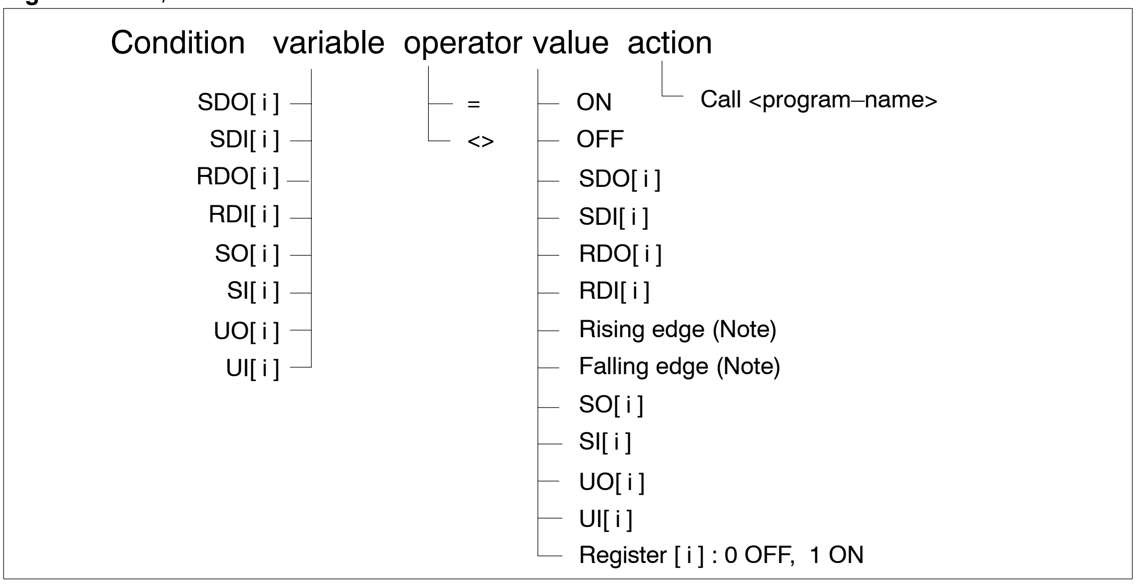


Figure 9–23. I/O Condition Wait Instruction 2



NOTE Falling edge: The falling edge of a signal is regarded as being a detection condition. The condition is not satisfied when the signal remains off. The detection condition is satisfied when the signal changes from the on state to the off state.
Rising edge: The rising edge of a signal is regarded to be a detection condition. The condition is not satisfied when the signal remains on. The detection condition is satisfied when the signal changes from the off to the on state.

Figure 9–24. Error Condition Compare Instruction

Condition error number = Value   Processing

Constant (Note)   Call <program–name>

NOTE An error number is specified with an alarm ID followed by an alarm number.
Error number = aabbb
where aa = alarm ID
bbb = alarm number

For an explanation of alarm IDs and numbers, refer to the alarm code table in the operator's manual.
(Example) For SRVO006 Hand broken, the servo alarm ID is 11, and the alarm number is 006. Thus,
Error number = 11006

In the condition compare instruction, multiple conditions can be specified on a single line in the condition statement, using the logical operators ("and" and "or"). This simplifies the program structure, allowing the conditions to be evaluated efficiently.

Instruction format

- Logical product (and)
```
WHEN <Cond.1> AND <Cond.2>, CALL <PRG Name>
```
- Logical sum (or)
```
WHEN <Cond.1> OR <Cond.2>, CALL <PRG Name>
```

If the "and" (logical product) and "or" (logical sum) operators are used in combination, the logic becomes complex, impairing the readability of the program and the ease of editing. For this reason, this function prohibits the combined use of the "and" and "or" logical operators.

If multiple "and" (logical product) or "or" (logical sum) operators are specified for an instruction on a single line, and one of the operators is changed from "and" to "or" or from "or" to "and," all other "and" or "or" operators are changed accordingly, and the following message appears:

```
TPIF-062 AND operator was replaced to OR
TPIF-063 OR operator was replaced to AND
```

Up to five conditions can be combined with "and" or "or" operators on a single line.

(Example)
```
WHEN <Cond.1> AND <Cond.2> AND <Cond.3> AND <Cond.4> AND <Cond.5>, CALL
<PRG Name>
```

Specification

Step

1 Enter a condition program name.
On the program list screen, press F2 CREATE and enter a program name.

2 Select Cond as the subtype.
Press F2 DETAIL to move to the program details screen.
Position the cursor to the subtype item and press F4 CHOICE.
Select Cond from the subwindow.

NOTE At this time, the operation group is automatically set as [*,*,*].
A condition program requires no operation group.

State monitoring screen

The state of state monitoring can be monitored using the program monitor screen and the system monitor screen.

Program monitor screen

For the program monitor currently being executed or stopped, the name and state (under execution, stopped) of the condition program is displayed, as well as the name of the parent program(*1) of the program that started the program monitor.

NOTE If program “A” calls program “B” with a subprogram call, and program “B” executes a monitor start instruction, the name of the parent program, “A,” is displayed in the program name column.

```
Program monitor                  JOINT  10%

      CH Prog.   Status    Program

   1  WRK FALL   Running   Sample
   2  WLD TIME   Paused    Sample
   3  NO WRK L   Paused    Sample 2




[TYPE]   SYSTEM   RESTART   PAUSE   END
```

Table 9–7. Items and Function Keys on the Program Monitor Screen

Item	Description
CH Prog.	Condition program name
Status	State of the program, either being executed or stopped
Program	Name of the parent program of the program that started the program monitor
F2 SYSTEM	Switches the screen to the system monitor screen. If the system monitor is disabled ($TPP_MON.$GLOBAL_MT=0), this key is not effective.
F3 RESTART	When pressed, this key restarts the stopped monitor.
F4 PAUSE	Stops the monitor.
F5 END	Terminates the monitor. The terminated monitor is cleared from the screen.

System monitor screen

All condition programs are displayed. System monitors can be started and stopped.

```
System monitor                   JOINT  10%

      CH Prog.   Status

   1  WRK FALL   Running
   2  WLD TIME
   3  NO WRK L
   4  VRFY HND



[TYPE]   PROGRAM    START          END
```

Table 9–8. Items and Function Keys on the System Monitor Screen

Item	Description
CH.Prog.	Condition program name
Status	State of the program, either being executed or not started (blank)
F2 PROGRAM	Switches the screen to the program monitor screen.
F3 START	Starts the system monitor.
F5 END	Stops the monitor. In the “State” column, a blank is displayed for the stopped monitor.

Notes/restrictions

If multiple condition instructions are specified in a condition program, multiple monitors are started at the same time.

```
1:WHEN (conditional-expression1),  CALL (Program name1)
2:WHEN (conditional-expression2),  CALL (Program name2)
3:WHEN (conditional-expression3),  CALL (Program name3)
```

If, before one monitor start instruction terminates, another monitor start instruction is executed, both monitors are executed at the same time.
If the condition program names specified in the monitor start instructions are the same, the first condition program is overwritten by the second.

The program monitor stops state monitoring under the following conditions:

- The MONITOR END instruction is executed.
- The program terminates.
- The program stops temporarily. (State monitoring restarts when the program restarts.)

Up to ten conditions can be monitored at the same time.
Up to five "and" or "or" operators can be specified in a single monitoring condition instruction.

```
                                                  Up to five
                                           ┌──────────────────────┐
condition (conditional–expression1) and (conditional–expression2)  .........  and (conditional–expression5)  ┐
condition (conditional–expressionn)or (conditional–expressionm)    .........  or (conditional–expression1)   │ Up to ten
                :                              :                                   :                        │
condition (conditional–expressiono) and (conditional–expressionp)  .........  and (conditional–expressionq)  ┘
```

While the program is being executed or while it is stopped, the condition statements (condition program) cannot be edited.

In the action program for a system monitor, an operation group cannot be specified. In the action program for a system monitor, the operation group must be specified as [*,*,*,*,*].

In the action program for a program monitor, an operation group can be specified. While the robot is operating, however, the robot cannot be operated with the program. While the robot is not operating, the robot can be operated with the program.

If the condition is satisfied, the condition program enters the END state. If condition monitoring is to continue, specify a monitor start instruction in the program. Clear the monitoring condition beforehand.
Example

```
MON1. TP
  1:WHEN R[1]=1 CALL ACT1
ACT1.TP
  1:R[1]=0  <――――――― Clear the condition
  2:
  3:(Action)
       :

  9:MONITOR MON1 <――――――― Start the monitoring condition again
```

If there is no line on which the condition on line 1 is dropped, the condition is immediately satisfied on the monitor start instruction on line 9, causing a MEMO–065 error.

The condition program cannot be executed directly.

9.11 Automatic Error Recovery Function

9.11.1 Overview

This section consists of the following items:

a) Outline of the automatic error recovery function
b) Defining a resume program
c) Teaching the RETURN_PATH_DSBL instruction
d) Setting screen of the automatic error recovery function
   – Enabling/disabling the automatic error recovery function
   – Defining alarm codes to be monitored
   – Defining the recovery switch SDI
   – Defining the error recovery information SDO (indicating the conditions for executing the resume program)
   – Enabling/disabling the alarm–time automatic start feature
   – Setting the maximum number of automatic start repetitions
   – Defining the automatic start count register
   – Defining automatic error recovery alarm conditions
e) Flowchart for resuming a suspended program
f) Manual operation screen of the automatic error recovery function
g) Execution of the resume program from the teach pendant and test mode
h) Changing conditions for executing the resume program
i) Other specifications and restrictions
j) Warnings (Be sure to read this section for safety.)

This function is an optional function.

9.11.2 Outline of the automatic error recovery function

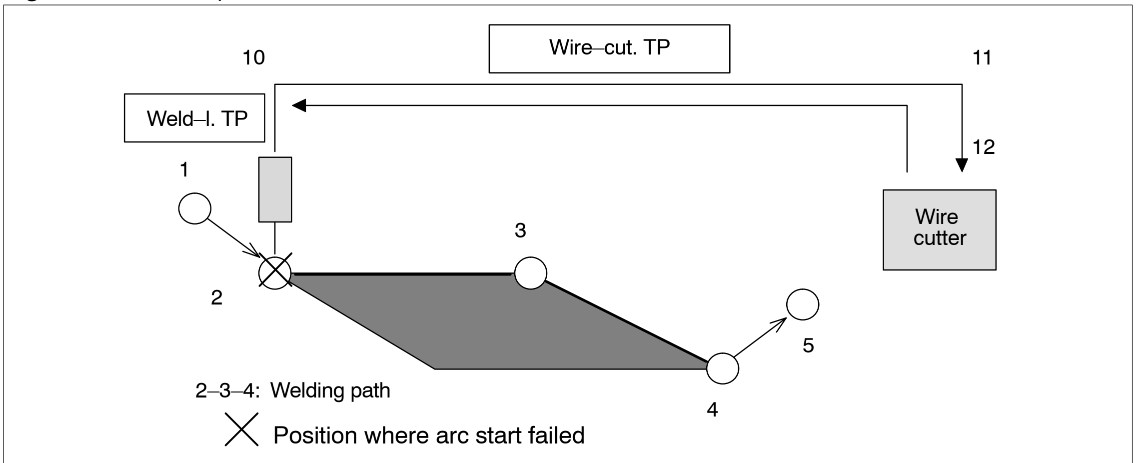
Background

Robots are sometimes stopped by various alarms even during production. If a robot is stopped, it is necessary to perform recovery operation then resume the program that was originally running.

For example, suppose that a robot is performing arc welding. An alarm due to an arc start failure may be issued, stopping the robot. In such a case, the operator must jog the robot to a safe position to, for example, cut the end of the wire or clean the nozzle, then resume the original program.

The automatic error recovery function is provided to support automatic operation of the above sequence.

Figure 9–25. Example:



Alarm code monitoring function

In the example shown above, the robot is operated by executing Weld–1.TP to perform welding along the path from 2 to 3 to 4. Assume that an arc start failure occurs at the arc start position 2. With the automatic error recovery function, another program called the resume program, which is Wire–Cut.TP in this case, can be started at the next start signal input. After this program terminates, another start signal input resumes the original program. If the resume operation function is then enabled (which is set on the welding system setting screen), the robot automatically returns to the original position where the robot was stopped, then the original program is resumed. If the return distance for resume operation is set, the robot returns from the stop position by the set distance, then the original program is resumed. If no arc is produced, a scratch start takes place.

In the above example, the automatic error recovery function operates only when an arc start miss alarm is issued. The alarm to be monitored may be changed, or the number of alarms to be monitored may be increased. For example, an arc off alarm can be added as an alarm to be monitored. Then, when an arc off alarm is issued, the same operation sequence as explained above can be performed automatically.

The standard maximum number of monitored alarm codes that can be defined is ten. If no alarm code is defined, the alarm code monitoring function is disabled. In this case, before the suspended original program (Weld_1.TP in the example) is resumed, the resume program is always executed.

Recovery switch SDI function

With the recovery switch SDI function, whether to start the resume program or not can be selected at the time of start input according to the defined SDI status. If the recovery switch SDI is off, the original program is resumed without executing the resume program. When this SDI is not defined, this function is disabled.

Error recovery information SDO function

With the error recovery information SDO function, whether the next start input resumes the original program or executes the resume program can be indicated. If the error recovery information SDO is on at start input, the resume program is executed. When this SDO is not defined, this function is disabled.

Alarm–time automatic start feature

When an alarm code is defined as explained before, and the defined alarm is issued, the program outputs an alarm signal and stops running. Input of the start signal executes the defined resume program. After the resume program terminates, another start signal input restarts the suspended original program.

When the alarm–time automatic start feature is enabled, and a defined alarm is issued, the resume program is automatically executed without outputting the alarm signal and stopping the robot. When the resume program has terminated, the original program is resumed automatically. When this feature is enabled, therefore, the input of the start signal is no longer needed.

Because the alarm signal is not output, other robots are not stopped when multiple robots are operating. The robot for which the alarm was issued moves by itself to the recovery station, and after recovery work, the original program is resumed.

> **CAUTION**
>
> Basically, the automatic error recovery function is designed so that it functions when the teach pendant is disabled. When the teach pendant is enabled, the automatic error recovery function does not function unless the manual test mode is set on the automatic error recovery manual operation screen. For manual testing, see “Execution of the resume program from the teach pendant and test mode.”

DI alarm function

By inputting a defined digital input signal, an automatic error recovery alarm can be issued. When this alarm is defined for the alarm–time automatic start feature, the resume program can be executed automatically by inputting the digital input signal.

As the message for an automatic error recovery alarm, a message defined for a user alarm can be used. The alarm severity can be set to either LOCAL or GLOBAL selectively. When LOCAL is selected, the alarm is issued only for a program that defines the resume program. The status of a digital input signal to be monitored can be set by selecting the signal type from among DI, RI, and WI, changing the signal number, and selecting the trigger status between on and off.

Function for disabling the resume operation function after execution of the resume program

In arc tool systems, the resume operation function is generally enabled. With this function enabled, return to the original stop position is always performed then arc is produced when the original program resumes after the resume program terminates.

In some systems, however, return to the original stop position should not sometimes be performed. For example, when the nozzle touch state is input through DI, a resume program is used to relieve the torch slightly in the torch direction. If the resume operation function operates, return to the original stop position is performed even when relieve operation has been performed. As a result, the nozzle touch state is observed again. In such a case, the resume operation function needs to be kept enabled, but it should be disabled only after the execution of the resume program.

This can be performed with the RETURN_PATH_DSBL instruction. By using this instruction within the resume program, the resume operation function can be disabled only when the original program is resumed next. This instruction is valid only when it is executed within a resume program; the instruction is invalid when executed in a program other than the resume program.

9.11.3 Defining a resume program

The automatic error recovery function executes a resume program defined in an original program, in lieu of the original program. To define a resume program, use the RESUME_PROG= instruction. To erase the defined resume program, use the CLEAR_RESUME_PROG instruction. These instructions can be displayed on the edit screen by following the procedure shown below.

[INST] → Program control

In the example given in Fig–1, the following programs are used:

```
WELD_1                          JOINT  10 %
                                      1/7
    1:J P[1] 100% FINE
    2:  RESUME_PROG=WIRE_CUT
    3:L P[2] 100mm/sec FINE
     :  Arc Start[1]
    4:L P[3] 100mm/sec CNT100
    5:L P[4] 100mm/sec FINE
     :  Arc End[2]
    6: CLEAR_RESUME_PROG
    7:L P[5] 100mm/sec FINE
 [End]

POINT  ARCSTART  WELD_PT  ARCEND  TOUCHUP >
```

```
WELD_CUT                        JOINT  10 %
                                      8/8
    1:L P[10] 100mm/sec FINE
    2:J P[11] 100% CNT50
    3:  WO[4]=PULSE,0.5sec                     Wire feed
    4:L P[12] 20mm/sec FINE
    5:  WAIT    .80(sec)                       Wire cut
    6:L P[11] 20mm/sec FINE
    7:L P[10] 50% FINE
 [End]

POINT  ARCSTART  WELD_PT  ARCEND  TOUCHUP >
```

In the above program example, the WIRE_CUT program is taught in the second line of the WELD program and is erased in the sixth line. Since the WIRE_CUT program is defined as the resume program between the third to seventh lines, it is executed as the resume program. In the seventh and subsequent lines, the resume program has been erased, so the resume program is not executed.

The resume program is erased also when:

- Backward execution is performed.
- The cursor line is changed manually.
- The program terminates.

CAUTION

When the RESUME_PROG instruction is executed within the resume program, it is defined as a resume program for the original program.

9.11.4 Teaching the RETURN_PATH_DSBL instruction

The RETURN_PATH_DSBL instruction appears in the menu containing the RESUME_PROG instruction.

[INST] → Program control

The RETURN_PATH_DSBL instruction is valid only when it is taught within resume program instructions. Use this instruction as shown in the sample program given below. If the instruction is taught as shown below, the resume operation function does not operate when the original program resumes after the resume program terminates, even if the resume operation function is enabled.

```
WELD_1                              JOINT  10 %
                                          1/7
   1:J P[1] 100% FINE
   2:  RESUME_PROG=WIRE_CUT
   3:L P[2] 100mm/sec FINE
    :  Arc Start[1]
   4:L P[3] 100mm/sec CNT100
   5:L P[4] 100mm/sec FINE
    :  Arc End[2]
   6: CLEAR_RESUME_PROG
   7:L P[5] 100mm/sec FINE
[End]

POINT  ARCSTART  WELD_PT  ARCEND  TOUCHUP >
```

```
WELD_CUT                            JOINT  10 %
                                          8/8
   1:L P[10] 100mm/sec FINE
   2:J P[11] 100% CNT50
   3:  WO[4]=PULSE,0.5sec                   Wire feed
   4:L P[12] 20mm/sec FINE
   5:  WAIT    .80(sec)                     Wire cut
   6:L P[11] 20mm/sec FINE
   7:L P[10] 50% FINE
   8: RETURN_PATH_DSBL
[End]

POINT  ARCSTART  WELD_PT  ARCEND  TOUCHUP >
```

9.11.5 Setting the automatic error recovery function

On the setting screen of the automatic error recovery function, the following settings can be made:

- Enabling/disabling the automatic error recovery function
- Defining alarm codes to be monitored
- Defining the recovery switch SDI
- Defining the error recovery information SDO (indicating conditions for executing the resume program)
- Enabling/disabling the alarm–time automatic start feature
- Setting the maximum number of automatic start repetitions
- Setting the automatic start count register
- Enabling/disabling the fast exit/entry feature
- Enabling/disabling dry run exit/entry operation
- Defining a maintenance program
- Defining the maintenance SDO
- Defining automatic error recovery alarm conditions

MENUS → 6 Set up → Err recovery

```
Error Recovery Set     G1      JOINT  10 %
                                     1/12
 Error recovery function common setup
  1 Error recovery function:   DISABLED
  2 Approval DI index No.:          0
  3 Incomplete end DO index No.:    0
  4 Reset DI index No.:             0
  5 Automatic start feature:   DISABLED

 RESUME PROGRAM type recovery
  6 Status DO index No.:            0
  7 Auto start Max count:           2
  8 Auto start count R[] No.:       0

MAINTENANCE PROGRAM type recovery
  9 Fast exit/entry feature:   DISABLED
 10 Dry run exit/entry:        DISABLED
 11 Maintenance program:       ********
 12 MAINT DO index No.:             0

[ TYPE ]  ALARM DI_ALARM
```

To alarm registration screen

Alarm setting screen

```
Error Recovery Set     G1      JOINT  10 %
                                     2/10

  1 Monitored alarm code             53013
  2 Monitored alarm code             53018
  3 Monitored alarm code                 0
  4 Monitored alarm code                 0
  5 Monitored alarm code                 0
  6 Monitored alarm code                 0
  7 Monitored alarm code                 0
  8 Monitored alarm code                 0
  9 Monitored alarm code                 0

[ TYPE ]                 DONE      HELP
```

53013 indicates “ARC–013 Arc Start failed.”
53018 indicates “ARC–018 Lost arc detect.”

For alarm codes, refer to the alarm code table in the operator's manual.

Enabling/disabling the automatic error recovery function

This item enables or disables the automatic error recovery function. When the automatic error recovery function is enabled, and neither monitored alarm codes nor the recovery switch SDI are defined, the resume program is always executed at restart from the suspended state (except when the error recovery information SDO is off). When this item is disabled, the resume program is not executed.

Defining alarm codes to be monitored

To define alarm codes to be monitored, press the F2 (ALARM) key. A screen for defining alarm codes is displayed.

When a defined alarm code is issued, and a program is suspended, the resume program is executed at restart.

Each alarm code consists of an alarm code ID and alarm number. The alarm code ID indicates the type of alarm. For an arc start failure alarm, for example, the following alarm code is indicated:

```
   ARC    –    013      Arc Start failed  =  5  3   0  1
 ID (53)     Number                          ID    Number
```

For alarm numbers, refer to the alarm code table in the operator's manual.

Up to ten alarm codes can be defined as standard. To change the maximum number of alarm codes (up to 20 codes) that can be defined, change system variable $RSMPRG_SV.$NUM_ALARM, turn the power off then on.

Pressing the F5 (HELP) key displays the following screen:

```
Error Recovery Set    G1     JOINT  10 %
HELP      Arrows to scroll, PREV to exit

Typical alarm code IDs are specified as
follows.

   PROG :  3,   SRVO : 11,   INTP : 12
   PRIO * 13,   MOTN : 15,   SPOT : 23
   SYST : 24,   PALT : 26,   LASR : 50
   SEAL : 51,   ARC  : 53,   MACR : 57
   SENS : 58,   COMP : 59
```

CAUTION

Do not define any warning alarm as an alarm code.

Even when a defined alarm is issued, the resume program is not executed if the recovery switch DI is off. When there is no alarm code defined, that is, when all defined values are 0, the alarm code monitoring function is disabled.

The specifications of the alarm code monitoring function are listed below.

Table 9–9. Specifications of the Alarm Code Monitoring Function

ALARM CODE DEFINITION	ALARM CODE FUNCTION STATUS	ISSUANCE OF DEFINED ALARM	EXECUTION OF RESUME PROGRAM AT RESTART
All 0s	Disabled	———	Executed
At least one alarm code is defined	Enabled	Issued	Executed
		Not issued	Not executed

Defining the recovery switch SDI

To use the recovery switch SDI function, define an SDI number. After the number is defined, power must be turned off then back on.

With this function, the operator can choose whether to execute the resume program or not at the time of restart from the suspended state by using a peripheral device. If this signal number is not defined, this function is disabled.

The specifications of the recovery switch SDI function are listed below.

Table 9–10. Specifications of the Recovery Switch SDI Function

SDI NUMBER DEFINITION	RECOVERY SWITCH SDI FUNCTION STATUS	SDI STATUS	EXECUTION OF RESUME PROGRAM AT RESTART
0	Disabled	———	Executed
Valid number defined	Enabled	On	Executed
		Off	Not executed

> **CAUTION**
> Caution: To continue a resume program at program restart after the resume program is suspended, input the on state of the recovery switch SDI. If it is off, the original program is executed.

Defining the error recovery information SDO (conditions for executing the resume program)

When the alarm code monitoring function and recovery switch SDI function are both disabled, the resume program is always executed at the time of restart after the original program is suspended. When both the functions are enabled, it is difficult to determine whether the original program or resume program is to be executed at restart.

The error recovery information SDO is on only when the resume program is executed at restart. When the signal is off, the original program is executed at restart. With this function, the operator can know which program is to be executed next.

If the following conditions are met, the error recovery information SDO goes on:

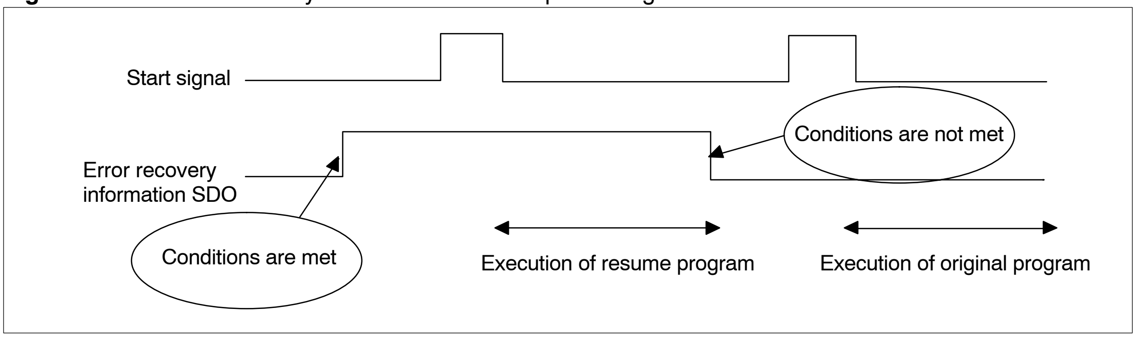
- The automatic error recovery function is enabled.
- The program to be executed is not in single step mode.
  The single step LED on the teach pendant indicates the single step status of a program currently selected (more precisely, the program set in $TP_DEFPROG). When the resume program is suspended, the error recovery information SDO is on even if the single step LED on the teach pendant lights. This is because the resume program to be executed is not in single step mode.
- The resume program is defined in the currently selected program (original program).
- The currently selected program (original program) has a motion group.
- The currently selected program (original program) is suspended, and the resume program is not yet completed.
- There is no optional function that disables the automatic error recovery function. See "Other specifications and restrictions."
- The user condition parameter ($AUTORCV_ENB) is true. See "Conditions for executing the resume program."
- When the teach pendant is enabled:
  - The operation mode (on the automatic error recovery manual operation screen) is TP_TEST.
- When the teach pendant is disabled:
  - The operation mode (on the automatic error recovery manual operation screen) is AUTO.
  - The remote conditions are met when system variable $RMT_MASTER is 0.
  - There is no alarm code defined. If any alarm code is defined, the alarm code is issued.
  - The recovery switch SDI function is disabled. If this function is enabled, the recovery switch SDI signal is on.

CAUTION

- The selected program means the program to be executed by inputting the start signal.
- While the resume program is being executed, single step operation cannot be performed.
- Even if the error recovery information SDO is on, the resume program is not executed when backward execution of the original program is performed.
- Backward execution in the resume program is possible.
- The update cycle period for the error recovery information SDO is 300 ms. When the conditions listed above have been changed, wait 300 ms before program execution.

The timing chart for the error recovery information SDO status and start signal is shown below.

Figure 9–26. Error Recovery Information SDO Output Timing Chart



Defining the incomplete end SDO

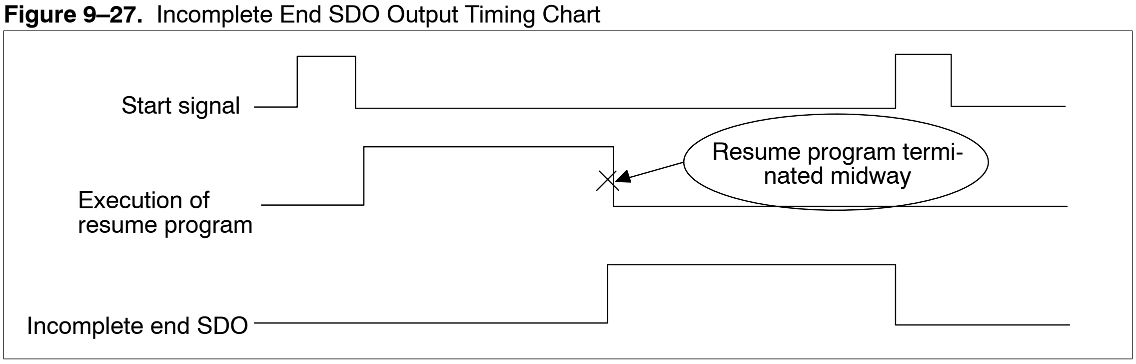
When an incomplete end SDO number is defined, the incomplete end SDO is output if a certain forced termination alarm is issued during execution of the resume program. The output incomplete end SDO is turned off by the next start signal input.

Before inputting the start signal, the operator must check the incomplete end SDO signal status. If this signal is on, the resume program terminates in the middle, so the robot is not in a specified position. In such a case, inputting the start signal causes the robot to perform resume operation to return from the current position to the stopped position of the original program, which may interfere with obstacles such as a jig. Therefore, before inputting the start signal, check the current robot position. If an interfering object exists, jog the robot to a position near the stopped position of the original program, then input the start signal.

This signal may be added to the PLC start signal acceptance conditions.

If this signal is set to 0, this function is disabled.

Figure 9–27. Incomplete End SDO Output Timing Chart



Defining the incomplete–end reset SDI

When the incomplete end SDO is included in the PLC start signal acceptance conditions, the operator requires a means to turn off the incomplete end SDO externally. Inputting the incomplete–end reset SDI signal turns off the incomplete end SDO. When the incomplete end SDO is output, the operator must first perform appropriate operation such as jogging the robot to near the stopped position of the original program, input the incomplete–end reset SDI, then input the start signal.

When this signal is set to 0, this function is disabled.

Enabling/disabling the alarm–time automatic start feature

When this item is enabled, and an alarm code to be monitored is defined, this feature functions if the defined alarm is issued. If a defined alarm is issued, this feature automatically executes the resume program. In this case, the alarm signal is not output. When the execution of the resume program has terminated, the original program is resumed automatically. During then, the operator need not input the start signal. Since the alarm signal is not output, other robots operating in the same line are not stopped.

CAUTION
Defined alarms must have the suspension alarm attribute.

CAUTION
If the error recovery information SDO is off when the resume program is executed automatically, alarm "INTP–135 Recovery DO OFF in auto start mode" is issued.

CAUTION
While the resume program is being executed, the UOP PAUSED signal is output. This is because the original program is in the suspended state. This specification is the same as that for multitasking systems.

WARNING
The alarm–time automatic start feature works on a program selected on the teach pendant. For example, suppose that program A having a resume program instruction has been executed from the teach pendant, then program B without a resume program instruction is selected and executed from the teach pendant. If an alarm defined in program A is then issued, the alarm signal is not output, but the resume program is not executed automatically. The reason for this is that the automatic start feature works on a selected program. In this example, program B that is currently selected does not define any resume program.

WARNING
When the automatic start feature item is enabled, and no monitored alarm code is defined, inputting the start signal for executing the resume program automatically executes the resume program then resumes the original program. In other words, when the start signal is input while the original program is suspended, the resume program is executed. As the resume program has terminated, the original program is then restarted automatically.

Setting the maximum number of automatic start repetitions

When a defined alarm is issued, the alarm–time automatic start feature automatically executes the resume program, then resumes the original program. If the defined alarm is issued again when the original program is resumed, the automatic start feature functions again. For example, the automatic start feature is activated by an alarm indicating an arc start failure, then the same alarm is issued again when the original program has resumed.

To prevent such an endlessly repeated condition, set the maximum number of automatic start repetitions.

The number of times the resume program is started repeatedly is counted internally. If the count exceeds the set value, "INTP–134 Over automatic start Max counter" is issued, and the error recovery information SDO is turned off at the same time. If this occurs, eliminate the cause of the alarm issued in the original program. Then input the start signal.

CAUTION
The number of repetitions counted internally is cleared when the execution of a move statement has terminated and when the CLEAR_RESUME_PROG instruction has been executed.

Defining the automatic start count register

As mentioned above, the resume program may be executed several times repeatedly by the automatic start feature. When the automatic start count register is defined, a different program can be executed as the resume program each time the resume program is executed. For example, when the resume program is executed for the first time by the automatic start feature, the register value is 1. When an alarm is issued again during execution of the original program, and the resume program is then executed again by the automatic start feature, the register value is 2. By executing a different subprogram in the resume program according to the register value, different resume program operation can be performed each time the repetition count is incremented.

> **CAUTION**
> When the resume program is executed by other than the automatic start feature, the register value is 0. Therefore, a resume program must be created so that the same subprogram is called when the register value is 0 and when the value is 1.

Enabling/disabling the fast exit/entry feature

If an alarm is issued during operation in a complicated environment, the robot moves from the stopped position to the taught point to execute the resume program. In this case, the robot may interfere with part of a workpiece or peripheral devices. After recovery operation, similar interference may occur when an attempt is made to execute the original program. The fast exit/entry feature is provided to avoid the possibility of such interference. The feature can be enabled or disabled by setting this item.

The fast exit/entry feature causes the following operation automatically:

1 From the stopped position, disable arc welding, and execute only the move statements of the original program up to the end.
2 Execute a maintenance program.
3 Disable arc welding, execute the move statements of the original program from the beginning to move the robot to the stopped position.
4 Enable arc welding, and resume the original program operation.

Even when this feature is enabled, the resume program takes priority over this feature if the resume program is enabled in the original program. In other cases, the maintenance program is executed.

Enabling/disabling dry run exit/entry

In the fast exit/entry feature, this item specifies whether exit from the stopped position and return to the stopped position after maintenance program execution are to be performed at dry run speed.

Defining a maintenance program

Define the name of a maintenance program used as the standard maintenance program. The maintenance program name can also be specified using the maintenance program instruction on the edit screen.

Defining the maintenance SDO

Define the number of the SDO for indicating that the fast exit/entry feature is operating.

Defining automatic error recovery alarm conditions

Define the conditions for issuing an automatic error recovery alarm on the definition screen that is displayed by pressing F3 (DI_ALARM) on the setting screen of the automatic error recovery function.

```
Error Recovery Set     G1      JOINT  10 %
                                    1/3
    UALM    Severity   Type       Value
  1 [  1]   LOCAL      DI[  1]     ON
  2 [  5]   GLOBAL     RI[  2]     OFF
  3 [ 10]   LOCAL      DI[  5]     ON

PLEASE POWER OFF AFTER CHANGING DI/DO
[ TYPE ]                  DONE    HELP
```

On this screen, the items shown below can be set. The alarm code of the automatic error recovery alarm is 12278.

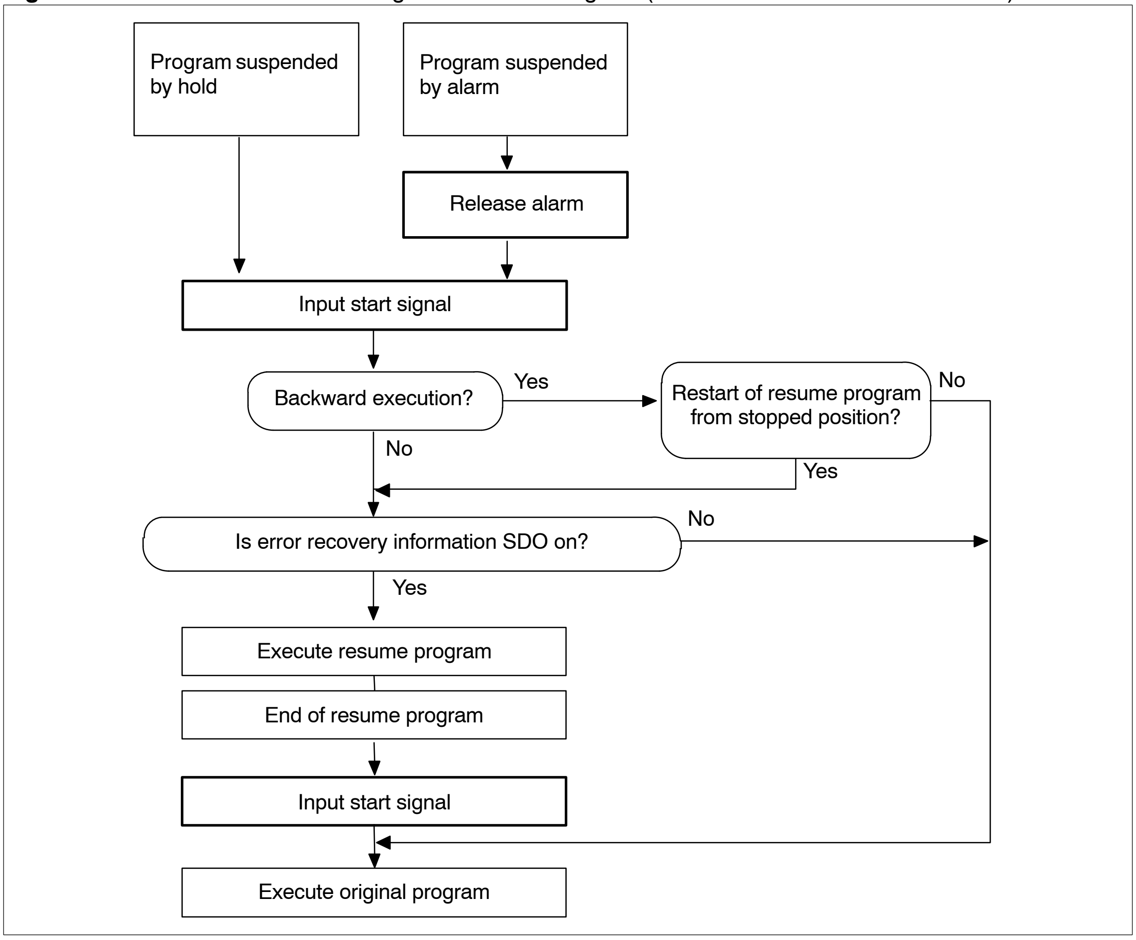
- User alarm number
  When the automatic error recovery alarm is issued, the user alarm message with the set number is displayed as an alarm message. When this item setting has been changed, the new setting becomes effective immediately.
- Alarm severity
  This item can choose whether the automatic error recovery alarm is a local alarm or global alarm. When LOCAL is set, the automatic error recovery alarm is issued only for the program that defines a resume program. If there is no program that defines a resume program, the alarm is regarded as a global alarm. If the automatic error recovery alarm is issued when there is no program being executed, a warning is generated. When this item setting has been changed, the new setting becomes effective immediately.
- Signal type
  Choose the type of the digital signal for issuing the automatic error recovery alarm from among DI, RI, and WI. When this item setting has been changed, the power must be turned off then back on for the new setting to become effective.
- Signal number
  Set the number of the digital signal for issuing the automatic error recovery alarm. When this setting has been changed, the power must be turned off then back on for the new setting to become effective.
- Detection signal status
  Set the status of the digital signal for issuing the automatic error recovery alarm to ON (high) or OFF (low). When this setting has been changed, the power must be turned off then back on for the new setting to become effective.

The standard number of automatic error recovery alarm conditions is three. This number can be increased to up to five by changing system variable $RSMPRG_SV.$NUM_DI_ALM. After this system variable has been changed, the power must be turned off then back on for the new setting to become effective.

9.11.6 Flowchart for resuming a suspended program

The resume program is executed according to the following flowchart:

Figure 9–28. Flowchart for Executing the Resume Program (When Automatic Start is Disabled)



CAUTION

- When forward execution is specified while the original program is suspended, the resume program is executed if the error recovery information SDO is on; if it is off, the original program resumes.
- When forward execution is specified while the resume program is suspended, the resume program resumes if the error recovery information SDO is on; if it is off, the original program is executed.
- When backward execution is specified while the original program is suspended, backward execution is performed for the original program without executing the resume program.
- When backward execution is specified while the resume program is suspended, backward execution is performed for the resume program.

9.11.7 Manual operation screen of the automatic error recovery function

A manual operation screen is supported for the automatic error recovery function. This screen contains the following:

- Error recovery information SDO status
- Name of the resume program defined in the currently selected program
- Operation mode setting
- Detail information about the error recovery information SDO

This screen can be selected by following the procedure shown below.

MENUS → MANUAL FCTNS → Err recorery

```
Error Recovery MNF     G1      JOINT   10 %
                                      1/1

 Error recovery DO status:           OFF
 Defined resume program:         WIRE_CUT

  1 Operation mode:              AUTO

[ TYPE ] DATAIL         [CHOICE]
```

↓

```
Error Recovery MNF     G1      JOINT   10 %
                                      1/11
  1 Auto error recovery enabled:  No
  2 PAUSED & resume prog incomp:  No
  3 Program has motion group:     No
  4 Not in single step mode:      No
  5 Resume program is defined:    No
  6 Mode is(TP_TEST):             No
  7 Approval DI is ON:            None
  8 Defined alarm occurs:         None
  9 Remote when $RMT_MASTER is 0: None
 10 No disabled options:          No
 11 User condition param enable:  Yes

[ TYPE ]                          DONE
```

Error recovery information SDO status

The error recovery information SDO status is indicated. Even when the error recovery SDO is not defined, its status can be indicated. From this information, the operator can know which program, the resume program or original program, is to be executed.

Defined resume program

The name of the resume program defined in the currently selected program is indicated. From this information, the operator can check whether a wrong resume program is defined or not.

> **CAUTION**
>
> If a wrong program is defined as the resume program, the robot operation is unpredictable. Therefore, check that the resume program is correct.

Operation mode

There are three operation modes. The standard setting is AUTO. When the display changes from this screen to another, AUTO is automatically set again.

- AUTO
  This mode should be set when the teach pendant is disabled. When this mode is selected, the resume program is executed according to the status of the alarm code monitoring function and recovery switch SDI function. If this mode is selected when the teach pendant is enabled, the resume program is not executed.
- NO_EXEC
  When this mode is selected, the error recovery information SDO is always off. Therefore, in this mode, the resume program is not executed.
- TP_TEST
  This mode should be set when the teach pendant is enabled. When this mode is selected, and when the teach pendant is enabled, the resume program is always executed regardless of the status of the alarm code monitoring function or error recovery switch SDI function.

Displaying detail conditions of the error recovery information SDO

When F2 (DETAIL) is pressed on the manual operation screen of the automatic error recovery function, detail conditions related to the error recovery information SDO status are displayed. When all items on the detail screen are set to Yes or None, the error recovery information SDO is turned on. When the error recovery information SDO is off, and you cannot find the cause of the SDO being off, check this screen.

- Auto error recovery enabled
  This item indicates whether this function is enabled or disabled on the setting screen of the automatic error recovery function.
- PAUSED & resume prog incomp
  This item indicates the following conditions:
  - The selected program must exist.
  - The selected program must be in the suspended state.
  - A resume program must be defined in the selected program, and the execution of the resume program must not have been completed.
- Program has motion group
  This item indicates that the selected program has a motion group.
- Not in single step mode
  This item indicates that the single step mode is not set.
  The single step LED on the teach pendant indicates the single step status of the selected program ($TP_DEFPROG). Even when the single step key is pressed while the resume program is suspended, and the single step LED goes on, the error recovery information SDO is held on. This is because the selected program is the original program, and the LED indicates that the original program is in single step mode; the resume program is not in single step mode.
- Resume program is defined
  This item indicates that a resume program is defined in the selected program.
- Mode is (xxxx)
  This item indicates that the operation mode is suitable for the current status. For example, when the teach pendant is disabled, “AUTO” is indicated in the portion “xxxx.” When the teach pendant is enabled, “TP_TEST” is indicated.
- Approval DI is ON
  This item indicates the recovery switch DI status. When the SDI number is not defined, or when the teach pendant is enabled, “None” is indicated.
- Defined alarm occurs
  This item indicates that an alarm code is defined, and that alarm is issued. When no alarm code is defined, or when the teach pendant is enabled, “None” is indicated.
- Remote when $RMT_MASTER is 0
  This item indicates that remote conditions are met. This function is enabled only when the teach pendant is disabled, system variable $RMT_MASTER is 0, and system variable $RSMPRG_SV.$CHK_REMOTE is true.

- No disabled options
  There are options that cannot be used together with the automatic error recovery function. This item indicates whether such options are present or not.
- User condition param enable
  This item indicates the status of system variable $AUTORCV_ENB for user conditions. For how to use this system variable, see "Changing conditions for executing the resume program."

9.11.8 Execution of the resume program from the teach pendant and test mode

Normally, the automatic error recovery function is used when production is started with the teach pendant disabled. When checking the resume program during teaching, set the operation mode to TP_TEST on the manual operation screen. In TP_TEST mode, the resume program can be executed regardless of the recovery switch DI status and whether a defined alarm is issued or not.

9.11.9 Changing conditions for executing the resume program

To use resume program execution conditions other than alarm codes, use user condition system variable $AUTORCV_ENB and the status monitoring function (A05B–2400–J628).

For example, to execute the resume program when R[1] is 1, create the following monitor program, and start MONIT1.CH on the system monitor screen.

```
MONIT1.CH
   1:  WHEN R[1]=1,CALL DO_RESUME
   2:  WHEN R[1]<>1,CALL NO_RESUME
```

```
DO_RESUME. TP
   1:  $AUTORCV_ENB=1
   2:  MONITOR MONIT_3
```

```
MONITI2.CH
   1:  WHEN R[1]=1,CALL DO_RESUME
```

```
DO_RESUME. TP
   1:  $AUTORCV_ENB=0
   2:  MONITOR MONIT_2
```

```
MONITI3.CH
   1:  WHEN R[1]<>1,CALL NO_RESUME
```

The start conditions can be changed by modifying the monitor program. For how to use the status monitoring function, refer to the operator's manual on the status monitoring function. In this case, the automatic start function is unavailable.

9.11.10 Other specifications and restrictions

- While the resume program is being executed, single step operation is not performed. Single step mode is valid only for the original program.
- When the cursor line is changed and executed while the original program is suspended, the resume program is not executed.
- While the resume program is being executed, the resume program execution status cannot be checked on the program edit screen. On the edit screen, the suspended original program is displayed.
- When a multitasking program (a main program and subprogram) is being executed with the alarm code monitoring function disabled and the recovery switch DI undefined, pressing the hold button causes both the main program and subprogram to stop. Suppose that a resume program is defined in the subprogram, but that no resume program is defined in the main program. In this case, when the main program is selected and re–executed, the resume program for the subprogram is not executed. This is because the selected program is the main program, and the error recovery information DO is off.

In this case, when the subprogram is selected and executed, the error recovery information DO is turned on, so the resume program is executed.

CAUTION

When using the automatic error recovery function in multitasking systems, define a resume program only in the main program. Even when a resume program is defined in a subprogram, that resume program cannot be executed.

Definitions in the main program and subprogram are shown below.

Main.TP

```
1: RUN Sub
2: RESUME_PROG=WIRECUT
```

Sub.TP

```
No resume program defined
```

- While the resume program is being executed, the suspended original program is displayed on the edit screen.
- When the cursor line in the original program is moved while the resume program is suspended, then program re–execution is performed, the resume program is resumed. After the resume program terminates, specifying program execution displays a popup menu confirming the cursor movement. When Yes is entered in response, the original program is executed starting from the new cursor line.
- In a single task system, when the resume program is suspended, selecting a program other than the original program on the program directory screen causes the original program to terminate.
- Never teach the arc and weaving instructions in the resume program. If an arc instruction is executed in the resume program while arc welding is being performed by the original program, alarm "ARC–034 Task does not control welding" is issued. In addition, weaving operation is not performed within the resume program.
- The automatic error recovery function supports the power failure handling function.
- The automatic error recovery function is disabled when one of the following options is loaded:
  - Arc sensor
  - AVC (TIG arc length control)
  - MIG EYE option
  - Root path memorization
  - Line tracking
  - Soft float
  - Continuous turn
  - Coordinated motion
  - Remote TCP
  - Accurate path function
  - Constant joint path function (path not overridden)
  - Multi robot control

9.11.11 Warnings

When using the automatic error recovery function, observe the following safety precautions:

- If a wrong program or a program causing wrong operation is defined as a resume program, the robot moves in a direction the operator cannot predict. Define a correct program.
- Before inputting the start signal and before pressing the execution key on the teach pendant, for safety, check the error recovery information DO status to confirm whether the original program or resume program is to be started.
- If the operation mode is set to TP–TEST on the manual operation screen of the automatic error recovery function, the resume program is started even when a defined alarm is not issued or when the recovery switch DI is off.
- When an operation mode other than AUTO is set on the manual operation screen of the automatic error recovery function, then the display is changed to another screen, the operation mode is set to AUTO again automatically. To use an operation mode other than AUTO, always keep displaying the manual operation screen of the automatic error recovery function.

9.12 HIGH–SENSITIVITY COLLISION DETECTION

9.12.1 Overview

High–sensitivity collision detection is intended to quickly detect when the tool or robot collides with the workpiece and to stop the robot.

This function has greatly been improved in detection sensitivity from the conventional collision detection function. It can detect a collision more quickly, thus greatly reducing possible damage to the tool and robot itself.

This high–sensitivity collision detection function makes unnecessary shock sensors and similar devices that have conventionally been used to protect the hand of the robot.

9.12.2 Specification

1) When a collision is detected, the function issues an alarm and stops the robot quickly by decelerating it in such a way that shocks to the robot can be decreased.
2) With the function, the user need not adjust detection sensitivity, which has previously been adjusted for an individual robot.
3) Program instructions can be used to enable/disable the function.
4) The function increases its detection sensitivity automatically during a teach operation, making it possible to reduce the possible damage that may occur due to an incorrect manipulation especially during a teach operation, during which it is likely that the robot is caused to collide with a workpiece.

9.12.3 High–Sensitivity Collision Detection

(1) Overview

- The function is enabled the moment the power is turned on.
- Set the function with load information and the information about devices installed on the robot.
  Since the function uses the load information and device information to detect a collision, it is necessary to set the function with these pieces of information. Be sure to specify the weight of the load, the position of its gravity center, and the weight of each device on the robot accurately.
  If the inertia (shape) of the load is large, it may be necessary to specify the inertia around the gravity center of the load. (If the tool is big, and simply specifying its weight and gravity center does not assure accurate detection, specify its inertia.)
  See Section 9.14, “Load Setting,” or Section 9.15, “Load Estimation,” for how to specify load information.
- If it is previously anticipated that a strong force will be exerted during an operation, disable the function for that operation, using program instructions. (See (2).)
- The function increases its detection sensitivity automatically during a teach operation.

(2) Program instructions

- COL DETECT ON / OFF

  This instruction can enable/disable collision detection during program execution.

Example

```
10: J P[1] 100% FINE
11:  COL DETECT OFF
12: L P[2] 2000mm/sec CNT100
13: L P[3] 2000mm/sec CNT100
14: L P[4] 2000mm/sec CNT100
15:  COL DETECT ON
16: J P[2] 50% FINE
```

This program disables collision detection with lines 12 to 14.

CAUTION

Collision detection is usually enabled.

When the program ends its operation or is aborted, collision detection is enabled automatically.

9.12.4 Cautions

(1) Under the following conditions, the function may detect a collision when it should not:

- The load or device information is incorrect.
- The weight or inertia of the load exceeds the capacity of the robot.
- The power supply voltage is too low.
- Heavy work caused by using acceleration override
- Heavy work such as reversing based on smooth interpolation
- Linear operation near a cardinal point where the axis is subjected to high–speed rotation

Action: If a collision is detected when it should not because of any of these causes, first try to remove the cause. If it cannot be removed, enclosing the portion that results in an incorrect detection between COL DETECT ON / OFF instructions may be able to avoid an alarm and stop.

(2) Collision detection is disabled under the following conditions:

- A softflow is enabled.
- Brake control is in effect (the brake is locked).

(3) Axis drop after detection

- To reduce the force caused in a collision to the robot, the collision detection function keeps position control disabled for 200 ms after the collision. So, a slight axis drop may occur after the collision is detected.

9.13 LOAD SETTING

9.13.1 Overview

Setting the information about the load on the robot appropriately can cause the following effects:

- Increase in motion performance (such as lower vibration and shorter cycle time)
- More effective reaction of functions related to dynamics (such as increase in performance related to collision detection and gravity compensation)

For effective use of the robot, it is recommended to appropriately set information about loads such as the hand, workpiece, and devices mounted on the robot.

A load estimation function is optionally available. This function enables the robot to calculate load information automatically.

9.13.2 Motion Performance Screens

There are three motion performance screen types: List screen, load setting screen, and device setting screen. They are used to specify load information and the information about devices on the robot.

These screens let you easily specify the information that has conventionally been set in system variables ($PAYLOAD, $PAYLOAD_X, $PAYLOAD_Y, $PAYLOAD_Z, $PAYLOAD_IX, $PAYLOAD_IY, and $PAYLOAD_IZ in the $PARAM_GROUP). They also let you switch the load setting among two or more loads.

1 Press MENUS to display the screen menu.
2 Select “6 SETUP” described on the next page.
3 Press F1 (TYPE) to display the screen switch menu.
4 Select Motion. The list screen appears. (If any other screen appears, press [PREV] several times until the list screen appears.) For a multigroup system, the list screen of another group can be reached by pressing F2 (GROUP).

```
MOTION PERFORMANCE           JOINT  10 %
    Group1
 No.    PAYLOAD[kg]          Comment
  1            0.00    [                ]
  2            0.00    [                ]
  3            0.00    [                ]
  4            0.00    [                ]
  5            0.00    [                ]
  6            0.00    [                ]
  7            0.00    [                ]
  8            0.00    [                ]
  9            0.00    [                ]
 10            0.00    [                ]

Active PAYLOAD number = 0
[ TYPE ]  GROUP  DETAIL ARMLOAD SETIND >
```

5 Load information can be specified for condition No. 1 to No. 10. As stated later, an appropriate condition number can be selected as the load is changed by a hand change.
Move the cursor to the desired No., and press F3 (DETAIL) to display the related load setting screen.

```
MOTION/PAYLOAD SET           JOINT  10 %

    Group 1
  1 Schedule No [  1]:[****************]
  2 PAYLOAD            [kg]          0.00
  3 PAYLOAD CENTER X   [cm]          0.00
  4 PAYLOAD CENTER Y   [cm]          0.00
  5 PAYLOAD CENTER Z   [cm]          0.00
  6 PAYLOAD INERTIA X  [kgfcms^2]    0.00
  7 PAYLOAD INERTIA Y  [kgfcms^2]    0.00
  8 PAYLOAD INERTIA Z  [kgfcms^2]    0.00

[ TYPE ]  GROUP  NUMBER  DEFAULT  HELP
```

6 Specify the mass and gravity center of the load, and inertia around its gravity center.
The X, Y, and Z directions displayed on the load setting screen are in reference to the default tool coordinate system (which is valid when no other tool coordinate system is set up).

NOTE 1 [kgf cm s^2] = 980 [kg cm^2]

When a value is entered, the confirmation message Path and Cycletime will change. Set it? appears. Press F4 (YES) or F5 (NO) whichever is necessary.

7 Pressing F3 “NUMBER” lets you go to the load setting screen for another condition No. For a multigroup system, pressing F2 “GROUP” lets you move to the setting screen of another group.

8 Press PREV to go back to the list screen. Press F5 “SETIND”, and enter a desired load setting condition No. The last condition No. selected is used during program execution and jog operation. (The initial condition No. is 0. Using the condition without changing from the initial setting causes the initial system variable setting to be used. Using the setting on the load setting screen requires enabling that setting.)

9 Pressing F4 “ARMLOAD” on the list screen lets you move to the device setting screen.

```
MOTION/ARMLOAD SET             JOINT  10 %

    Group 1
  1 ARM LOAD AXIS #1  [kg]          0.00
  2 ARM LOAD AXIS #3  [kg]          0.00

[ TYPE ]  GROUP          DEFAULT  HELP
```

10 Specify the mass of the devices on the J1 and J3 arms.
Entering values displays the message Path and Cycletime will change. Set it?. Press F4 (YES) or F5 (NO) whichever is necessary.
After setting the mass of a device, turn the power off and on again.

9.13.3 Program Instructions

Pressing F5 “SETIND” on the list screen lets you switch the screen, using program instructions rather than selecting a desired load setting condition No. (Even after program execution is finished, the last condition No. selected is used during later program execution and jog operation.)

```
Instruction                    JOINT  10 %
 1 Miscellaneous     5 Multiple control
 2 Skip              6 SENSOR
 3 Payload           7 Program control
 4 Offset/Frames     8 ---next page---
PRG
```

(1) Additional setting [i]

This instruction changes the load setting condition No. to be used to i.

```
PAYLOADstatement               JOINT  10 %
 1 PAYLOAD[...]      5
 2                   6
 3                   7
 4                   8
PRG
```

Example

```
1:  Additional setting [i]
```

This program selects load setting condition 1.

Multi–operation group environment

The PAYLOAD[i] instruction usually selects a load setting condition No. for all operation groups enabled for the program. For a multigroup system, however, it is possible for this instruction to specify what group to be subjected to load setting condition No. switching.

```
PRG                                    JOINT  10 %
                                                1/2




    1:  PAYLOAD[...]
 [End]





Enter value
 GROUP  DIRECT INDIRECT
```

Pressing F1 (GROUP) displays a menu that contains choices for specifying a group. You can select a group from the menu.

Example

PAYLOAD[i]
This program selects load setting condition No. 1 for groups 2 and 3.

9.14 COLLISION DETECTION for AUXILIARY AXIS

9.14.1 General

The Collision Detection Function is the feature that stops the robot immediately and reduces the damage to the robot, when the robot collides with other objects.

Generally, this feature has been applied for the robot axes.

But, this feature has not been applied for the auxiliary axis. Because of the auxiliary axis is design by customer, then the parameters for this feature can not be set beforehand.

To apply this feature to the auxiliary axis, the parameter tuning is required with tuning procedure on this manual.

NOTE

To tune the collision detection parameters for auxiliary axis,

Collision Detection for Auxiliary Axis Option (A05B–2400–J645)

or

High Sensitive Collision Detection Package Option (A05B–2400–J684) that includes above is required.

9.14.2 CAUTION

The load ratio of auxiliary axis should be less than 5.

Load ratio = (Load Inertia + Motor Inertia) / Motor Inertia

When the auxiliary axis is designed, you must consider above.

If the load ratio of auxiliary axis is more than 5 times, the motion performance and sensitivity for collision detection may deteriorate.

9.14.3 INITIAL SETTING

1 Setup auxiliary axis (Gear ratio, acceleration time, and etc.) normally

2 Turn power on

3 Set the following system variables

$SBR[n].$PARAM[112] = 2097152 / ($SBR[n].$PARAM[47])
$SBR[n].$PARAM[119] = 7282
$SBR[n].$PARAM[120] = –7282

n : Hardware axis number of auxiliary axis
n=7~ for aux. Axis / n=1~6 for robot axes

4 Cycle power

9.14.4 TUNING PROCEDURE

The sensitivity of collision detection will be tuned by below procedure. It should be tuned without mis–detection.

1 Create the program that includes heavy motion like an inverse motion with CNT100 beforehand.
If the program for production is already exist, It can be used to tune. In this case, the sensitivity can be optimized for production with this program.
(However, if other program was run, the mis–detection might occur. Also, if this program was modified, the re–tuning might be required.)

2 Run the above program.
Also this program must not be paused.
Because of the disturbance torque, see below, will be cleared at just re–start the program

3 Measure the max. / min. disturbance torque on STATUS/AXIS/DISTURB screen after running the program.

```
 PNS001          Line 1              ABORT
 STATUS Axis       G1            FINE 100%

     Disturbance Torque (A)
     Curr./ Max.(Allowed)/Min.(Allowed)
 J1 :     0.0  20.0(  40.0)  -19.0( -40.0)
 J2 :     0.0  19.0(  40.0)  -20.0( -40.0)
 J3 :     0.0  22.0(  40.0)  -10.0( -40.0)
 J4 :     0.0  12.0(  20.0)   -5.0( -20.0)
 J5 :     0.0  10.0(  20.0)  -11.0( -20.0)
 J6 :     0.0   8.0(  20.0)   -4.0( -20.0)
 J7 :     0.0  24.0(  56.0)  -30.0( -56.0)

[TYPE]  MONITOR  TRACKING  DISTURB [UTIL] >
```

As said above, the disturbance torque will be reset at the start of each program.
If there are some programs,

- make new program that call all program for tuning and run this main program.
- record the max. / min. disturbance torque for each programs and find max. / min. value in these recorded value.

4 Move the cursor to allowed value in parentheses for the axis.
Change the allowed value to same as measured max. or min. value.

```
 PNS001          Line 1              ABORT
 STATUS Axis       G1            FINE 100%

     Disturbance Torque (A)
     Curr./ Max.(Allowed)/Min.(Allowed)
 J1 :     0.0  20.0(  40.0)  -19.0( -40.0)
 J2 :     0.0  19.0(  40.0)  -20.0( -40.0)
 J3 :     0.0  22.0(  40.0)  -10.0( -40.0)
 J4 :     0.0  12.0(  20.0)   -5.0( -20.0)
 J5 :     0.0  10.0(  20.0)  -11.0( -20.0)
 J6 :     0.0   8.0(  20.0)   -4.0( -20.0)
 J7 :     0.0  24.0(  24.0)  -30.0( -30.0)

[TYPE]  MONITOR  TRACKING  DISTURB [UTIL] >
```

CAUTION

When the disturbance torque exceeds above allowed value, the following *WARNING* occurs

SRVO–053 Disturbance excess (G:x,A:x)

Following servo alarm (servo power off) occurs when the disturbance torque exceeds below *ALARM LEVELs*.

Upper Limit = Max. allowed value + 0.3 × Max. current of amp.

Lower Limit = Min. allowed value – 0.3 × Max. current of amp.

SRVO–050 Collision Detect alarm (G:x,A:x)

Part of 0.3×Max. current of amp. is the margin to prevent the mis–detection.

For example in above screen with 40A amplifier,

Upper Limit = 24.0 + 0.3 × 40 = 36 A

Lower Limit = –30.0 – 0.3 × 40 = –42 A

5 Run the programs again with above disturbance allowed setting, and confirm that there is no mis–detection.

6 Finished

9.15 Gravity Compensation

Gravity compensation calculates the bending of the robot arm caused by the tool/work on the flange, the equipment on the arm, and the self weight of the arm. Then it compensates the motor position depending on the calculation of the bending, and it improves the absolute position accuracy.

Gravity Compensation option (A05B–2350–J649) is necessary to use this function.

This function can not be used with Softfloat (A05B–2350–J612) or Small Circle (not supported).

9.15.1 System Variables

Gravity Compensation

$PARAM_GROUP[group].$SV_DMY_LNK[8] FALSE

BOOLEAN RW PU TRUE/FALSE

[Name] Gravity Compensation Enable/Disable

[Meaning] TRUE Gravity Compensation Enable
FALSE Gravity Compensation Disable Gravity compensation is disabled when the robot is shipped. Please set this variable to TRUE and cycle power before use.
To set back to be disabled, set this variable to FALSE, do controlled start, and execute robot setup again. (By doing that, the motion parameters are set back to the default values. If the motion parameters have been modified, they need to be modified again.)

$PARAM_GROUP[group].$MOUNT_ANGLE 0

REAL RW PU –100000 ~ 100000 (deg)

[Name] Mount Angle of Robot

[Meaning] Set 0deg for floor mount type, 180deg for upside down type, or the mount angle for wall mount or angle mount type. Cycle power after setting.

9.15.2 MOTION Screen

1 Payload and armload (equipment on the arm) parameters are set in this screen.
2 This setting screen has three sub–screens. (MOTION screen / PAYLOAD SET screen / ARMLOAD SET screen)
3 This screen is sub–screen in SYSTEM.

MOTION Screen (Default screen)

```
MOTION                                JOINT  100%
    Group 1
No.          PAYLOAD[kg]          Comment
  1            100.00       [           ]
  2            120.00       [           ]
  .              .                  .
  .              .                  .
 10            120.00       [           ]

Active PAYLOAD number = 1
[ TYPE ]  GROUP  DETAIL  ARMLOAD  SETIND >
```

4 Payload information (Schedule No.1 to 10) can be setup. Move cursor to the line of one of the schedule numbers, and press F3(DETAIL) to enter the payload set screen.

PAYLOAD SET Screen

```
MOTION/PAYLOAD/SET                JOINT    100%
     Group 1
 1.  Schedule  No [   1] : [Comment      ]
 2.  PAYLOAD               [kg]      100.00
 3.  PAYLOAD CENTER X      [cm]       10.00
 4.  PAYLOAD CENTER Y      [cm]        0.00
 5.  PAYLOAD CENTER Z      [cm]       10.00
 6.  PAYLOAD INERTIA X     [kgfcms^2] 0.00
 7.  PAYLOAD INERTIA Y     [kgfcms^2] 0.00
 8.  PAYLOAD INERTIA Z     [kgfcms^2] 0.00

[ TYPE ]  GROUP  NUMBER  DEFAULT  HELP
```

5 Setup the payload, payload center, and payload inertia. X, Y, and Z directions in this screen mean X, Y, and Z axes of the default (the settings are all 0) tool frame.

After the value is input, the message “Path and Cycletime will change. Set it ?” is displayed. Please input F4(YES) or F5(NO).

6 To enter the payload set screen of the other schedule number, press F3(NUMBER). To enter the screen for other group, press F2(GROUP). (Only in the multi–group system)

7 Press PREV key to go back to the motion screen (default screen). Press F5(SETIND) and input the schedule number to use.

8 Press F4(ARMLOAD) in the motion screen (default screen) to enter the armload set screen.

ARMLOAD SET Screen

```
MOTION/ARMLOAD/SET                JOINT    100%
     Group 1
 1.  ARM LOAD AXIS #1      [kg]       20.00
 2.  ARM LOAD AXIS #3      [kg]       10.00

[ TYPE ]  GROUP          DEFAULT  HELP
```

9 Setup the armload on axis #1 and axis #3.

After the value is input, the message “Path and Cycletime will change. Set it ?” is displayed. Please input F4(YES) or F5(NO).

After setting up the armload, cycle power.

10. PALLETIZING FUNCTION

This chapter explains the palletizing function.

☐ Contents of this chapter

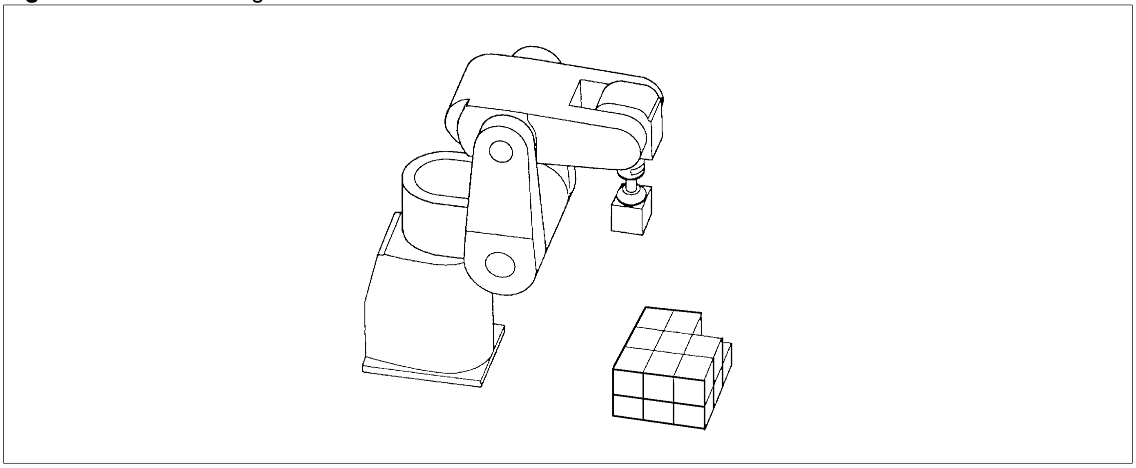
10.1 Palletizing Function

Palletizing function

Palletizing is a function for the orderly stacking of workpieces by only teaching several representative points.

- A stacking pattern can be created easily by teaching representative stack points.
- A path pattern can be created by teaching path points (approach points and retraction points).
- Multiple path patterns can be set to perform palletizing in a wide variety of patterns.

Figure 10–1. Palletizing

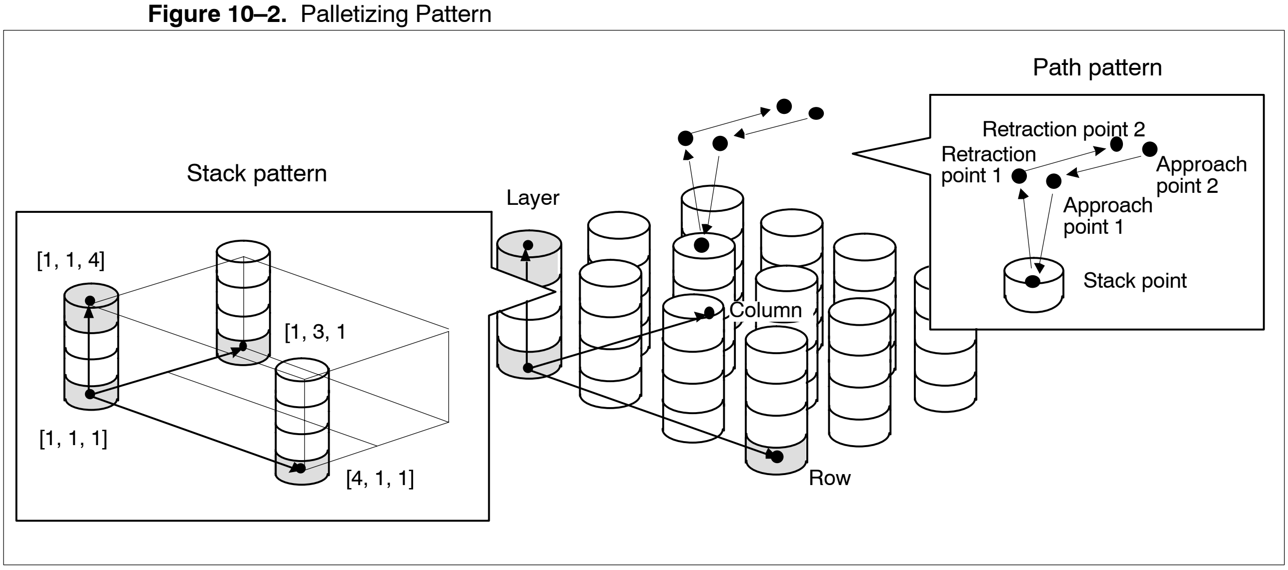


Structure of the palletizing function

The palletizing function consists of the following two patterns:

- Stacking pattern : Determines the method of stacking workpieces.
- Path pattern : Determines the path along which the robot hand moves to stack workpieces.

Figure 10–2. Palletizing Pattern



Types of palletizing

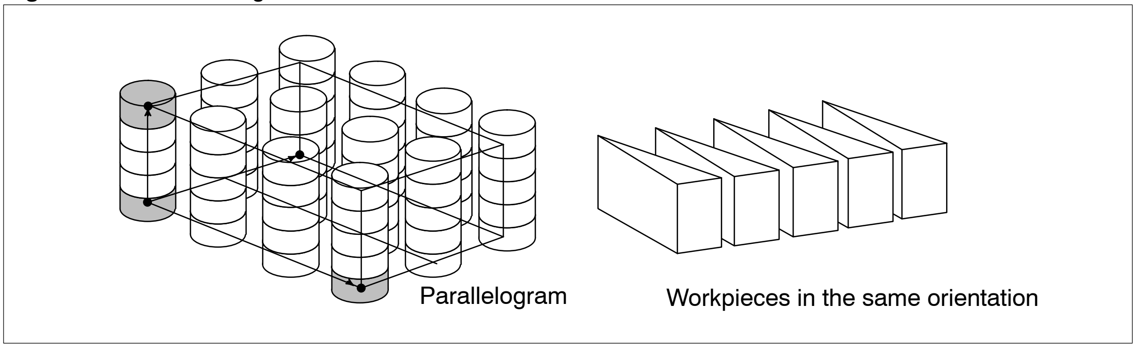
There are the following four types of palletizing according to the methods for setting stack and path patterns (See Section 10.3).

- Palletizing B and palletizing BX
- Palletizing E and palletizing EX

Palletizing B only the palletizing–B function can be taught.

Palletizing B can achieve a comparatively easy stacking pattern by one kind of path pattern. In R–J2 Mate, an easy attitude control can be done.

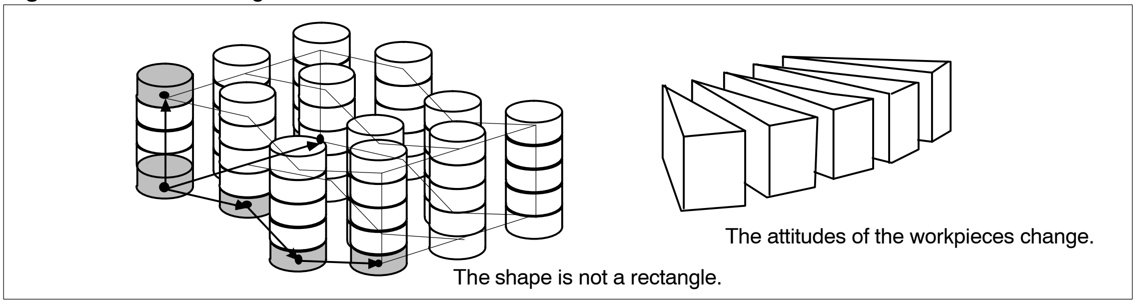
Figure 10–3. Palletizing–B Function



Palletizing E

Palletizing E can be used for more complex stack patterns (such as when the attitudes of workpieces are to be changed or when the shape made by the stacked workpieces, as viewed from below, is not a parallelogram).

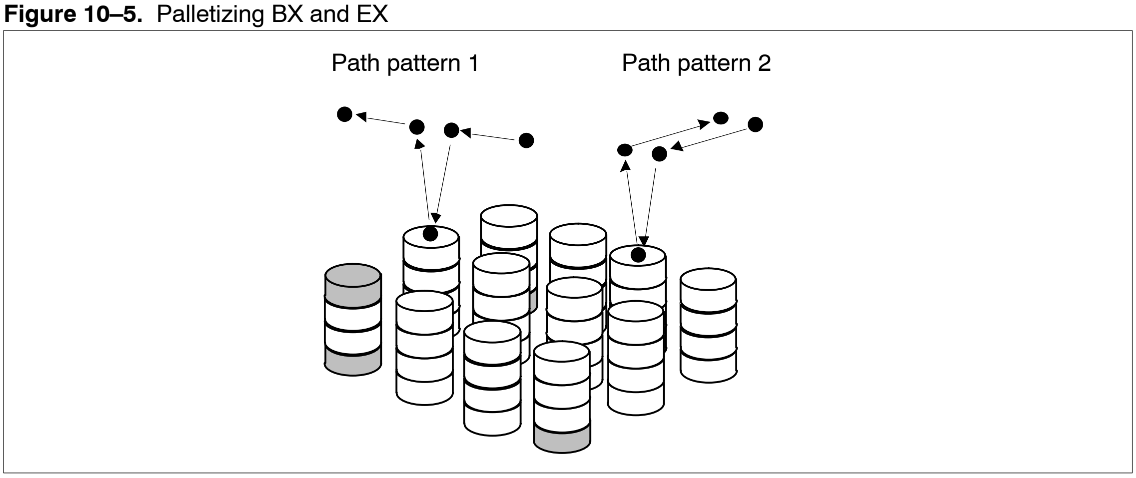
Figure 10–4. Palletizing E



Palletizing BX and EX

For palletizing BX and EX, multiple path patterns can be set. For palletizing B and E, only one path pattern can be set.

Figure 10–5. Palletizing BX and EX



10.2 Palletizing Instructions

The following palletizing instructions are available:

Table 10–1. Palletizing instructions

Instruction	Function
Palletizing instruction	Calculates the current path based on a stacking pattern, path pattern, and the value held in the palletizing register, and rewrites the position data of a palletizing motion instruction.
Palletizing motion instruction	A motion instruction dedicated to palletizing. It has position data of an approach point, stack point, or retraction point.
Palletizing end instruction	Increments (or decrements) the value of a palletizing register.

Palletizing instruction

Based on the value held in the palletizing register, the palletizing instruction calculates the position of the current stack point from a stack pattern, and also calculates the current path from a path pattern. It then rewrites the position data of a palletizing motion instruction.

Figure 10–6. Format of the Palletizing Instruction

PALLETIZING–[pattern]_i

B, BX, E, EX

Palletizing number (1 to 16)

Palletizing motion instruction

The palletizing motion instruction is a motion instruction that uses three path points — an approach point, stack point, and retraction point — as position data. This instruction is dedicated to palletizing. Each palletizing instruction rewrites such position data.

Figure 10–7. Format of the Palletizing Motion Instruction

J PAL_i [A_1] 100% FINE

Palletizing number (1 to 16)

Path point
A_n : Approach point, n = 1 to 8
BTM : Stack point
R_n : Retraction point, n = 1 to 8

Palletizing end instruction

The palletizing end instruction calculates the next stack point and increments (or decrements) the palletizing register value.

Figure 10–8. Format of the Palletizing End Instruction

PALLETIZING–END_i

Palletizing number (1 to 16)

Example

```
1:     PALLETIZING-B_3
2: J   PAL_3[ A_2 ] 50% CNT50
3: L   PAL_3[ A_1 ] 100mm/sec CNT10
4: L   PAL_3[ BTM ] 50mm/sec FINE
5:     hand1 open
6: L   PAL_3[ R_1 ] 100mm/sec CNT10
7: J   PAL_3[ R_2 ] 50% CNT50
8:     PALLETIZING-END_3
```

Palletizing number

Upon completion of the teaching of palletizing data, palletizing numbers are written automatically together with the instructions (palletizing instruction, palletizing motion instructions, and palletizing end instruction).

When a new palletizing operation is taught, a palletizing number is assigned automatically.

Palletizing register instruction

The palletizing register instruction is used to control palletizing. It performs stack point specification, comparison, and branch. (See Section 10.4.1)

Figure 10–9. Palletizing Register

PL [i] = (value)

Palletizing register number (1 to 32)

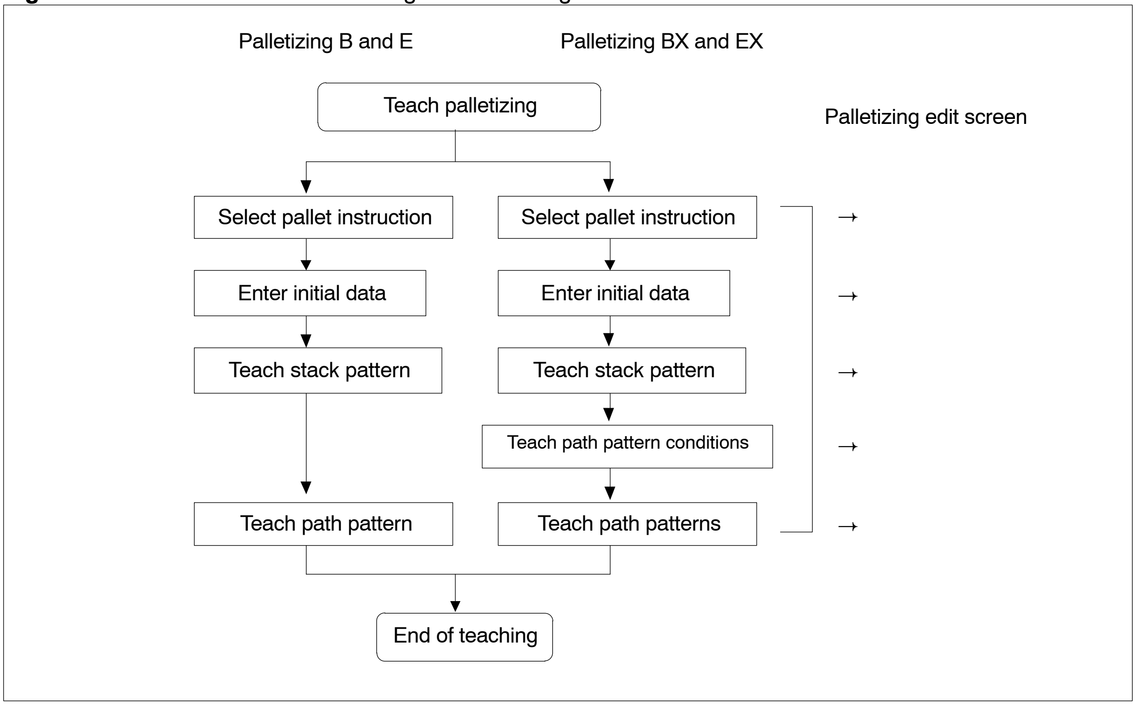
PL [i] : Pallet register [i]

[i, j, k] : Palletizing register element Row, column, layer (1 to 27)

10.3 Teaching the Palletizing Function

The palletizing function is taught using the following procedure:

Figure 10–10. Procedure for Teaching the Palletizing Function



The palletizing function is taught on the palletizing edit screens. One of the palletizing edit screens appears automatically when a palletizing instruction is selected.

When the palletizing function is taught, necessary palletizing instructions such as a palletizing instruction, palletizing motion instruction, and palletizing end instruction are inserted automatically.

The following sections explain the teaching of palletizing EX. For palletizing B, BX, or E, assume that some functions of palletizing EX are restricted.

NOTE To improve the motion accuracy of palletizing, TCP should be accurately set. (See Section 3.8.1,"Setting a tool coordinate system")

10.3.1 Selecting a palletizing instruction

To select a pallet instruction, select the type of palletizing to be taught (palletizing B, BX, E, or EX).

Procedure 10–1 Selecting a palletizing instruction

Condition

- Make sure that the teach pendant is enabled.
- Make sure that the palletizing instruction is selected on the program edit screen.

```
PROGRAM1                          JOINT 30%
                                       6/6
  5: J P[2] 300mm/sec CNT50
[End]

POINT                             TOUCHUP >
```

Step

1 Press NEXT ">" to display the next page. Press F1 "[INST]" to display a submenu.

```
[INST]
```

F1

```
Instruction                    JOINT  30 %
 1 Registers           5 JMP/LBL
 2 I/O                 6 CALL
 3 IF/SELECT           7 Palletizing
 4 WAIT                8 ---next page---
 PROGRAM1
                                       6/6
   5: J P[2] 300mm/sec CNT50
```

2 Select "Palletizing."

```
           JOINT  30%
       5 JMP/LBL
       6 CALL
       7 Palletizing
       8 --next page--
```

ENTER

```
 PALLETIZING system
 1 PALLETIZING-B   5
 2 PALLETIZING-END 6
 3                 7
 4                 8

 PROGRAM1
                                       6/6
   5: J P[2] 300mm/sec CNT50
```

3 Select "PALLETIZING–EX." The screen then changes to the initial data input screen automatically, one of the palletizing edit screen.

```
PALLETIZING system
 1 PALLETIZING-B   5
 2 PALLETIZING-END 6
 3                 7
 4                 8
 PROGRAM1
```

ENTER

```
 PROGRAM1                      JOINT  30 %
 PALLETIZING Configuration
   PALETIZING_1  [                ]
   TYPE = [PALLET  ]   INCR = [   1 ]
     PAL REG   = [  1]  ORDER = [RCL  ]
     ROWS     = [  1   FIX  ]
     COLUMNS  = [  1   FIX  ]
     LAYERS   = [  1   FIX  ]
     AUXILIARY POS = [ NO   ]
    APPR =[ 1] RTRT =[ 1]
 Press ENTER

  PROG                            DONE
```

10.3.2 Inputting initial data

On the initial data input screen, specify how palletizing is performed. The data set on the initial data input screen is used for subsequent teach screens. The initial data screen contains the following items:

For palletizing B

```
PRG1                           JOINT  30%
PALLETIZING Configuration

  PALLETIZING_1   [                     ]
  TYPE = [PALLET  ]       INCR = [ 1 ]
  PAL REG    = [ 1 ]     ORDER = [RCL   ]
    ROWS     = [ 5  ]
    COLUMNS  = [ 4  ]
    LAYERS   = [ 3  ]
    AUXILIARY POS = [ NO  ]
  APPR = [ 2 ] RTRT = [ 2 ]
Press ENTER
PROG                             DONE
```

For palletizing BX

```
PRG1                           JOINT  30%
PALLETIZING Configuration

  PALLETIZING_1   [                     ]
  TYPE = [PALLET  ]       INCR = [ 1 ]
  PAL REG    = [ 1 ]     ORDER = [RCL   ]
    ROWS     = [ 5  ]
    COLUMNS  = [ 4  ]
    LAYERS   = [ 3  ]
    AUXILIARY POS = [ NO  ]
  APPR = [ 2 ] RTRT = [ 2 ] PATTERN = [ 2 ]
Press ENTER
PROG                             DONE
```

For palletizing E

```
PRG1                           JOINT  30%
PALLETIZING Configuration

  PALLETIZING_1   [                     ]
  TYPE = [PALLET  ]       INCR = [ 1 ]
  PAL REG    = [ 1 ]     ORDER = [RCL   ]
    ROWS     = [ 5  2 LINE  FIX  ]
    COLUMNS  = [ 4  2 LINE  FIX  ]
    LAYERS   = [ 3  2 LINE  FIX  1  ]
    AUXILIARY POS = [ NO  ]
  APPR = [ 2 ] RTRT = [ 2 ]
Press ENTER
PROG                               DONE
```

For palletizing EX

```
PRG1                           JOINT  30%
PALLETIZING Configuration

  PALLETIZING_1   [                     ]
  TYPE = [PALLET  ]       INCR = [ 1 ]
  PAL REG    = [ 1 ]     ORDER = [RCL   ]
    ROWS     = [ 5  2 LINE  FIX  ]
    COLUMNS  = [ 4  2 LINE  FIX  ]
    LAYERS   = [ 3  2 LINE  FIX  1  ]
    AUXILIARY POS = [ NO  ]
  APPR = [ 2 ] RTRT = [ 2 ] PATTERN = [ 2 ]
Press ENTER
PROG                             DONE
```

Table 10–2. Types of Palletizing

	Arrangement mode	Layer pattern	Attitude control	Path pattern count
B	2–point teaching only	Not set	Always fixed	1
BX	2–point teaching only	Not set	Always fixed	1 to 16
E	2–point teaching, all–point teaching, or interval specification	Set	Fixed or split	1
EX	2–point teaching, all–point teaching, or interval specification	Set	Fixed or split	1 to 16

When a pallet instruction is selected, the initial data input screen corresponding to the selected type of palletizing appears. For palletizing EX, all palletizing functions can be specified. For palletizing B, BX, and E, restrictions are imposed on the specification of the functions.

This section explains how to enter initial data for palletizing EX. For palletizing B, BX, or E, assume that some functions of palletizing EX are restricted.

Table 10–3. Initial Palletizing Data

Palletizing number	A number is assigned automatically when a palletizing statement is taught. PALLETIZING_N: 1 to 16
Palletizing type	Specify whether the palletizing register is to be incremented or decremented by the palletizing end instruction. (See Section 10.4.1.) Select stacking (PALLET) or unstacking (DEPALLET).
Register increment	Specify the value by which the value held in the palletizing register is to be incremented or decremented by the palletizing end instruction. (See Section 10.4.1.)
Palletizing register	Specify the palletizing register to be used by the palletizing instruction and palletizing end instruction.
Order	Specify the stacking (unstacking) order of row, column, and layer. R: Row, C: Column, L: Layer
Numbers of rows, columns, and layers	Numbers of rows, columns, and layers for a stacking pattern. (See Subsection 10.3.3.) 1 to 127
Arrangement mode	How rows, columns, and layers are arranged for a stack pattern. The 2–point or all–point teaching, or interval specification can be specified (only for palletizing E or EX).
Attitude control	Control the attitude at rows,columns,and layers for a stacking pattern. Select E or EX.
Layer pattern count	How workpieces are stacked can be specified for each layer (only for palletizing E or EX). 1 to 16
Number of approach points	Number of approach points in a path pattern. (See Subsection 10.3.5.) 0 to 7
Number of retraction points	Number of retraction points in a path pattern. (See Subsection 10.3.5.) 0 to 7
Path pattern count	Number of path patterns (Section 10.3.4) (only for palletizing BX or EX). 1 to 16

Initial data related to the stacking method

In the palletizing function, the stack point is controlled using a palletizing register. (See Subsection 10.4.1, “Palletizing Register.”) How the palletizing register is controlled can be specified as initial data. According to this data, the way of stacking is determined.

Palletizing type Register increment

```
TYPE = [PALLET]  INCR = [1]
PAL REG =  [1]   ORDER = [RCL]
```

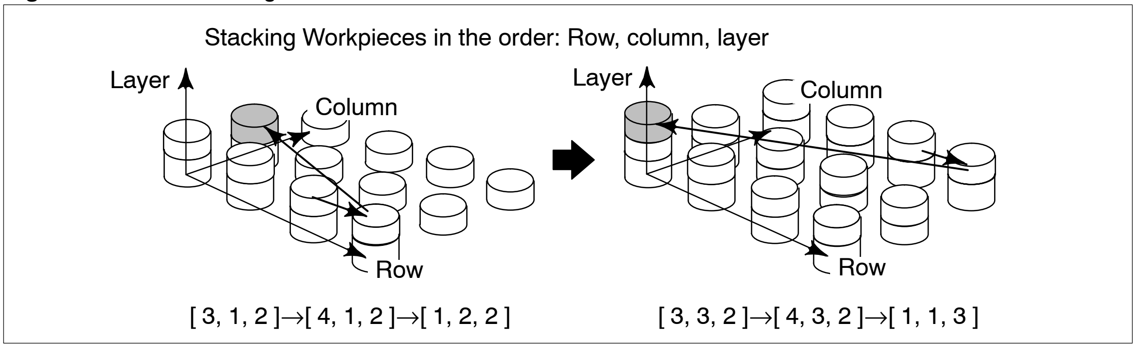
Palletizing register number Order

- For the pallet type (TYPE), specify either PALLET or DEPALLET (standard setting: PALLET). (See Subsection 10.4.1, “Palletizing Register.”)
- For the register increment (INCR), specify by which amount the stack (unstack) position advances or retracts. That is, specify a value by which the palletizing register is incremented or decremented by the palletizing end instruction. The standard setting is 1. (See Subsection 10.4.1, “Palletizing Register.”)
- As the palletizing register, specify the register number of a palletizing register used for stack control.

> **CAUTION**
> Make sure that the specified palletizing register number is not used by another palletizing function.

- For the order (ORDER), specify the stacking/unstacking order of row, column, and layer.

Figure 10–11. Palletizing Order



Initial data related to a stacking pattern

As the stacking pattern data, specify the numbers of rows, columns, and layers, attitude control type and also specify whether to provide an auxiliary position. (See Section 10.3.3, “Teaching a Stacking Pattern.”)

Numbers of rows, columns, and layers

Attitude control

```
Rows     = [4 FIX]
COLUMNS  = [3 FIX]
LAYERS   = [5 FIX]
AUXILIARYPOS = [NO FIX]
```

Auxiliary position

Initial data related to a path pattern

As the initial path pattern data, specify the number of approach points and the number of retraction points. (See Section 10.3.5, “Teaching a Path Pattern.”)

```
APPR = [2]   RTRT = [2]   PATTERN = [1]
```

Approach point count

Retraction point count

Path pattern count

Procedure 10–2 Inputting initial palletizing data

Step

1 As the palletizing instruction, select palletizing–EX. The initial data input screen then appears. (See Subsection 10.3.1, "Selecting a Palletizing Instruction.")

```
PALLETIZING system
 1 PALLETIZING-B    5
 2 PALLETIZING-END  6
 3                  7
 4
```

ENTER

```
PROGRAM1                           JOINT  30 %
PALLETIZING Configuration
  PALETIZING_1  [                      ]
  TYPE = [PALLET  ]   INCR = [   1 ]
    PAL REG    = [  1]  ORDER = [RCL  ]
    ROWS     = [  1   FIX  ]
    COLUMNS  = [  1   FIX  ]
    LAYERS   = [  1   FIX  ]
    AUXILIARY POS = [ NO  ]
   APPR =[ 1] RTRT =[ 1]

Press ENTER

 PROG                              DONE
```

NOTE PALLETIZING displayed on the initial data input screen indicates the fourth palletizing instruction in the program.

2 To enter a comment, follow the procedure below.

a Place the cursor on the comment line, then press the ENTER key. The character input submenu appears.

```
               JOINT 30%

 [                     ]
 ]   INCR=[ 1 ]
```

ENTER

```
                              JOINT 30%
1  Upper Case
2  Lower Case
3  Punctuation
4  Options                 ---Insert---
PROGRAM1
```

b Select the type of character input to be used — upper case, lower case, punctuation or options — with the ↑ and ↓ keys.

c Press an appropriate function key, then enter characters.

d After a comment has been completely entered, press the ENTER key.

```
PROGRAM1                          JOINT 30%
PALLETIZING Configuration

  PALLETIZING_1    [WORK4                ]


PROG                                  DONE
```

3 To select a palletizing type, move the cursor to the TYPE field, then select a function key.

```
PROG   PALLET   DEPALL
```

F2

```
PROGRAM1                          JOINT 30%
PALLETIZING Configuration

  TYPE = [PALLET]     INCR = [1]


PROG   PALLET   DEPALL                DONE
```

4 To enter a numeric value, press a numeric key, then press the ENTER key.

1 ENTER

```
PROGRAM1                             JOINT 30%
PALLETIZING Configuration

  TYPE = [PALLET]      INCR = [1 ]

PROG                                    DONE
```

5 Specify a palletizing order by selecting the function keys in the target order.

PROG R c

F2

```
PROGRAM1                             JOINT 30%
PALLETIZING Configuration

  PAL REG = [ 1 ]   ORDER = [R__]

PROG      R      C      L        DONE
```

After the second item has been selected, the third item is determined automatically.

PROG R c

F3

```
PROGRAM1                             JOINT 30%
PALLETIZING Configuration

  PAL REG = [ 1 ]   ORDER = [RCL]

PROG      R      C      L        DONE
```

6 To specify the type of attitude control, move the cursor to the setting field and select the function key.

PROG FIX INTER

F2

```
PROGRAM1                          JOINT  30 %
PALLETIZING Configuration
    LAYERS   = [  1   FIX  ]
    AUXILIARY POS = [ NO   ]

 PROG    FIX     INTER            DONE
```

7 To select whether there is an auxiliary position or not, move the cursor to the setting field and select function key menu.

PROG YES NO

F3

```
PROGRAM1                          JOINT  30 %
PALLETIZING Configuration
    LAYERS   = [  1   FIX  ]
    AUXILIARY POS = [ NO   ]

 PROG    YES     NO               DONE
```

8 To specify whether to set auxiliary points, position the cursor at the auxiliary point field and select the desired function key menu.

PROG YES NO

F3

```
PRG2                               JOINT  30%
PALLETIZING Configuration

    LAYERS   = [  1   200   FIX  1 ]
    AUXILIARY POS = [ NO  ]

PROG      YES     NO               DONE
```

NOTE When specifying the setting of auxiliary points, also select either of FIX/INTER.

9 Enter the approach point count and retraction point count.

10 To stop the initial data setting, press F1 “PROG.”

> **CAUTION**
> When the initial data setting is stopped before it is completed, the values set up to that time are invalidated.

```
PROGRAM1                            JOINT  30 %
PALLETIZING Configuration
  PALETIZING_1  [WORK PALLET      ]
  TYPE = [PALLET  ]   INCR = [    1 ]
    PAL REG    = [  1]  ORDER = [RCL  ]
    ROWS     = [  5    FIX  ]
    COLUMNS = [  4    FIX  ]
    LAYERS   = [  3    FIX  ]
    AUXILIARY POS = [ NO  ]
   APPR =[ 2] RTRT =[ 2]
Press ENTER

 PROG                               DONE
```

11 When all data items have been entered, press F5 “DONE.” The initial data input screen then disappears and the palletizing stacking pattern teach screen appears.

DONE

F5

```
PROGRAM1                               JOINT 30%
PALLETIZING Bottom Points

  1: *P [ 1, 1, 1 ]
  2: *P [ 5, 1, 1 ]
  3: *P [ 1, 4, 1 ]
  4: *P [ 1, 1, 3 ]

BACK                       RECORD     DONE  >
```

When the palletizing stacking pattern teach screen is displayed after the setting or changing of initial palletizing data is completed with F5 “DONE,” the palletizing register is initialized automatically. (See Section 10.4.1, “Palletizing Register.”)

10.3.3 Teaching a stacking pattern

On the palletizing stacking pattern teach screen, teach representative stack points of a stacking pattern. From these representative points, a target stack point is calculated automatically at the time of palletizing.

Stacking Pattern Teach Screen

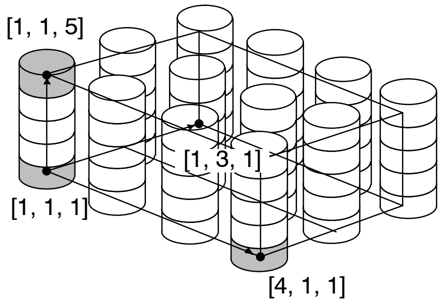
```
PROGRAM1                          JOINT 30%
PALLETIZING Bottom Points

   1: P [ 1, 1, 1 ]
   2: P [ 5, 1, 1 ]
   3: P [ 1, 4, 1 ]
   4: P [ 1, 1, 3 ]

BACK                   RECORD     DONE >
```

A list of the positions to be taught is displayed based on the initial palletizing data. Following this list, teach the positions of the representative stack points.

For the stacking pattern without an auxiliary position, individually teach four tops of the quadrangle of the stacking pattern.

Figure 10–12. Stacking Pattern with No Auxiliary Position

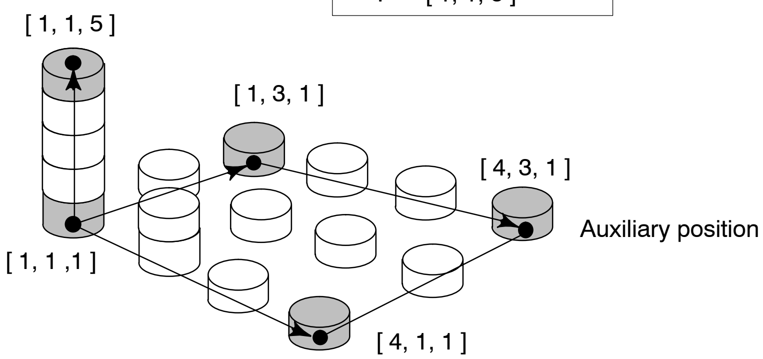
Initial data	Points to be taught
ROWS = [4 FIX] COLUMNS = [3 FIX] LAYERS = [5 FIX] AUXILIALY POS = [YES]	P [1, 1, 1] P [4, 1, 1] P [1, 3, 1] P [1, 1, 5]



For a stacking pattern with an auxiliary position, when the shape of the first layer is a trapezoid, also teach the fifth position using the function provided.

Figure 10–13. Stacking Pattern with an Auxiliary Position

Initial data	Points to be taught
ROWS = [4 FIX] COLUMNS = [3 FIX] LAYERS = [5 FIX] AUXILIALY POS = [YES]	P [1, 1, 1] P [4, 1, 1] P [1, 3, 1] P [4, 3, 1] P [1, 1, 5]



Types of arrangement modes/2–point teaching

When 2–point teaching is selected, teach the representative two points at both ends to set all the points in the row, column, and layer directions (standard).

NOTE The following explanation is not relevant to palletizing B and BX. See Section 10.3.4.

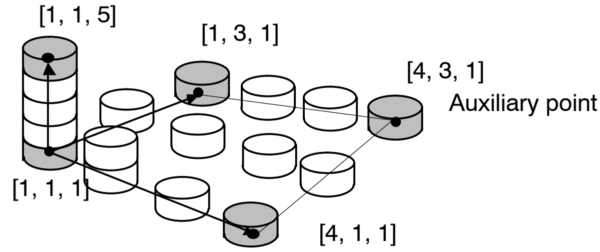
Figure 10–14. Teaching Method by 2–Point Teaching

Initial data

ROWS = [4]
COLUMNS= [3]
LAYERS = [5]
AUXILIARY POS = [YES]

Points to be taught

P [1, 1, 1]
P [4, 1, 1]
P [1, 3, 1]
P [4, 3, 1]
P [1, 1, 5]



All–point teaching

When selecting all–point teaching, directly teach all the points in the row, column, and layer directions.

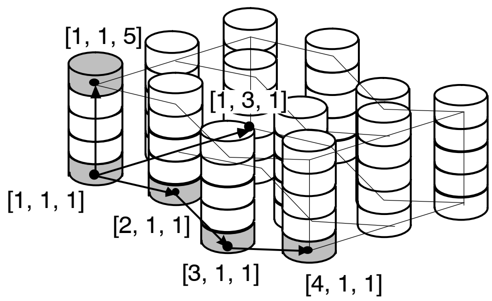
Figure 10–15. Teaching Method by All–Point Teaching

Initial data

ROWS = [4 FREE FIX]
COLUMNS= [3 2LINE FIX]
LAYERS = [5 2LINE FIX 1]
AUXILIARY POS = [NO]

Points to be taught

P [1, 1, 1]
P [2, 1, 1]
P [3, 1, 1]
P [4, 1, 1]
P [1, 3, 1]
P [1, 1, 5]



Interval specification

When selecting the interval specification, specify the two points at both ends in each of the row, column, and layer directions, as well as the distance between workpieces, to set all points.

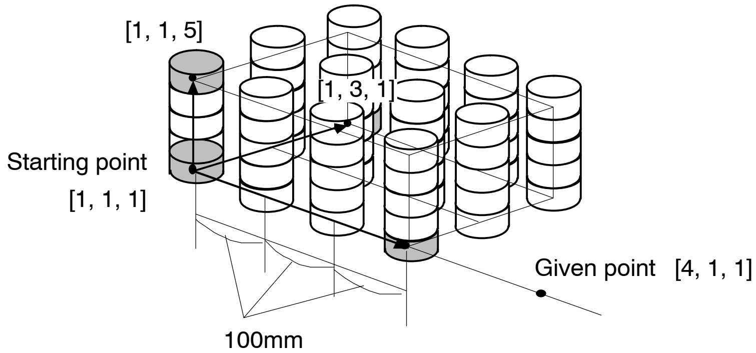
Figure 10–16. Teaching Method by Interval Specification

Initial data

```
ROWS       = [4  100      FIX  ]
COLUMNS= [3  2LINE  FIX  ]
LAYERS     = [5  2LINE  FIX  1]
AUXILIARY POS = [NO  ]
```

Points to be taught

```
P  [ 1, 1, 1 ]
P  [ 4, 1, 1 ]
P  [ 1, 3, 1 ]
P  [ 1, 1, 5 ]
```



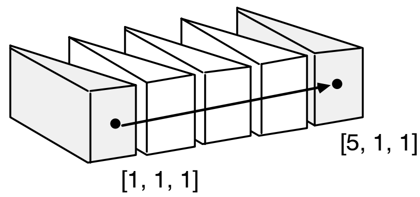
Types of attitude control

When the fixed attitude is specified, workpieces at all the stack points always take the attitude taught at point [1,1,1] (standard).

Figure 10–17. Attitudes of Workpieces at Stack Points when the Fixed Attitude Is Specified

Initial data

```
ROWS       = [5  2LINE  FIX  ]
COLUMNS= [1  2LINE  FIX  ]
LAYERS     = [1  2LINE  FIX  1]
AUXILIARY POS = [NO  ]
```

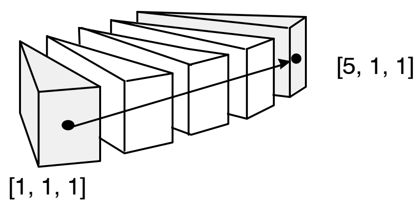


For the split attitude, when 2–point teaching is specified, workpieces take the attitudes obtained by splitting the attitudes taught at the two end points. When all–point teaching is specified, workpieces take the attitudes at the taught points.

Figure 10–18. Attitudes of Workpieces at Stack Points when the Split Attitude Is Specified

Initial data

```
ROWS       = [5  2LINE  INTER  ]
COLUMNS= [1  2LINE  FIX  ]
LAYERS     = [1  2LINE  FIX  1]
AUXILIARY POS = [NO  ]
```



Layer pattern count

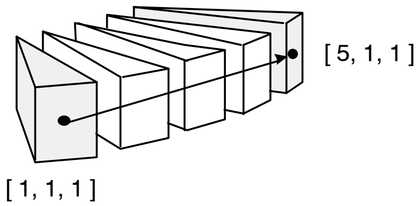
To change the stack pattern every few layers, enter the number of layer patterns.
The layer pattern count is valid only when 2–point teaching is specified for the layer arrangement (for other cases, the layer pattern count is always 1).

For the first layer, the workpieces are always stacked at the stack points in layer pattern 1.
When the layer pattern count is N, the numbers of layers and layer patterns are the same until layer N. For layer (N+1) and beyond, layer patterns starting from layer pattern 1 are repeated.
Specify the height of a layer only for layer pattern 1. For the height of each layer pattern, the deviation from the position in pattern 1 is corrected.

Figure 10–19. Attitude at the stacking point in division attitude

Initial data

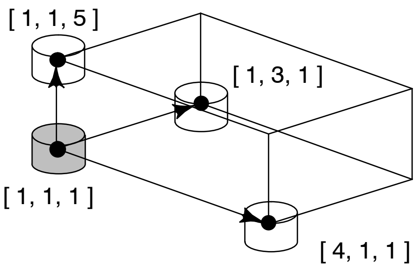
```
ROWS          = [5 INTER]
COLUMNS   = [1 FIX]
LAYERS       = [1 FIX]
AUXILIALY POS = [NO ]
```



When the total number of layers is less than 16, a number not greater than the total number can be set for the layer pattern count. If a number less than the layer pattern count is subsequently specified for the number of layers, the layer pattern count is automatically changed to the number of layers.

Procedure 10–3 Teaching a palletizing stacking pattern

Step 1 Based on the initial data setting, a list of stack points to be taught is displayed.



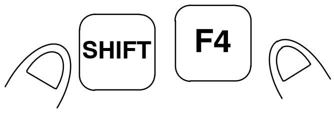

```
PROGRAM1                          JOINT 30%
PALLETIZING Bottom Points

   1: *P [ 1, 1, 1 ]
   2: *P [ 4, 1, 1 ]
   3: *P [ 1, 3, 1 ]
   4: *P [ 1, 1, 5 ]

BACK                     RECORD     DONE >
```

NOTE The number of representative stack points to be recorded depends on the numbers of rows, columns, and of layers set on the initial data input screen. In the above sample screen, four rows, three columns, and five layers are set. For each point, row, column, and then layer numbers are specified.

2 Move the robot by jog feed to a position which is to be taught as a representative stack point.

3 Place the cursor on the appropriate line, and press F4 “RECORD” while holding down the SHIFT key. The current robot position is then recorded.

```
PROGRAM1
PALLETIZING Bottom PointS

   1:*P[ 1, 1, 1 ]

            RECORD   DONE >
```




```
PROGRAM1                          JOINT 30%
PALLETIZING Bottom Points

   1: *P [ 1, 1, 1 ]
   2: *P [ 4, 1, 1 ]
   3: *P [ 1, 3, 1 ]
   4: *P [ 1, 1, 5 ]

BACK                     RECORD     DONE >
```

The positions not yet taught are marked with an asterisk. The positions already taught are marked with ”–”.

4 To display detailed position data for a taught representative stack point, move the cursor to the target stack point number, then press F5 “POSITION.” The detailed position data is then indicated.

```
PROGRAM1
PALLETIZING Bottom Points

   1:-P[ 1, 1, 1 ]

            POSITION>
```

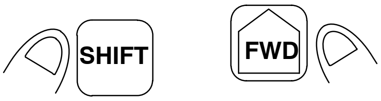
F5

```
Position Detail                  JOINT 30%
PAL_1[BTM]    UF:0   UT:1   CONF: NT 0
 X    516.129  mm    W  180.000  deg
 Y    111.347  mm    P    0.000  deg
 Z   1010.224  mm    R    0.000  deg
PROGRAM1
PALLETIZING Bottom Points

  1: *P [ 1, 1, 1 ]
  2: *P [ 4, 1, 1 ]
  3: *P [ 1, 3, 1 ]
  4: *P [ 1, 1, 5 ]

               CONFIG    DONE
```

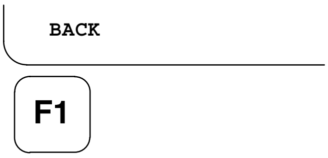
Numeric values can also be entered directly to specify position data. To return to the previous stacking pattern teach screen, press F4 “DONE.”

```
PROGRAM1                         JOINT 30%
PALLETIZING Bottom Points

  1: *P [ 1, 1, 1 ]
  2: *P [ 4, 1, 1 ]
  3: *P [ 1, 3, 1 ]
  4: *P [ 1, 1, 5 ]

BACK            RECORD    DONE          >
```

5 Pressing the FWD key while holding down the SHIFT key causes the robot to move to the representative stack point indicated by the cursor. This operation can be performed to confirm the taught point.

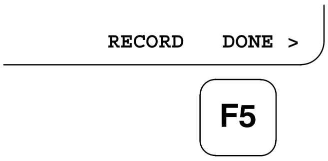


6 To return to the previous initial data teach screen, press F1 “BACK.”



7 Press F5 DONE to display the path pattern condition setting screen (BX or EX) or path pattern teaching screen (B or E) (Section 10.3.4 or 10.3.5).

NOTE When layer patterns are used (E or EX) and F5 DONE is pressed, the screen for specifying the stack pattern for the next layer appears.




```
PROGRAM1                          JOINT  30 %
PALLETIZING Route Points                1/3
      IF PL[ 1]=[*, 3-1, *]
   1:J P[A_2] 30% FINE
   2:* P[A_1] 30% FINE
   3:* P[BTM] 30% FINE
   4:* P[R_1] 30% FINE
   5:* P[R_2] 30% FINE
Teach Route Points
 BACK     POINT          RECORD   DONE >
```

10.3.4 Setting path pattern conditions

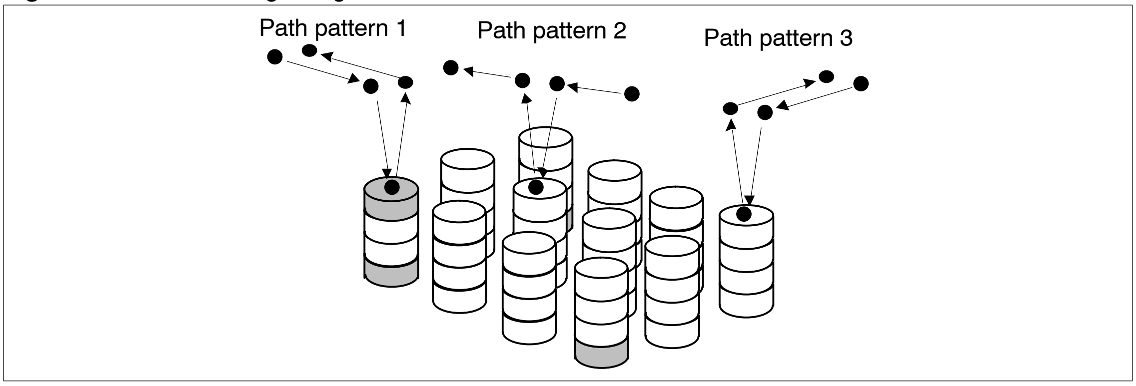
The screen explained in this section is not displayed for palletizing B or E. See Section 10.3.5.
The palletizing path pattern condition setting screen is used to set conditions indicating which path pattern is to be used for each stack point in advance when multiple path patterns are to be set on the path pattern teaching screen (See Section 10.3.5).
For palletizing BX and EX, multiple path patterns can be set independently for stack points. For palletizing B and E, this screen is not displayed because only one path pattern can be set.

Path pattern condition setting screen

```
PRG2                                    JOINT  30%
PALLETIZING Route Patterns
  PTN[1] = [  *   , 3-1,   * ]
  PTN[2] = [  *   , 3-2,   * ]
  PTN[3] = [  *   , 3-0,   * ]







BACK            DIRECT   MODULO   DONE  >
```

To specify a path for each stack point, as many path patterns as required must be specified when initial data is set. For each path pattern, set a path pattern condition.

Figure 10–20. Palletizing Using Three Path Patterns



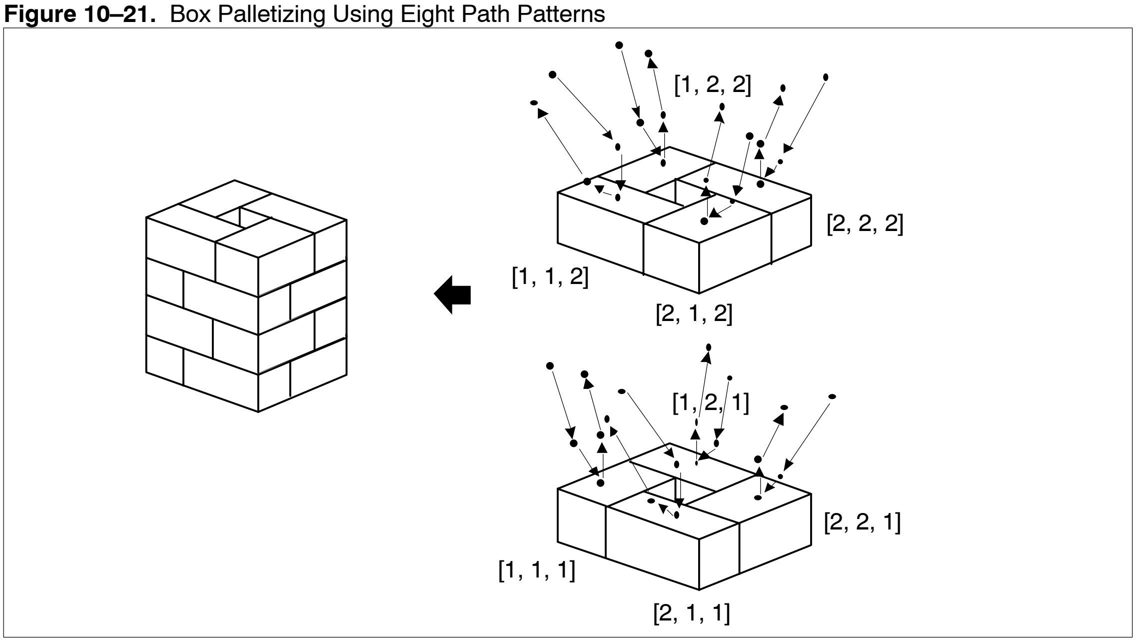
How to use path pattern conditions

- The execution of palletizing uses the path pattern with the condition number for which the row, column, and layer numbers at the stack point match the row, column, and layer (element) values of a path pattern condition.
- In direct specification mode, specify numbers between 1 to 127 for a stack point. An asterisk indicates an arbitrary stack point.
- In the remainder specification mode, specify path pattern condition element ”m–n” using a remainder system for a stack point.
  Layer element ”3–1” indicates a layer corresponding to a stack point value for which a remainder of 1 is obtained by dividing the value by 3.
- If the current stack point corresponds to no path pattern condition, an alarm occurs. If the current stack point corresponds to two or more path pattern conditions, a path pattern condition is used according to the following conditions:
  a. A path pattern condition specified in direct specification mode is used.
  b. When two or more path pattern conditions are specified in direct specification mode, a path pattern condition specified in the remainder specification mode is used. When two or more path pattern conditions are specified in remainder specification mode, a path pattern condition in which the greatest value is specified for m is used.
  c. When two or more path pattern conditions satisfy conditions a and b above, the path pattern condition having the smallest path pattern condition number is used.
  The following shows the priority among the sample path pattern conditions:

```
Example  PTN  [1] = [ * ,  1   ,  2   ]
         PTN  [2] = [ * ,  *   ,  2   ]
         PTN  [3] = [ * ,  3-2 ,  4–1 ]
         PTN  [4] = [ * ,  *   ,  4-1 ]
         PTN  [5] = [ * ,  *   ,  2-1 ]
         PTN  [6] = [ * ,  *   ,  *   ]
```

For the example shown on the previous page, pattern 1 is used for the stack points in column 1, pattern 2 is used for the stack points on column 2, and pattern 3 is used for the stack points in column 3.

Figure 10–21. Box Palletizing Using Eight Path Patterns




```
Example  PTN  [1] = [ 1 ,  1 ,  2-1 ]
         PTN  [2] = [ 2 ,  1 ,  2-1 ]
         PTN  [3] = [ 1 ,  2 ,  2-1 ]
         PTN  [4] = [ 2 ,  2 ,  2-1 ]
         PTN  [5] = [ 1 ,  1 ,  2-0 ]
         PTN  [6] = [ 2 ,  1 ,  2-0 ]
         PTN  [7] = [ 1 ,  2 ,  2-0 ]
         PTN  [8] = [ 2 ,  2 ,  2-0 ]
```

In the above example, eight path patterns are defined and repeated for every two layers because different paths must be set according to the box position.

Procedure 10–4 Setting palletizing path pattern conditions

Step

1 Condition items to be entered are displayed according to the value set for the pattern count as initial data.

```
PRG2                                JOINT   30%
PALLETIZING Route Pattern
  PTN[1] = [  *  ,  *  ,  *  ]
  PTN[2] = [  *  ,  *  ,  *  ]
  PTN[3] = [  *  ,  *  ,  *  ]



Enter value
BACK           DIRECT   MODULO   DONE   >
```

2 In direct specification mode, position the cursor to the point to be changed and enter a numeric value. To specify an asterisk (*), enter zero.

```
            JOINT   30%
Pattern
[  *  ,  *  ,  *  ]
[  *  ,  *        ]
[  *  ,  *        ]

                 1   INPUT
```

```
PRG2                                JOINT   30%
PALLETIZING Route Pattern
  PTN[1] = [  *  ,  1  ,  *  ]

BACK           DIRECT   MODULO   DONE   >
```

3 In remainder specification mode, press F4 MODULO. The target item is divided into two sub–items. Enter a value for each sub–item.

```
DIRECT   MODULO   DONE

            F4
```

```
PRG2                                JOINT   30%
PALLETIZING Route Pattern
  PTN[1] = [  *  ,  1 - 0,  *  ]

BACK           DIRECT   MODULO   DONE   >
```

4 To specify values in direct specification mode, press F3 DIRECT.

```
DIRECT   MODULO   DONE   >

  F3
```

5 Press F1 BACK to display the previous stack point teaching screen again.

```
BACK          DIRECT

  F1
```

6 Press F5 DONE to display the next path pattern teaching screen.

```
DIRECT   MODULO   DONE   >
```

F5

```
PRG2                                JOINT  30%
PALLETIZING Route Points
      IF PL[1]=[*, *, *]                   1/1
   1:Joint *P[A_2] 30% FINE
   2:Joint *P[A_1] 30% FINE
   3:Joint *P[BTM] 30% FINE
   4:Joint *P[R_1] 30% FINE
   5:Joint *P[R_2] 30% FINE
Teach Route Points
BACK   POINT             RECORD   DONE  >
```

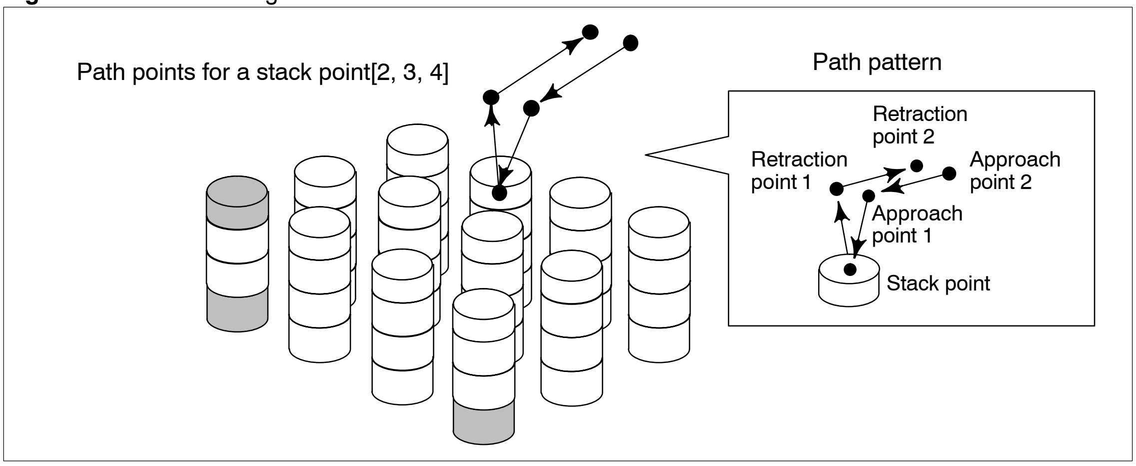
10.3.5 Teaching a path pattern

On the palletizing path pattern teach screen, set several path points which are passed before and after a workpiece is stacked (or unstacked) at a stack point. The path points change depending on the position of the stack point.

Path Pattern Teach Screen

```
PROGRAM1                            JOINT  30 %
PALLETIZING Route Points                  1/3
      IF PL[ 1]=[*, 3-1, *]
   1:J P[A_2] 30% FINE
   2:* P[A_1] 30% FINE
   3:* P[BTM] 30% FINE
   4:* P[R_1] 30% FINE
   5:* P[R_2] 30% FINE
Teach Route Points
 BACK     POINT          RECORD   DONE >
```

Figure 10–22. Palletizing Path



Procedure 10–5 Teaching a palletizing path pattern

Step 1 Based on the initial data setting, a list of the path points to be taught is displayed.

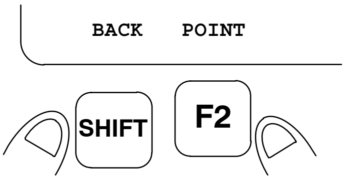
```
PROGRAM1                             JOINT  30 %
PALLETIZING Route Points                   1/3
      IF PL[ 1]=[*, *, *]
   1:* P[A_2] 30% FINE
   2:* P[A_1] 30% FINE
   3:* P[BTM] 30% FINE
   4:* P[R_1] 30% FINE
   5:* P[R_2] 30% FINE
Teach Route Points
 BACK     POINT          RECORD    DONE >
```

NOTE The number of path points to be recorded depends on the number of approach points set on the initial data input screen and the number of input points. In the above sample screen, the number of approach points is 2 and the number of retraction points is 2.

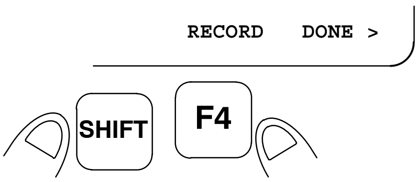
2 Move the robot by jog feed to a position which is to be taught as a path point.

3 Move the cursor to the setting field to be taught and teach the position using one of the following operation.

a Press F2 “POINT” while holding down the SHIFT key. When pressing F2 “POINT”, without pressing and holding the SHIFT key, default logical motion menu is displayed and then you can set the motion type or feedrate,etc.(This key is displayed only at teaching route pattern points.)



b Press and hold the SHIFT key and press F4 “RECORD”.




```
PROGRAM1                             JOINT  30 %
PALLETIZING Route Points                   1/3
      IF PL[ 1]=[*, 3-1, *]
   1:J P[A_2] 30% FINE
   2:* P[A_1] 30% FINE
   3:* P[BTM] 30% FINE
   4:* P[R_1] 30% FINE
   5:* P[R_2] 30% FINE

 BACK     POINT          RECORD    DONE >
```

The positions not yet taught are marked with an asterisk *.

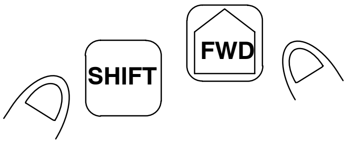
4 To display detailed position data for a taught path point, move the cursor to the target path point number, and press F5 “POSITION.” The detailed position data is then indicated.


```
Position Detail                      JOINT  30 %
P_1[A_2]  GP:1 UF:0 UT:1   CONF:NT00
 X    516.129   mm  W    180.000  deg
 Y    111.347   mm  P      0.000  deg
 Z   1010.224   mm  R      0.000  deg
PROGRAM2
PALLETIZING Route Points                   1/3
      IF PL[ 1]=[*, 3-1, *]
   1:J P[A_2] 30% FINE
   2:* P[A_1] 30% FINE
   3:* P[BTM] 30% FINE

                   CONFIG   DONE
```

Numeric values can also be entered directly to specify position data. To return to the previous path pattern teach screen, press F4 “DONE.”

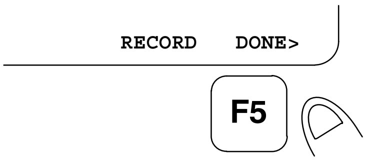
```
PROGRAM1                                 JOINT 30%
PALLETIZING Route Points                      1/3
     IF PL[1]=[*,*,*]
  1: Joint P[A_2] 100% FINE
  2: Joint P[A_1] 100% FINE
  3: Joint P[BTM] 100% FINE
  4: Joint P[R_1] 100% FINE
  5: Joint P[R_2] 100% FINE

BACK   POINT           RECORD     DONE >
```

5 Pressing the FWD key while holding down the SHIFT key causes the robot to move to the path point indicated by the cursor. This operation can be performed to confirm the taught point.



6 To return to the stacking pattern teach screen, press F1 “BACK.”

7 Press F5 “DONE” to terminate the palletizing edit screen and return to the program screen. The palletizing instructions are automatically written in the program.




```
PROGRAM1                                 JOINT 30%
                                             13/13
  5: L  P[2] 300mm/sec CNT50
  6:      PALLETIZING-B_1
  7: J  PAL_1[A_2] 100% FINE
  8: J  PAL_1[A_1] 100% FINE
  9: J  PAL_1[BTM] 100% FINE
 10: J  PAL_1[R_1] 100% FINE
 11: J  PAL_1[R_2] 100% FINE
 12:      PALLETIZING-END_1
[End]

POINT                              TOUCHUP >
```

8 Press F1 BACK to specify the previous path pattern. Press F5 DONE to specify the next path pattern.

9 After teaching of all the path patterns is complete, press F5 DONE to exit from the palletizing edit screen and display the program screen again. The palletizing instruction is automatically written in the program.

```
PRG1                                     JOINT  30%
                                              13/13
  5: L  P[2] 300mm/sec CNT50
  6:      PALLETIZING-EX_4
  7: L  PAL_4[A_2] 30% FINE
  8: L  PAL_4[A_1] 30% FINE
  9: L  PAL_4[BTM] 30% FINE
 10: L  PAL_4[R_1] 30% FINE
 11: L  PAL_4[R_2] 30% FINE
 12:      PALLETIZING-END_4
[END]

POINT                              TOUCHUP >
```

10 Editing, such as modifying a hand instruction executed at a stack position or the motion format at a path point can be performed on this program screen, in the same way as for normal programs.

```
PALLET1                         JOINT 30%
                                    14/14
  5: L   P[2] 300mm/sec CNT50
  6:       PALLETIZING-B_1
  7: L  PAL_1[A_2] 1000cm/min CNT30
  8: L  PAL_1[A_1] 300mm/sec CNT30
  9: L  PAL_1[BTM] 50mm/sec FINE
 10:      hand open
 11: L  PAL_1[R_1] 300mm/sec CNT30
 12: L  PAL_1[R_2] 1000cm/min CNT30
 13:      PALLETIZING-END_1
[End]

POINT                          TOUCHUP >
```

For details of palletizing programs, see Section 10.4, “Executing the Palletizing Function.”

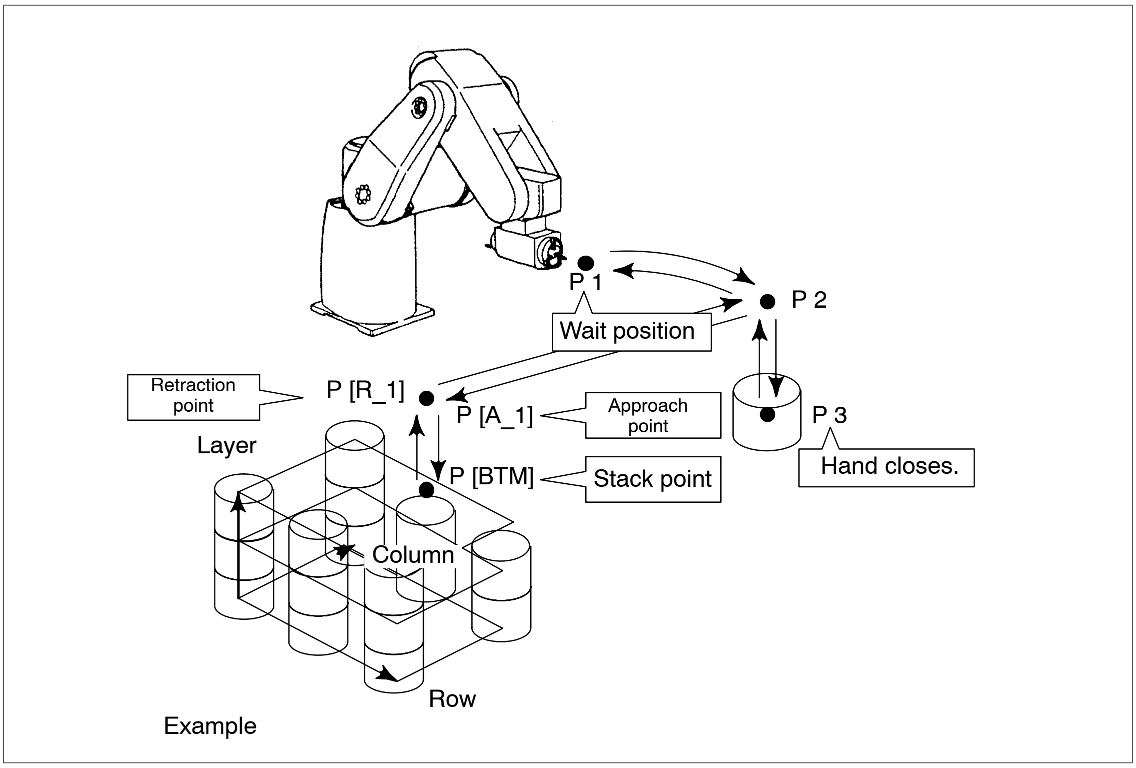
10.3.6 Notes on teaching the palletizing function

- The palletizing function is enabled only when a program contains these three instructions: A palletizing instruction, palletizing motion instruction, and palletizing end instruction. When just one of the three instructions is taught into a subprogram by another operation such as copying, normal operation cannot be performed.
- When all palletizing data has been taught, palletizing numbers are automatically written together with the instructions (a palletizing instruction, palletizing motion instruction, and palletizing end instruction). The user need not be concerned about the duplication of these numbers in other programs. (Each program has its own data for palletizing numbers.)
- In the palletizing motion instruction, C (circular motion) cannot be specified as the motion format.
- When palletizing, with a system with extended axes, there are some special conditions. For a system with extended axes, refer to Section 10.6 ”Palletizing Function with extended axes”

10.4 Executing the Palletizing Function

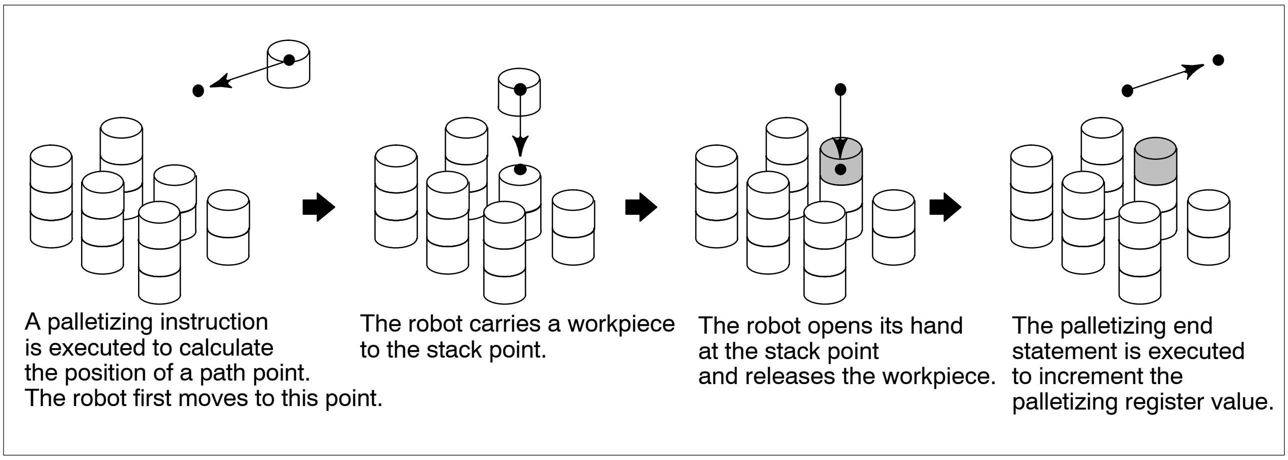
The execution of the palletizing function is shown below.

Figure 10–23. Example of Palletizing




```
 5: J  P[1] 100% FINE
 6: J  P[2] 70% CNT50
 7: L  P[3] 50mm/sec FINE
 8:    hand close
 9: L  P[2] 100mm/sec CNT50
10:    PALLETIZING-B_3
11: L  PAL_3[ A_1 ] 100mm/sec CNT10
12: L  PAL_3[ BTM ] 50mm/sec FINE
13:    hand open
14: L  PAL_3[ R_1 ] 100mm/sec CNT10
15:    PALLETIZING-END_3
16: J  P[2] 70% CNT50
17: J  P[1] 100% FINE
```

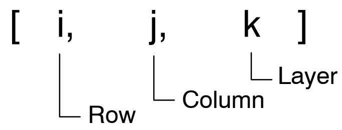
Figure 10–24. Workpiece Stacking Process



10.4.1 Palletizing register

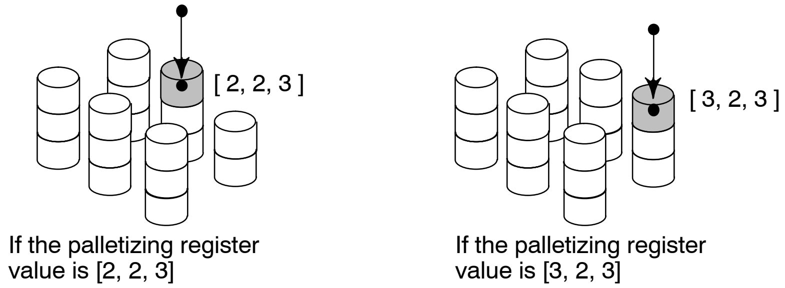
The palletizing register manages the position of the current stack point. When a palletizing instruction is executed, the value held in the palletizing register is referenced, and the actual stack point and path points are calculated. (See Section 7.5, “Palletizing Register.”)

Figure 10–25. Palletizing Register



The palletizing register indicates the row, column, and layer which are used for calculating the position of the stack point when the palletizing instruction is executed.

Figure 10–26. Relationship between the Palletizing Register and Stack Point



Updating the Palletizing Register

The palletizing register is incremented (or decremented) by executing the palletizing end instruction. The increment (decrement) method is determined depending on the initial data setting.

For 2–row, 2–column, and 2–layer palletizing with ORDER = [RCL] specified, executing the palletizing end instruction changes the palletizing register as follows:

Figure 10–27. Example of 2–Row, 2–Column, and 2–Layer Palletizing

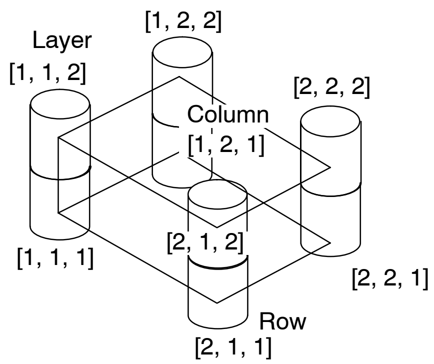


Table 10–4. Incrementing (Decrementing) Order of the Palletizing Register

	TYPE = [PALLET]		TYPE = [DEPALLET]	
	INCR = [1]	INCR = [–1]	INCR = [1]	INCR = [–1]
Initial value	[1, 1, 1]	[2, 2, 1]	[2, 2, 2]	[1, 1, 2]
↓	[2, 1, 1]	[1, 2, 1]	[1, 2, 2]	[2, 1, 2]
↓	[1, 2, 1]	[2, 1, 1]	[2, 1, 2]	[1, 2, 2]
↓	[2, 2, 1]	[1, 1, 1]	[1, 1, 2]	[2, 2, 2]
↓	[1, 1, 2]	[2, 2, 2]	[2, 2, 1]	[1, 1, 1]
↓	[2, 1, 2]	[1, 2, 2]	[1, 2, 1]	[2, 1, 1]
↓	[1, 2, 2]	[2, 1, 2]	[2, 1, 1]	[1, 2, 1]
↓	[2, 2, 2]	[1, 1, 2]	[1, 1, 1]	[2, 2, 1]
↓	[1, 1, 1]	[2, 2, 1]	[2, 2, 2]	[1, 1, 2]

Initializing the palletizing register

When F5 “DONE” is pressed upon completion of the setting or changing of initial palletizing data, the palletizing stacking pattern teach mode is set. At this time, the palletizing register is initialized automatically. (See Subsection 10.3.2, “Inputting Initial Data.”)

Table 10–5. Initial Value of the Palletizing Register

Initial data		initial value		
TYPE	**INCR**	**ROWS**	**COLUMNS**	**LAYERS**
PALLET	Positive value	1	1	1
	Negative value	Total number of rows	Total number of columns	1
DEPALLET	Positive value	Total number of rows	Total number of columns	Total number of layers
	Negative value	1	1	Total number of layers

10.4.2 Controlling the palletizing function by a palletizing register

In 5–row, 1–column, and 5–layer palletizing, suppress stack operation for the fifth workpiece in each even–numbered layer. (Stack five workpieces in odd–numbered layers, and stack four workpieces in even–numbered layers.)

```
PROGRAM2                           JOINT 30%
                                        1/17
   1:    PL[1] = [1,1,1]
   2:    LBL[1]
   3:    IF PL[1] = [2-0,*,5],JMP LBL[2]
   4: L  P[1] 100mm/sec FINE
   5:      hand close
   6:    PALLETIZING-B_1
   7: L  PAL_1[A_1] 100mm/sec CNT30
   8: L  PAL_1[BTM] 50mm/sec FINE
   9:      hand open
  10: L  PAL_1[R_1] 300mm/sec CNT30
  11:    LBL[2]
  12:    IF PL[1] = [5 1,5],JMP LBL[3]
  13:    PALLETIZING-END_1
  14:    JMP LBL[1]
  15:    LBL[3]
  16:    END
 [End]

 POINT                            TOUCHUP >
```

Palletizing register [1] is loaded with [1,1,1].

If the row is 5 and the layer is an even number, a jump to label [2] is performed.

If the row, column, and layer values are [5,1,5], a jump to label [3] is performed.

Set the values of the row, column and layer for the next stack operation in the palletizing register [1].

Procedure 10–6 Displaying the palletizing status

```
PROGRAM1                           JOINT 30%
                                        6/82
  5: L P[2]  300mm/sec CNT50
  6:   PALLETIZING-B_1
  7: L PAL_1 [ A_2 ] 1000cm/min CNT30


[MODIFY]                             LIST >
```

Step 1 To display the palletizing status, place the cursor on the palletizing instruction, then press F5 “[LIST].” The current stack point and the value of the palletizing register are displayed.

LIST>

F5

```
Pallet List                        JOINT 30%
PALLETIZING-B_1
 CURRENT BOTTOM POINT   [ 3, 1, 1 ]
 PALLET REGISTER [ 1 ]= [ 4, 1, 1 ]
 ROUTE PATTERN [ 1 ] = [ *, *, * ]
PROGRAM1
                                        6/82
```

10.5 Modifying the Palletizing Function

Modifying the palletizing function

The palletizing data and palletizing instructions which were taught can be modified later.

Procedure 10–7 Modifying palletizing data

```
PROGRAM1                                JOINT 30%
                                            6/82
  5: L  P[2] 300mm/sec CNT50
  6:      PALLETIZING-B_1
  7: L  PAL_1[A_2] 1000cm/min CNT30

Select item
[MODIFY]                                  LIST >
```

Step

1 Place the cursor on the palletizing instruction which is to be modified, then press F1 "[MODIFY]" to display the modification menu.

2 Select a target palletizing edit screen in the modification menu.

```
1 CONFIG
2 BOTTOM
3 ROUTE

MODIFY
```

F1 ENTER

```
PROGRAM1                                JOINT 30%
PALLETIZING Bottom Points

  1: --P [ 1, 1, 1 ] --
  2: --P [ 4, 1, 1 ] --
  3: --P [ 1, 3, 1 ] --
  4: --P [ 1, 1, 5 ] --
[End]
BACK                        RECORD      DONE >
```

To return to the previous palletizing edit screen, press F1 "BACK."

To proceed to the next palletizing edit screen, press F5 "DONE."

NOTE After palletizing data has been modified, the normal edit screen can be called from any palletizing screen. In this case, the new data after modification is kept valid.

3 When the modification is completed, press NEXT ">" to display the next page. Then press F1 "PROG."

```
PROG
```

F1

Procedure 10–8 Changing the palletizing number

```
PROGRAM1                                  JOINT 30%
                                              6/52
  6:       PALLETIZING-B_1
  7: L  PAL_1[A_1] 300mm/sec CNT30
  8: L  PAL_1[BTM] 50mm/sec FINE
  9:       hand open
 10: L  PAL_1[R_1] 300mm/sec CNT30
 11:       PALLETIZING-END_1
```

Step 1 Place the cursor on the palletizing instruction having the palletizing number which is to be changed, then enter a new number.

```
PROGRAM1

  6: PALLETIZING-B_1
```

2 ENTER

```
PROGRAM1                                  JOINT 30%
                                              6/52
  6:       PALLETIZING-B_2
  7: L  PAL_2[A_1] 300mm/sec CNT30
  8: L  PAL_2[BTM] 50mm/sec FINE
  9:       hand open
 10: L  PAL_2[R_1] 300mm/sec CNT30
 11:       PALLETIZING-END_2
```

At the same time the palletizing number of the palletizing instruction is changed, the palletizing numbers of the palletizing motion and palletizing end instructions are also changed.

CAUTION

When changing palletizing numbers, make sure that the new numbers are not used by other palletizing instructions.

10.6 Palletizing Function with Extended Axes

This section explains the special items when using palletizing on a system with extended axes.

When teaching the bottom point or route points for palletizing, which are different from the usual teaching of motion instruction, the position which is removed the position of extended axes is recorded.

Execution

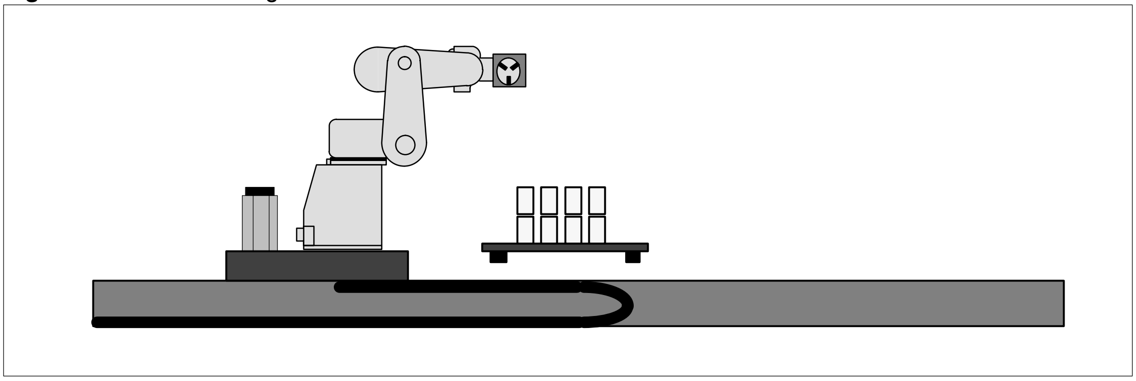
When palletizing is executed, palletizing will be done at the position of the extended axes at that time. (The robot does not automatically return to the position of the extended axes at teaching bottom points or route points.) For example, on a system with a linear extended axis, when the palletizing instruction is executed at a point 1000 mm away from the position at teaching palletizing, the robot will perform the same motion as taught at the position which is 1000 mm away from the teaching position.

Position teaching/modifying

When teaching the bottom/route points for palletizing on a system with extended axes or modifying the position, the following attention is necessary.

- When teaching bottom/route points for palletizing, it must be done at the fixed point.
  – When the position of extended axes shifts while teaching the bottom/route points for the same palletizing, the robot will follow a different path then the taught path when the program is executed.
- When modifying the position, move the robot to the position of the extended axes at teaching bottom/route points before modifying.

Figure 10–28. Palletizing function with extended axes



10.7 Palletizing All–Point Teaching

Palletizing all–point teaching allows the stacking (or unloading) of workpieces without changing the taught form by changing a system variable.

Operation

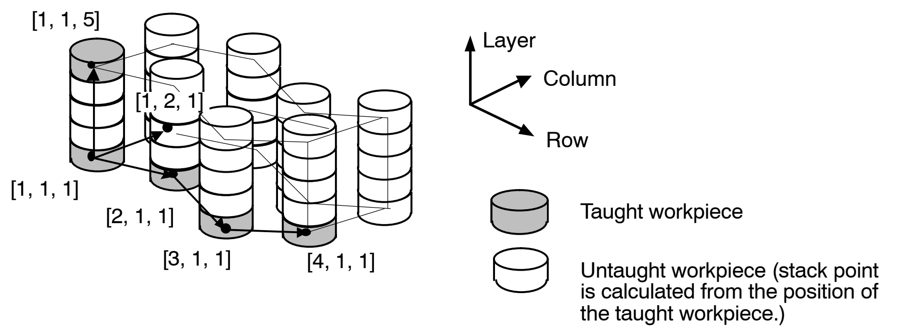
Set the following values to stack (or unload) workpieces without changing the taught attitude and form:

1. On the system variable screen, set system variable $PALCFG.$FREE_CFG_EN to TRUE (the initial value is TRUE).
2. On the palletizing initial data screen, set INTER for attitude control in the row, column, or layer direction for which FREE is specified as the arrangement mode.

   For each taught workpiece in the specified direction, all workpieces corresponding to the taught workpiece are stacked (or unloaded) with the same attitude and form as that for the taught workpiece.

   The following shows an example of palletizing for an irregular arrangement of four rows, two columns, and five layers.

Figure 10–29. Palletizing All–Point Teaching



The following palletizing initial data is set:

```
ROWS    = [4   FREE   INTER]
COLUMNS = [2   LINE   FIX]
LAYERS  = [5   LINE   FIX   1]
```

In this example, FREE and INTER are set in the row direction. When system variable $PALCFG.$FREE_CFG_EN is set to TRUE under this condition, the following forms are used:

- Form for stacking (unloading) workpieces on row 1: Form in P[1,1,1]
- Form for stacking (unloading) workpieces on row 2: Form in P[2,1,1]
- Form for stacking (unloading) workpieces on row 3: Form in P[3,1,1]
- Form for stacking (unloading) workpieces on row 4: Form in P[4,1,1]

Notes

Note the following points when using this function:

1. FREE and INTER can be set at the same time in only one of the row, column, and layer directions (when this function is not to be used, set system variable $PALCFG.$FREE_CFG_EN to FALSE).

   This is because if FREE and INTER are set at the same time in two or more of the row, column, and layer directions, two or more forms to be taken at the position of an untaught workpiece (workpiece for which the stack point is calculated from the position of a taught workpiece) are made.

   If such a setting is made in a program, the program causes the error indicated by PALT–024 Caluculation error occured and cannot be executed.
2. Carefully teach a program so that the program is not stopped due to a form mismatch alarm.

   If the form in the current position differs from the form data for the destination position, the robot cannot move in Linear operation mode (a form mismatch alarm occurs and execution of the program is stopped).

   The form at a stack point is used for the form at an approach or retraction point during palletizing. Therefore, if the pallet operation instruction to be executed first is in Linear operation mode, a form mismatch may occur depending on the form of the robot when an attempt is made to execute the line.

   To avoid such a problem, specify Joint for the operation mode of the first pallet operation instruction.

   For example, to avoid a form mismatch alarm, the following programming can be used for palletizing with three approach points and two retraction points.

```
:
10:PLLETIZING-EX_1
11:J  PAL_1[A_3]   100% FINE
12:L  PAL_1[A_2]   500mm/sec CNT50
13:L  PAL_1[A_1]   300mm/sec CNT10
14:L  PAL_1[BTM]   100mm/sec FINE
15:Open hand 1
16:L  PAL_1[R_1]   300mm/sec CNT10
17:L  PAL_1[R_2]   500mm/sec CNT50
18:PALLETIZING-END_1
:
```

APPENDIX

A. APPENDIX

This appendix summarizes the items necessary for using this model. It may also be used as an index.

☐ Contents of this appendix

A.1 List of Menus

Figure A–1. Function menu (Page 1)

Miscellaneous menu	Miscellaneous function
1 ABORT (ALL)	Program abort
2 Disable FWD/BWD	Disabling FWD/BWD from teach pendant
3 CHANGE GROUP	Change Motion Group*14
4 TOGGLE SUB GROUP	Sub group toggle*18
5 TOGGLE WRIST JOG	Wrist jog toggle
6	
7 RELEASE WAIT	Release Wait
8	
9	
0 — NEXT —	

Figure A–2. Function menu (Page 2)

Miscellaneous menu	Miscellaneous function
1 QUICK/FULL MENUS	Quick/full menu switch
2 SAVE	Save
3 PRINT SCREEN	Screen print
4 PRINT	Print
5	
6 UNSIM ALL I/O	Release the all of the simulated I/O
7	
8	
9	
0 — NEXT —	

Figure A–3. Screen Menu (Page 1)

Screen menu	Screen change menu	Screen
1 UTILITIES	Hints	Hint F5→ Help
	Prog Adjust*2	Program Adjunst Schedule List F2→ Program Adjust Schedule Detail
	Program shif*3	Program Shift
	Mirror Image*4	Mirror Image Shift
	Tool offset*5	Tool Offset
	Frame offset*6	UFrame Offset
	Angle entry*7	Angle Entry Shift
2 TEST CYCLE	Test Cycle	Test cycle
3 MANUAL FCNTS	Macros*1	Manual operation
4 ALARM	Alarm Log	Alarm history F5→ Alarm detail
5 I / O	Digital	Digital I/O F2→ Digital configuration F4→ Digital detail
	Group	Group I/O F2→ Group Configuration F4→ Group detail
	Analog*9	Analog I/O F2→ Analog configuration F4→ Analog detail
	Robot	Robot I/O F2→ Robot I/O detail
	UOP	Peripheral device I/O
	SOP	System Operator Panel I/O
	Inter Conect	DI–to–DO connection setting
	Link Device	I/O Link Device List F3→ I/O Unit Model B List F3→ I/O Points Setup
6 SETUP	General	General item setting
	Frames	Frame entry F2→ Frame detail
	Macro*8	Macro entry
	Ref Position*8	Reference position selection F3→ Reference position setting
	RSR/PNS	PNS setting*1
	Soft float*12	Soft float condition list F3→ Soft float condition detail
	Port Init	Port selection F3→ Port setting
	Ovrd Select	External override setting
	User Alarm	User alarm setting screen
	Torq Limit*	Torque Limit Setup
	Coord*	Coordinate Motion Setup
	Stroke limit*d	Stroke Limit Setup
	Space fnct.*11	Rectangular Space Check Setup
	Motion DO*14	Motion Group DO Setup
	Cont Turn*13	Continuous Turn Setup
7 FILE*15	File	File
	File Memory	File Memory
8		
9 USER		User
0 — NEXT —		

Figure A–4. Screen Menu (Page 2)

Screen menu	Screen change menu	Screen
1 SELECT		Program selection
2 EDIT		Program edit
3 DATA	Registers	Register
	Position Reg*16	Position register F4→ Position data information
	Pallet regis	Palletizing register
4 STATUS	Axis	Robot axis status
	Version ID	Software version
	Prg Timer*17	Program Timer Status List F2→ Program Timer Detail
	Sys Timer*17	System Timer Status
	Safety Signl	Safety Signal Status
	Order File*a	Order file
	Exec–hist	Execution history
	Memory	Memory status list F2→ Memory status detail
	Condition	Condition Monitor
5 POSITION		Current position
6 SYSTEM	Clock	Calendar
	Variables	System variable
	Servo Param*c	Servo parameter
	Master/Cal*b	Positioning
	OT Release	Over Travel Release
	Axis Limits	Joint operating area setting
	Config	System configuration
	Motion	Motion Performance
7		
8		
9		
0 — NEXT —		

Setting

The menu items indicated by * (alphabetical character) in Figure A–3 and Figure A–4 are displayed when the corresponding setting is made, as indicated below:

Table A–1. Option list

*	Setting
a	Can be displayed by setting $ODRDSP_ENB to 1.
b	Can be displayed by setting $MASTER_ENBL to 1.
c	Can be displayed by setting $SVPRM_ENB to 1.
d	Basic option only for Robot S–420 series

Options

The menu items indicated by * (numeral) in Figure A–1 to Figure A–4 are displayed when the corresponding option is added, as indicated below:

Table A–2. Option list

*	Option	Specification
1	Palletizing	A05B–2440–J500
2	Online position correction	A05B–2440–J517
3	Program shift	A05B–2440–H510
4	Mirror image	A05B–2440–H510
5	Tool offset	A05B–2440–H510
6	User coordinate system input	A05B–2440–H510
7	Angle input shift	A05B–2440–J614
8	Option instruction	A05B–2440–H510
9	Analog I/O	A05B–2440–H550
10	External program selection	A05B–2440–H510
11	Area check	A05B–2440–H510
12	Soft float function	A05B–2440–J612
13	Continuous rotation function	A05B–2440–J613
14	Multi–motion	A05B–2440–J601
15	Floppy disk drive connection	A05B–2440–H510
16	Position register	A05B–2440–H510
17	Hour meter	A05B–2440–H510
18	Extended axis control	A05B–2440–J518
19	Condition monitor function	A05B–2440–H518
20	Torqre limit function	A05B–2440–J611
21	Cooperative control	A05B–2440–J619

A.2 Types of Screens

Figure A–5. [1 UTILITIES]

Hint screen

```
UTILITIES Hints                   JOINT 10%

MENUS - Provides a list of all menus.
FCTN  - Provides helpful functions.
SELECT, EDIT, DATA, POSN, I/O, STAT-
    Displays menus with those names.
Function keys with [ ] show more choices.
F1 [TYPE] function key lists related
    screens within a menu.

  TYPE                             HELP
Hints
```

Program Adjust Schedule List Screen

```
UTILITIES Prog Adj    G1       JOINT  10%
    Program          Lines    Status  1/10
  1 TEST1            1-2      EDIT
  2 ********         0-0      ********
  3 ********         0-0      ********
  4 ********         0-0      ********
  5 ********         0-0      ********
  6 ********         0-0      ********
  7 ********         0-0      ********
  8 ********         0-0      ********
  9 ********         0-0      ********
 10 ********         0-0      ********

  TYPE  DETAIL                        >
Prog Adjust
 COPY   CLR_ADJ  CLR_ALL              >
```

Program Adjust Schedule Detail Screen

```
UTILITIES Prog Adj    G1       JOINT  10%
                                      1/11
Current schedule:1        Status: EDIT
 1  Program name:              TEST1
 2  Starting line number:      1
 3  Ending line number:        2
 4  X adjustment:              26.000mm
 5  Y adjustment:              0.000 mm
 6  Z adjustment:              0.000 mm
 7  W adjustment:              0.000 dg
 8  P adjustment:              0.000 dg
 9  R adjustment:              0.000 dg

[TYPE]  UNITS  SCHED   [CHOICE]        >
COPY  CLR_ADJ  CLR_ALL [CHOICE]        >
```

Program Shift (Program name) Screen

```
PROGRAM SHIFT          G2       JOINT  10%
Program                                1/6
 1 Original Program :          [TEST1     ]
 2   Range:                        WHOLE
 3   Start  line:(not used)        *****
 4   End    line:(not used)        *****
 5   New Program :             [TEST2     ]
 6   Insert line:                      0

Use shifted up,down arrows for next page

  TYPE                                 >
Program shif
 CLEAR                                 >
```

Program Shift (Position) Screen

```
PROGRAM SHIFT          G1       JOINT  10%
Shift amount/Teach                     1/3
Position data
X :********   Y :********   Z :********
1   Rotation:                       OFF

2   Source position        P1:

3   Destination position   Q1:

[TYPE]  EXECUTE            ON     OFF   >
CLEAR    DIRECT            ON     OFF   >
```

Figure A–6. [1 UTILITIES]

Program Shift (Direct entry) Screen

```
PROGRAM SHIFT          G1       JOINT  10%
Shift amount/Direct entry              1/3

  1 X (mm):                         100.00
  2 Y (mm):                           0.00
  3 Z (mm):                           0.00






[TYPE]  EXECUTE                          >

CLEAR    TEACH                           >
```

Mirror Image Shift (Program name) Screen

```
MIRROR IMAGE SHIFT    G1       JOINT  10%
Program                                1/6
 1 Original Program :         [TEST1     ]
 2   Range:                      WHOLE
 3   Start  line:(not used)      *****
 4   End    line:(not used)      *****
 5 New Program :              [TEST3     ]
 6   Insert line:(not used)      *****


Use shifted up,down arrows for next page


 TYPE                                    >
Mirror Image
CLEAR                                    >
```

Mirror Image Shift (Position) Screen

```
MIRROR IMAGE SHIFT    G1       JOINT  10%
Shift amount/Teach                     1/3
Position data
X :********   Y :********   Z :********

1   Rotation:                      OFF

2   Source position        P1:


3   Destination position   Q1:


[TYPE]  EXECUTE           ON      OFF    >
CLEAR                                    >
```

Tool Offset (Program name) Screen

```
TOOL OFFSET            G1       JOINT  10%
Program                                5/6
 1 Original Program :         [TEST1     ]
 2   Range:                      WHOLE
 3   Start  line:(not used)      *****
 4   End    line:(not used)      *****
 5 New Program :              [TEST4     ]
 6   Insert line:(not used)      *****


Use shifted up,down arrows for next page


 TYPE                                    >
Tool offset
CLEAR                                    >
```

Tool Offset (Utool number) Screen

```
TOOL OFFSET            G1       JOINT  10%
UTOOL number                           1/3

  1   Old UTOOL number:                0
  2   New UTOOL number:                0
  3   Convert type:          TCP fixed







[TYPE]  EXECUTE                          >
CLEAR                     [CHOICE]       >
```

Figure A–7. 2 TEST CYCLE]

Uframe Offset (Program name) Screen

```
UFRAME OFFSET          G1        JOINT  10%
Program                                 5/6
 1 Original Program :          [TEST1    ]
 2   Range:                       WHOLE
 3   Start  line:(not used)       *****
 4   End    line:(not used)       *****
 5 New Program :               [TEST4    ]
 6   Insert line:(not used)       *****


Use shifted up,down arrows for next page


 TYPE                                     >
Frame offset

CLEAR                                     >
```

Uframe Offset (Uframe number) Screen

```
UFRAME OFFSET          G1        JOINT  10%
UFRAME number                           1/3

  1   Old UFRAME number:                  0
  2   New UFRAME number:                  0
  3   Convert Position data (Y/N):       NO


[TYPE]  EXECUTE                           >

CLEAR                      [CHOICE]       >
```

Angle Entry Shift (Program name) Screen

```
ANGLE ENTRY SHIFT      G1        JOINT  10%
Program                                 5/6
 1 Original Program :          [TEST1    ]
 2   Range:                       WHOLE
 3   Start  line:(not used)       *****
 4   End    line:(not used)       *****
 5 New Program :               [TEST6    ]
 6   Insert line:(not used)       *****


Use shifted up,down arrows for next page


 TYPE                                     >
Angle entry

CLEAR                                     >
```

Angle Entry Shift (Shift amount) Screen

```
ANGLE ENTRY SHIFT      G1        JOINT  10%
Shift amount                            1/7
Position data of P1
X :********   Y :********   Z :********

1   Rotation plane           P1:
2                            P2:
3                            P3:
4   Rotation axis enable:          FALSE
5   Rotation axis            P0: Not used
6   Angle(deg):                     0.00
7   Repeating times:                   1

[TYPE]  EXECUTE             REFER  RECORD >

CLEAR                                     >
```

Angle Entry Shift (Rotation axis direct entry) Screen

```
ANGLE ENTRY SHIFT      G1        JOINT  10%
Shift amount                            1/4
Rotation center axis direct entry
  1 Frame:                   WORLD\FRAME\
  2 X (mm):                       100.00
  3 Y (mm):                         0.00
  4 Z (mm):                         0.00


[TYPE]  EXECUTE         [CHOICE]          >

CLEAR                   [CHOICE]          >
```

Figure A–8. [2 TEST CYCLE]

Test cycle screen

```
TEST CYCLE                          JOINT 30%
                                         2/7
GROUP:1
  1 Robot lock:                  OFF
  2 Dry run:                     OFF
  3 Cart. dry run speed:         300.000mm/s
  4 Joint dry run speed:         25.000%
  5 Digital/Analog I/O:          ENABLE
  6 Step statement type:         STATEMENT
  7 Step path node:              OFF


 TYPE                            ON   OFF

Test cycle
```

Figure A–9. [3 MANUAL FCTNS]

Manual operation screen

```
MANUAL Macros                       JOINT 10%
                                         3/6
     Instruction
  1 Open hand1                      NOT ASSIGN
  2 Close hand1                     NOT ASSIGN
  3 Relax hand1                     NOT ASSIGN
  4 Open hand2                      NOT ASSIGN
  5 Close hand2                     NOT ASSIGN
  6 Relax hand2                     NOT ASSIGN


 TYPE            EXEC

  Macros
```

Figure A–10. [4 ALARM]

Alarm Screen

```
INTP-204 (TEST2, 1) Invalid value for in

Alarm : Active          G1       JOINT  10%
                                        1/1
   1 INTP-204 (TEST2, 1) Invalid value fo





   TYPE            HIST
 Alarm Log
```

Alarm history screen

```
INTP-224 (SAMPLE1, 7) Jump label is fail
MEMO-027 Spedified line does not exist
Alarm                          JOINT  30 %
                                       1/7
  1 INTP-224 (SAMPLE1, 7) Jump label is
  2 SRVO-002 Teach pendant E-stop
  3 R E S E T
  4 SRVO-027 Robot not mastered(Group:1)
  5 SYST-026 System normal power up




[TYPE]                       CLEAR   HELP
```

Alarm detail screen

```
INTP-224 (SAMPLE1, 7) Jump label is fail

INTP-224 (SAMPLE1, 7) Jump label is fail
MEMO-027 Specified line does not exist
30-MAY-44 07:15
STOP.L                            00000110
Alarm
                                      1/7
  1 INTP-224 (SAMPLE1, 7) Jump label is
  2 SRVO-002 Teach pendant E-stop
  3 R E S E T
  4 SRVO-027 Robot not mastered(Group:1)
  5 SYST-026 System normal power up
[TYPE]                       CLEAR   HELP
```

Figure A–11. [5 I / O]

Digital input screen

```
 I/O Digital In                       JOINT 30%

        #  SIM STATUS
  SDI[ 1]  U  ON  [                       ]
  SDI[ 2]  U  OFF [                       ]
  SDI[ 3]  U  OFF [                       ]
  SDI[ 4]  U  ON  [                       ]
  SDI[ 5]  U  ON  [                       ]
  SDI[ 6]  U  OFF [                       ]
  SDI[ 7]  U  OFF [                       ]
  SDI[ 8]  U  ON  [                       ]
  SDI[ 9]  U  ON  [                       ]

  TYPE     CONFIG    IN/OUT    ON     OFF
 Digital
```

Digital output configuration screen

```
 I/O Digital Out                       JOINT  30%
   #      RANGE      RACK   SLOT   START PT
   1   SDO[  1-  8]    0      1       21
   2   SDO[  9- 16]    0      1       29
   3   SDO[ 17- 24]    0      1       37
   4   SDO[ 25- 32]    0      2        1
   5   SDO[ 33- 40]    0      2        9
   6   SDO[ 41- 48]    0      2       17
   7   SDO[ 49- 56]    0      2       25
   8   SDO[ 57- 64]    0      2       33
   9   SDO[ 65- 72]    *      *        *

 [TYPE]  MONITOR  IN/OUT  DETAIL   HELP >
```

Group output screen

```
 I/O Group Out                        JOINT  30%
       #   SIM VALUE
   GO[   1]  S      1  [              ]
   GO[   2]  U     10  [              ]
   GO[   3]  U     23  [              ]
   GO[   4]  *      0  [              ]
   GO[   5]  *      0  [              ]
   GO[   6]  *      0  [              ]
   GO[   7]  *      0  [              ]
   GO[   8]  *      0  [              ]
   GO[   9]  *      0  [              ]

   TYPE    CONFIG  IN/OUT SIMULATE UNSIM
 Group
```

Group output configuration screen

```
 I/O Group Out                          JOINT  30%
  GO #   RACK   SLOT   START PT   NUM PTS
   1       0      2       17         4
   2       0      2       21         4
   3       0      2       25         8
   4       *      *        *         *
   5       *      *        *         *
   6       *      *        *         *
   7       *      *        *         *
   8       *      *        *         *
   9       *      *        *         *

 [TYPE]  MONITOR  IN/OUT  DETAIL   HELP >
```

Analog input screen

```
 I/O Analog In                        JOINT  30%
       #   SIM VALUE                   1/25
   AI[   1]  S     85  [              ]
   AI[   2]  U      0  [              ]
   AI[   3]  *      0  [              ]
   AI[   4]  *      0  [              ]
   AI[   5]  *      0  [              ]
   AI[   6]  *      0  [              ]
   AI[   7]  *      0  [              ]
   AI[   8]  *      0  [              ]
   AI[   9]  *      0  [              ]
   AI[  10]  *      0  [              ]
  TYPE     CONFIG  IN/OUT SIMULATE UNSIM
 Analog
```

Analog input configuration screen

```
 I/O Analog In                          JOINT  30%
                                          1/25
  AI #   RACK   SLOT   CHANNEL
   1       0      1       1
   2       0      1       2
   3       0      0       0
   4       0      0       0
   5       0      0       0
   6       0      0       0
   7       0      0       0
   8       0      0       0
   9       0      0       0

 [TYPE]  MONITOR  IN/OUT  DETAIL   HELP >
```

Figure A–12. [5 I / O]

Robot data output screen

```
I/O Robot Out                        JOINT 30%

        #       STATUS
  RDO[ 1]        OFF  [                       ]
  RDO[ 2]        OFF  [                       ]
  RDO[ 3]        OFF  [                       ]
  RDO[ 4]        ON   [                       ]
  RDO[ 5]        ON   [                       ]
  RDO[ 6]        OFF  [                       ]
  RDO[ 7]        *    [                       ]
  RDO[ 8]        *    [                       ]
  RDO[ 9]        *    [                       ]

  TYPE   DETAIL  IN/OUT       ON    OFF

 Robot
```

Robot output detail screen

```
I/O Robot Out                        JOINT  30%
       Robot Output Detail
  1  Comment:  [  1] [                  ]
  2  Comment:  [  2] [                  ]
  3  Comment:  [  3] [                  ]
  4  Comment:  [  4] [                  ]
  5  Comment:  [  5] [                  ]
  6  Comment:  [  6] [                  ]
  7  Comment:  [  7] [                  ]
  8  Comment:  [  8] [                  ]
  9  Comment:  [  9] [                  ]

[TYPE]  MONITOR  IN/OUT
```

Peripheral device data input screen

```
I/O UOP In                          JOINT  30%
      #   STATUS
  UI[   1]    ON      [        *Hold]
  UI[   2]   OFF      [  Falt reset]
  UI[   3]   OFF      [        Start]
  UI[   4]    ON      [       Enable]
  UI[   5]    ON      [         PNS1]
  UI[   6]   OFF      [         PNS2]
  UI[   7]   OFF      [         PNS3]
  UI[   8]    ON      [         PNS4]
  UI[   9]     *      [             ]
  UI[  10]     *      [             ]

  TYPE   CONFIG  IN/OUT    ON     OFF

 UOP
```

System Operation Panel Output Screen

```
I/O SOP Out             G1      JOINT  10%
      #    STATUS                       1/15
  SO[   0]   OFF      [Remote LED   ]
  SO[   1]   OFF      [Cycle start  ]
  SO[   2]   OFF      [Hold         ]
  SO[   3]   OFF      [Fault LED    ]
  SO[   4]   OFF      [Batt alarm   ]
  SO[   5]   OFF      [User LED#1   ]
  SO[   6]   OFF      [User LED#2   ]
  SO[   7]    ON      [TP enabled   ]
  SO[   8]   OFF      [             ]
  SO[   9]   OFF      [             ]

  TYPE           IN/OUT     ON     OFF

 SOP
```

DI–to–DO connection setting screen (RDI–>SDO)

```
I/O INTER CONNECT                    JOINT 30%

NO.  Enb/Disabl        INPUT        OUTPUT
 1   DISABLE         RI[ 1]   ->  SDO[ 0]
 2   DISABLE         RI[ 2]   ->  SDO[ 0]
 3   DISABLE         RI[ 3]   ->  SDO[ 0]
 4   DISABLE         RI[ 4]   ->  SDO[ 0]
 5   DISABLE         RI[ 5]   ->  SDO[ 0]
 6   DISABLE         RI[ 6]   ->  SDO[ 0]
 7   DISABLE         RI[ 7]   ->  SDO[ 0]
 8   DISABLE         RI[ 8]   ->  SDO[ 0]
 9   DISABLE         RI[ 9]   ->  SDO[ 0]

  TYPE        IN/OUT    ENABLE   DISABLE

Inter Conect
```

DI–to–DO connection setting screen (SDI–>RDO)

```
I/O INTER CONNECT                    JOINT 30%

NO.  Enb/Disabl        INPUT        OUTPUT
 1   DISABLE         SDI[ 1]  ->  RDO[ 0]
 2   DISABLE         SDI[ 2]  ->  RDO[ 0]
 3   DISABLE         SDI[ 3]  ->  RDO[ 0]
 4   DISABLE         SDI[ 4]  ->  RDO[ 0]
 5   DISABLE         SDI[ 5]  ->  RDO[ 0]
 6   DISABLE         SDI[ 6]  ->  RDO[ 0]
 7   DISABLE  1 RDI->SDO  7]  ->  RDO[ 0]
 8   DISABLE  2 SDI->RDO  8]  ->  RDO[ 0]
 9   DISABLE  3 SDI->SDO  9]  ->  RDO[ 0]

  TYPE        IN/OUT    ENABLE   DISABLE

Inter Conect
```

Figure A–13. [5 I / O]

DI–to–DO connection setting screen (DI–>DO)

```
INTER CONNECT                    JOINT  10%
                                       1/24
 No. Enb/Disabl    INPUT        OUTPUT
  1  DISABLE      DI [   0] ->  DO [   1]
  2  DISABLE      DI [   0] ->  DO [   0]
  3  DISABLE      DI [   0] ->  DO [   0]
  4  DISABLE      DI [   0] ->  DO [   0]
  5  DISABLE      DI [   0] ->  DO [   0]
  6  DISABLE      DI [   0] ->  DO [   0]
  7  DISABLE      DI [   0] ->  DO [   0]
  8  DISABLE      DI [   0] ->  DO [   0]
  9  DISABLE      DI [   0] ->  DO [   0]

  TYPE          [SELECT]  ENABLE  DISABLE
Inter Conect
```

I/O Link Device List Screen

```
I/O Link Device       G1        JOINT  10%
                                       1/4
  Device Name      Comment       RackSlot
  1 PrcI/O CA [                 ] 0    1
  2 Model A   [                 ] 1    0
  3 Model B   [                 ] 2    0
  4 Model C   [                 ] 3    0

  TYPE               DETAIL          CLR_ASG
Link Device
```

I/O Link Device (Model B) List Screen

```
I/O Link Device       G1        JOINT  10%
   Model B                            1/30
   Slot Base     Exp.       Comment
  1 *******    *******   [            ]
  2 *******    *******   [            ]
  3 *******    *******   [            ]
  4 *******    *******   [            ]

[TYPE]            LIST           CLR_ASG
```

I/O Link Device (I/O Points Setup) Screen

```
I/O Link Device       G1        JOINT  10%
                                      1/2
  90-30 PLC
   Port Name                    Points
   Digital Input:                   0
   Digital Output:                  0

[TYPE]            LIST           CLR_ASG
```

Figure A–14. [6 SETUP]

General item setting screen

```
SETUP General                    JOINT  10%
                                      1/4
  1 Break on hold:         DISABLED
  2 Current language:      DEFAULT
  3 Ignore Offset command: DISABLED
  4 Ignore Tool_offset:    DISABLED

 TYPE                      ENABLED DISABLED
General
```

Tool frame entry screen

```
SETUP Frames                     JOINT  30%
   Tool Frame Setup/ Direct Entry    1/5
            X      Y      Z    Comment
     1:    0.0    0.0  140.5 *************
     2:    0.0    0.0  200.0 *************
     3:    0.0    0.0    0.0 *************
     4:    0.0    0.0    0.0 *************
     5:    0.0    0.0    0.0 *************
                   1 Tool Frame
                   2 Jog Frame
                   3 User Frame
 TYPE     DETAIL   OTHER    CLEAR   SETIND
Frames
```

Tool frame setup screen (Three Point Method)

```
SETUP Frames                     JOINT  30%
 Tool Frame Setup/ Three Point     3/4
 Frame Number: 3
   X:     0.0   Y:     0.0   Z:     0.0
   W:     0.0   P:     0.0   R:     0.0
  Comment:                 REF FRM
   Approach point 1:        RECORDED
   Approach point 2:        UNINIT
   Approach point 3:        UNINIT
             1 Three Point
             2 Six Point
             3 Direct Entry
 [TYPE]   METHOD  FRAME   MOVE_TO   RECORD
```

Tool frame setup screen (Six Point Method)

```
SETUP Frames                     JOINT  30%
 Tool Frame Setup/ Six Point        4/7
 Frame Numer: 3
   X:  100.0   Y:     0.0   Z:  300.0
   W:     0.0  P:     0.0   R:    0.0
  Comment:********************
   Approach point 1:        RECORDED
   Approach point 2:        RECORDED
   Approach point 3:        RECORDED
   Orien 1 Three Point      UNINIT
   X Dir 2 Six Point        UNINIT
   Z Dir 3 Direct Entry     UNINIT
 [TYPE]   METHOD  FRAME   MOVE_TO   RECORD
```

Tool frame setup screen (Direct Entry Method)

```
SETUP Frames                     JOINT  30 %
 Tool Frame Setup/ Direct Entry    3/7
 Frame Number: 3
  1  Comment:********************
  2  X:                        0.000
  3  Y:                        0.000
  4  Z:                      200.000
  5  W:                        0.000
  6  P:                        0.000
  7  R:  1 Three Point         0.000
     Con 2 Six Point      B, 0, , 0
         3 Direct Entry
 [TYPE]   METHOD  FRAME
```

User frame entry screen

```
SETUP Frames                     JOINT  30 %
   User Frame Setup/ Three Point     1/5
            X      Y      Z    Comment
     1: 1243.6    0.0   43.8 REF FRM
     2: 1243.6  525.2   43.8 *************
     3: 1243.6-525.2    43.8 *************
     4:    0.0    0.0    0.0 *************
     5:    0.0    0.0    0.0 *************
                  1 Tool Frame
                  2 Jog Frame
                  3 User Frame
 TYPE    DETAIL   OTHER    CLEAR   SETIND
Frames
```

Figure A–15. [6 SETUP]

User frame setup screen (Three Point Method)

```
SETUP Frames                        JOINT  30%
 User Frame Setup/ Three Point     3/4
 Frame Number: 1
   X: 1243.6    Y:    0.0   Z:   43.8
   W:    0.1    P:    2.3   R:    3.2
  Comment:********************
   Orient Origin Point:  RECORDED
   X Direction Point:    RECORDED
   Y Direction Point:    UNINIT
        1 Three Point
        2 Four Point
        3 Direct Entry

[TYPE]  METHOD  FRAME   MOVE_TO  RECORD
```

User frame setup screen (Four Point Method)

```
SETUP Frames                        JOINT  30%
 User Frame Setup/ Four Point      5/5
 Frame Number: 1
   X: 1243.6    Y:    0.0   Z:   10.0
   W:    0.1    P:    2.3   R:    3.2
  Comment:               REF FRM
   Orient Origin Point:  USED
   X Direction Point:    USED
   Y Direction Point:    USED
   Syst 1 Three Point    USED
        2 Four Point
        3 Direct Entry

[TYPE]  METHOD  FRAME   MOVE_TO  RECORD
```

Jog frame entry screen

```
SETUP Frames                        JOINT  30%
 Jog Frame Setup / Three Point     2/5
       X       Y       Z     Comment
    1: 1243.6   0.0    0.0 WORK AREA 1
    2: 1003.0 525.2   60.0 WORK AREA 2
    3: 1003.0 236.0   90.0 WORK AREA 3
    4:    0.0   0.0    0.0 *************
    5:    0.0   0.0    0.0 *************

                 1 Tool Frame
                 2 Jog Frame
                 3 User Frame

[TYPE]  DETAIL  OTHER   CLEAR  SETIND
```

Jog frame setup screen (Direct Entry Method)

```
SETUP Frames                        JOINT  30%
 Jog Frame Setup / Direct Entry    2/7
 Frame Number: 1
  1  Comment:   WORK AREA 1
  2      X:                  1243.600
  3      Y:                     0.000
  4      Z:                    10.000
  5      W:                     0.123
  6      P:                     2.340
  7      R:                     3.200
  8     1 Three Point
        2 Direct Entry

[TYPE]  METHOD  FRAME   MOVE_TO  RECORD
```

Reference position selection screen

```
REF POSN                            JOINT 30%

No.   Enb/Dsbl     @Pos      Comment
 1    ENABLE       TRUE      [ REFPOS1   ]
 2    DISABLE      FALSE     [ REFPOS2   ]
 3    DISABLE      FALSE     [ REFPOS3   ]



 TYPE             DETAIL  ENABLE   DISABLE
Ref Position
```

Reference position setting screen

```
REF POSN                            JOINT 30%
 Reference Position                     1/12
   Ref.Position Number
    1   Comment              [ REFPOS1     ]
    2   Enable/Disable         ENABLE
    3   Signal definition:         DO[ 0]
    4   J1    129.000       +/-     2.000
    5   J2    -31.560       +/-     2.000
    6   J3      3.320       +/-     2.000
    7   J4    179.240       +/-     2.000
    8   J5      1.620       +/-     2.000
    9   J6     33.000       +/-     2.000

[TYPE]                              RECORD
```

Figure A–16. [6 SETUP]

PNS Setup Screen

```
RSR/PNS                        G1       JOINT   10%
                                                7/8
   1 RSR or PNS                              [RSR]
   2 RSR1 program number [ENABLE ] [  12]
   3 RSR2 program number [ENABLE ] [  21]
   4 RSR3 program number [ENABLE ] [  33]
   5 RSR4 program number [ENABLE ] [  49]
   6 Base number                   [ 100]
   7 Acknowledge function         [TRUE ]
   8 Acknowledge pulse width(msec) [ 400]

   TYPE                      TRUE     FALSE
 RSR/PNS
```

PNS setting screen

```
 RSR/PNS                               JOINT 30%
                                             1/3
 1   RSR or PNS                          [ PNS]
 2   Base number                         [ 100]
 3   Acknowledge pulse width(msec)  [ 200]

   TYPE                        PNS       RSR
 RSR/PNS
```

Port selection screen

```
  SETUP Port Init                       JOINT 30%
   Connector port         Comment         1/3
     1   PORT            [ Handy file         ]

   TYPE           DETAIL
 Port Init
```

Port setting screen

```
 SETUP Port Init                        JOINT 30%
  PORT                                        1/3
    1     Device              [  Handy file ]
    2     Speed (Baud rate)          [ 9600 ]
    3     Parity bit                 [ None ]
    4     Stop bit                   [ 2bits]
    5     Time out value (sec)       [    0 ]

 [TYPE]           LIST   [CHOICE]
```

External override setting screen

```
  OVERRIDE SELECT                       JOINT 30%

   1    Function Enable:ENABLE

   2    Signal1:                    DI[  1][ ON]
   3    Signal2:                    DI[ 32][OFF]

        Signal1      Signal2      Override
   4      OFF          OFF            15%
   5      OFF          ON             30%
   6      ON           OFF            65%
   7      ON           ON            100%

   TYPE                      ENABLE   DISABLE
 Ovrd Select
```

Macro instruction setting screen

```
 Macro Command                            JOINT 30%
 Instruction name    Program    Assign1/20
  1 [ Open hand1  ] [HOPN1   ]     MF [ 1]
  2 [ Close hand1 ] [HCLS1   ]     MF [ 2]
  3 [ Relax hand1 ] [HRLX1   ]     MF [ 3]
  4 [ Open hand2  ] [        ]     MF [11]
  5 [ Close hand2 ] [        ]     MF [12]
  6 [ Relax hand2 ] [        ]     MF [13]
  7 [             ] [        ]        [  ]
  8 [             ] [        ]        [  ]
  9 [             ] [        ]        [  ]

   TYPE    CLEAR
 Macro
```

Figure A–17. [6 SETUP]

User alarm setting screen

```
Setting/User Alarm              JOINT  30%
                                      1/10
Alarm No.          User Message
  [1]:     [                            ]
  [2]:     [                            ]
  [3]:     [                            ]
  [4]:     [                            ]
  [5]:     [                            ]
  [6]:     [                            ]
  [7]:     [                            ]
  [8]:     [                            ]
  [9]:     [                            ]

  TYPE
User Alarm
```

Stroke Limit Setup Screen (Special machine only)

```
Stroke limit setup    G1       JOINT  10%
                                      1/4
     GROUP :1         AXIS :J1
 No. LOWER >-180.0    UPPER < 180.0
  1: 0.0      deg     0.0      deg
  2: 0.0      deg     0.0      deg
  3: 0.0      deg     0.0      deg
Default
  0: -180.0   deg     180.0    deg

Active limit:
 $MRR_GRP[1].$SLMT_J1_NUM = 0
  TYPE     GROUP#  AXIS#
Stroke limit
```

Preventive interference area schedule list screen

```
Rectangular Space      G1      JOINT  10%
LIST SCREEN                           1/3
 No.Enb/Dsbl     Comment         Usage
   1 ENABLE  [          ]Common Space
   2  DISABLE[          ]Common Space
   3  DISABLE[          ]Common Space

  TYPE             DETAIL  ENABLE  DISABLE
Space fnct.
```

Preventive interference area

```
Rectangular Space     G1        JOINT  10%
DETAILED SCREEN                       1/6

     SPACE :1         GROUP :1
     USAGE : Common Space

  1  Enable/Disable:    ENABLE
  2  Comment:         [**********]
  3  Output Signal:     DO [   0]
  4  Input  Signal:     DI [   0]
  5  Priority:          High
  6  inside/outside:    Inside

[TYPE]  SPACE            ENABLE  DISABLE
```

Preventive interference area setup screen

```
Rec                    G1      JOINT  10%
SPACE SETUP                           1/4

     SPACE :1            GROUP :1

     UFRAME :0           UTOOL :1
  1 :  BASIS VERTEX      [SIDE LENGTH  ]
  2 :X 0.0      mm       0.0      mm
  3 :Y 0.0      mm       0.0      mm
  4 :Z 0.0      mm       0.0      mm

[TYPE]   OTHER                    RECORD
```

Figure A–18. [6 SETUP]

Rectangular Space Check Schedule Detail Screen

```
 Program monitor       G1       JOINT  10%
                                      1/2
          CH Prog.  Status    Program
      1   WORKDROP  Running   SAMPLE1
      2   HANDCHCK  Paused    SAMPLE2




      TYPE    SYSTEM  RESTART  PAUSE     END

  Condition
```

Rectangular Space Check Space Definition Screen

```
 System  monitor       G1       JOINT  10%
                                      1/2
          CH Prog.  Status
      1   WORKDROP  Running
      2   HANDCHCK




  [TYPE]  PROGRAM           START    END
```

Figure A–19. [7 FILE]

File screen

```
 FILE                         JOINT  30%
                                    1/14
   1 *        *   (all files)
   2 *        KL  (all KAREL source)
   3 *        CF  (all command files)
   4 *        TX  (all text files)
   5 *        LS  (all KAREL listings)
   6 *        DT  (all KAREL data files)
   7 *        PC  (all KAREL p-code)
   8 *        MN  (all MN programs)
   9 *        TP  (all TP programs)
  10 *        VR  (all variable files)
 Press DIR to generate directory
 [TYPE]  [ DIR ]  LOAD  [BACKUP] [UTIL ]>

  DELETE   COPY  DISPLAY                 >
```

Figure A–20. [1 SELECT]

Program selection screen

```
Select                                  JOINT 30%
        58740bytes free                       3/5
 No.  Program name        Comment
  1  SAMPLE1          [SAMPLEPROGRAM1    ]
  2  SAMPLE2          [SAMPLEPROGRAM2    ]
  3  SAMPLE3          [SAMPLEPROGRAM3    ]
  4  PROG001          [PROGRAM001        ]
  5  PROG002          [PROG 1 comment
                            2 Protection
                            3 Last modifie
                            4 Size
                            5 Copy Sourse

[TYPE]   CREATE  DELETE  MONITOR  ATTR >

 COPY    DETAIL  LOAD      SAVE   PRINT>
```

Program registration screen

```
                                        JOINT 30%
  1  Words
  2  Upper Case
  3  Lower Case
  4  Options            ---Insert---
Select

   ---Create Teach Pendant Program---
  Program Name      [SAMPLE3     ]
  Sub type          [Jobs        ]
                                ---End---
Select function
        DETAIL    EDIT
```

Program information screen

```
Program detail                      JOINT  30%
                                         1/6
  Creation Date:           10-MAR-1994
  Modification Date:       11-MAR-1994
  Copy Source:       [****************]
  Positions: FALSE  Size:      312 Byte
   1  Program name:         [SAMPLE3 ]
   2  Sub Type:      [             None]
   3  Comment:       [SAMPLE PROGRAM 3]
   4  Group Mask:          [1,*,*,*,*]
   5  Write protect:        [      OFF]
   6  Ignore pause:         [      OFF]

    END      PREV     NEXT
```

Figure A–21. [2 EDIT]

Program edit screen

```
SAMPLE1                                JOINT 30%
                                             1/6
  1: J   P[1] 100% FINE
  2: J   P[2] 70% CNT50
  3: L   P[3] 1000cm/min CN
  4: L   P[4] 500mm/sec FIN  1 Insert
  5: J   P[1] 100% FINE      2 Delete
[End]                        3 Copy
                             4 Find
                             5 Replace
                             6 Renumber

[INST]                                EDCMD
 POINT                               TOUCHUP
```

Program edit screen/control instruction selection screen

```
Instruction                                JOINT 30%
  1 Registers          5 JMP/LBL
  2 I/O                6 CALL
  3 IF/SELECT          7 Palletizing
  4 WAIT               8 --next page--
PROGRAM1
                                               3/4
   3: L P[3] 500mm/sec CNT10
 [End]



[INST]                                   [EDCMD]
```

Figure A–22. [3 DATA]

Register screen

```
DATA Registers                  JOINT  30%
                                      1/32
R[  1:                ]=0
R[  2:                ]=5
R[  3:                ]=12
R[  4:                ]=50000
R[  5:                ]=0
R[  6:                ]=0
R[  7:                ]=0
R[  8:                ]=0
R[  9:                ]=0

TYPE
```

```
Registers
```

Position register screen

```
DATA Position Reg               JOINT  30%
                                      1/10
PR[  1:REF POS 1      ]=*
PR[  2:REF POS 2      ]=*
PR[  3:REF POS 3      ]=*
PR[  4:REF POS 4      ]=*
PR[  5:               ]=*
PR[  6:               ]=*
PR[  7:               ]=*
PR[  8:               ]=*
PR[  9:               ]=*

 TYPE           RECORD POSITION  CLEAR
```

```
Position Reg
```

Position register screen/position data information screen

```
Position Detail                 JOINT 30%
PR[1]       UF:F UT:1    CONF:N 00
 X: 1500.374  mm    W:   40.000  deg
 Y: -342.992  mm    P:   10.000  deg
 Z:   956.895  mm    R:   20.000  deg
DATA Position Reg
                                      1/10
   PR  [ 1:ZERO POS        ]  = R
   PR  [ 2:                ]  = *
   PR  [ 3:                ]  = *
   PR  [ 4:                ]  = *

Enter value
                   CONFIG  DONE   [REPRE]
```

Figure A–23. [3 DATA]

Palletizing register screen

```
DATA: Pallet Regist              JOINT  30%
                                       1/32
 PL  [ 1:pallet work     ]  =  [ 1,  1,  1 ]
 PL  [ 2:                ]  =  [ 1,  1,  1 ]
 PL  [ 3:                ]  =  [ 1,  1,  1 ]
 PL  [ 4:                ]  =  [ 1,  1,  1 ]
 PL  [ 5:                ]  =  [ 1,  1,  1 ]
 PL  [ 6:                ]  =  [ 1,  1,  1 ]
 PL  [ 7:                ]  =  [ 1,  1,  1 ]
 PL  [ 8:                ]  =  [ 1,  1,  1 ]
 PL  [ 9:                ]  =  [ 1,  1,  1 ]


[TYPE]
```

Figure A–24. [4 STATUS]

Robot axis status screen (Status 1 screen)

```
STATUS Axis                         JOINT 30%

                                       GRP[1]
       Flag Bits1/2       History
  J1: 0000000000000000 (0000000000000000)
      0000000000000000 (0000000000000000)
  J2: 0000000000000000 (0000000000000000)
      0000000000000000 (0000000000000000)
  J3: 0000000000000000 (0000000000000000)
      0000000000000000 (0000000000000000)
  J4: 0000000000000000 (0000000000000000)
      0000000000000000 (0000000000000000)

  TYPE   STATUS1  STATUS2  PULSE  [ UTIL ]

Axis
```

Robot axis status screen (Torque monitor screen)

```
STATUS Axis                         JOINT 30%

                                       GRP[1]
       Torque Monitor
     Ave.    /     Max.      Inpos OT VRDY
  J1:      0.000/      0.000   1    0   OFF
  J2:      0.000/      0.000   1    0   OFF
  J3:      0.000/      0.000   1    0   OFF
  J4:      0.000/      0.000   1    0   OFF
  J5:      0.000/      0.000   1    0   OFF


[TYPE]  MONITOR  TRACKING DISTURB [ UTIL ]
```

Software version screen (Software version screen)

```
STATUS Version ID                 JOINT  10%
     SOFTWARE:                  ID:    1/11
  1  FANUC LR Tool              M6.1011
  2  S/W Serial No.             007051
  3  Controller ID              F00000
  4  Default Personality (from FD)
  5  LRM100iB                   M6.1011
  6  Servo Code                 V10.09S
  7  Cart. Mot. Parameter       V3.00
  8  Joint Mot. Parameter       V3.00
  9  Software Edition No.       M6.1011
 10  Boot MONITOR               M6.1011
 11  Teach Pendant              7D01/09I

 [ TYPE ] SOFTWARE CONFIG  MOTOR   SERVO

Version ID
```

Software version screen (Motor ID screen)

```
STATUS Version ID                 JOINT  10 %
   GR: AX: MOTOR ID AND INFO:       1/16
 1  1   1  ACbM0.5/4000 20A  H1 DSP1-L
 2  1   2  ACbM0.5/4000 20A  H2 DSP1-M
 3  1   3  ACbM0.5/4000 10A  H3 DSP2-L
 4  1   4  ACbM0.4/4000 10A  H4 DSP2-M
 5  1   5  ACbM0.4/4000 10A  H5 DSP3-L
 6 **  **  ****************************
 7 **  **  ****************************
 8 **  **  ****************************
 9 **  **  ****************************
10 **  **  ****************************

 [ TYPE ]SOFTWARE CONFIG  MOTOR   SERVO
```

Program Timer List Screen

```
PRG TIMER LISTING      G1        JOINT  10%
                                      1/10
                     count     comment
 1 Timer[ 1]       4.01(s)[            ]
 2 Timer[ 2]       6.01(s)[            ]
 3 Timer[ 3]       6.01(s)[            ]
 4 Timer[ 4]       4.01(s)[            ]
 5 Timer[ 5]       2.00(s)[            ]
 6 Timer[ 6]       0.00(s)[            ]
 7 Timer[ 7]       0.00(s)[            ]
 8 Timer[ 8]       0.00(s)[            ]
 9 Timer[ 9]       0.00(s)[            ]

  TYPE    DETAIL

Prg Timer
```

Program Timer Detail Screen

```
PRG TIMER DETAIL       G1        JOINT  10%
                                       1/1

    Timer[ 1]
     Comment        :[*************]
     Count          :    4.01(sec)

     Start program :[     TEST2]
           Line     :          1
     Stop  program :[     TEST2]
           Line     :          6


[TYPE]  LISTING
```

Figure A–25. [4 STATUS]

System Timer Screen

```
 SYS TIMER               G1       JOINT  10%
                                       1/4

     GROUP : 1
       Timer type  Total(h)  Lap(m)
     On Power time:     2.3      0.0[OFF]
     Servo on time:     2.3      0.0[OFF]
     Running  time:     0.0      0.0[OFF]
     Waiting  time:     0.0      0.0[OFF]

      TYPE   GROUP#  ON/OFF  RESET
 Sys Timer
```

Safety Signal Status Screen

```
STATUS Safety                     JOINT  10 %
    SIGNAL NAME                 STATUS   1/10
  1 SOP E-Stop:                  FALSE
  2 TP E-Stop:                   TRUE
  3 Ext E-Stop:                  FALSE
  4 Fence Open:                  FALSE
  5 TP Deadman:                  FALSE
  6 TP Enable:                   FALSE
  7 Hand Broken:                 FALSE
  8 Overtravel:                  FALSE
  9 Low Air Alarm:               FALSE
 10 SVOFF Input:                 FALSE

[ TYPE ]
Safety Signl
```

Execution history screen

```
 Execution history            JOINT VFINE
                                      1/6
     Program name  Line. Dirc.  Stat.
   1 PNS0001          3  FWD    Done
   2 PNS0001          6  BWD    Paused
   3 PNS0001          7  FWD    Paused
   4 PNS0001          6  FWD    Done
   5 PNS0001          5  FWD    Done

      TYPE                           CLEAR
 Exec-hist
```

Memory status list screen

```
STATUS Memory                    JOINT  10 %
                Total      Available
   Pools      ----------------------
   TPP  CMOS   550.0 KB     540.8 KB
   PERM CMOS   999.8 KB     363.5 KB
   TEMP DRAM  1726.9 KB    1207.7 KB

    Description:
    TPP:   Used by .TP, .MR, .JB, .PR
    PERM:  Used by .VR, RD:, Options
    TEMP:  Used by .PC, .VR, Options

      TYPE  DETAIL                   HELP
   Memory
```

Memory status detail screen

```
STATUS Memory                    JOINT  10 %
           Total       Free   Lrgst Free
 Pools   -----------------------------
   TPP     550.0 KB  529.2 KB  529.2 KB
   PERM    999.8 KB  367.1 KB  367.1 KB
   SYSTEM 1010.4 KB   59.2 KB   59.2 KB
   IMAGE   255.9 KB   76.9 KB   76.5 KB
   TEMP   1726.9 KB 1175.4 KB 1172.6 KB
 Hardware -----------------------------
   FROM     16.0 MB
   DRAM     16.0 MB
   CMOS      1.0 MB

[TYPE]   BASIC                     HELP
```

Figure A–26. [5 POSITION]

Current position screen (Joint coordinates)

```
POSITION                         JOINT 30%
 Joint                              TOOL:1

 J1:  90.045   J2:  42.195   J3:  29.336
 J4:  40.000   J5:  10.000
 E1:  *****



   TYPE     JNT     USER     WORLD

Position
```

Current position screen (User coordinates)

```
POSITION                         JOINT 30%
User                       Frame:1   TOOL:1
      Configuration: N,0,,0
  X: 1500.374  Y: -342.992  Z: 956.895
  W:    40.000  P:    10.000  R:  20.000
  E:    *****



 TYPE      JNT      USER     WORLD
```

Figure A–27. [6 SYSTEM]

System clock screen

```
SYSTEM Clock                  JOINT  10 %
Clock Display


    DATE                     95/03/22

    TIME                     18:24:38




Please select function
  TYPE                       ADJUST

 Clock
```

System variable screen

```
SYSTEM Variables                 JOINT 10%
                                      1/98
  1  $ACC_MAXLMT           500
  2  $ACC_MINLMT           0
  3  $ANGTOL               [9] of Real
  4  $APPLICATION          [3] of STRING[21]
  5  $AP_CUREQ             [3] of APCVREQ_T
  6  $AP_CURTOOL           0

   TYPE

 Variables
```

Positioning screen

```
MANUAL OT RELEASE                  JOINT  30%

  AXIS      OT MINUS     OT PLUS
  1            --          TRUE
  2            --          FALSE
  3            --           --
  4            --           --
  5            --           --
  6            --           --
  7            --           --
  8            --           --
  9            --           --

  TYPE    RELEASE

 OT Release
```

Positioning screen

```
  SYSTEM Master/Cal                JOINT  30 %

   1 FIXTURE POSITION MASTER
   2 ZERO POSITION MASTER
   3 QUICK MASTER
   4 SINGLE AXIS MASTER
   5 SET QUICK MASTER REF
   6 CALIBRATE

    Press 'ENTER' or number key to select.



   TYPE     LOAD   RES_PCA          DONE

 Master/Cal
```

Figure A–27.(Continued) [6 SYSTEM]

Joint operating area setting screen

```
 SYSTEM Axis Limits              JOINT  10 %
   AXIS   GROUP       LOWER      UPPER  1/16

    1       1        -150.00      150.00 dg
    2       1         -60.00      100.00 dg
    3       1        -110.00       50.00 dg
    4       1        -240.00      240.00 dg
    5       1        -120.00      120.00 dg
    6       0           0.00        0.00 mm
    7       0           0.00        0.00 mm
    8       0           0.00        0.00 mm
    9       0           0.00        0.00 mm

      TYPE

 Axis Limits
```

System configuration screen

```
 System/Config                          JOINT 30%
                                             1/37
  1 Use HOT START:                        FALSE
  2 I/O power fail recovery: RECOVER ALL
  3 Autoexec program                 [********]
            for Cold start:
  4 Autoexec program                 [********]
            for Hot start:
  5 HOT START done signal:             DO[0]
  6 Restore selected program:          TRUE
  7 Enable UI signals :                TRUE
  8 START for CONTINUE only :         FALSE
  9 CSTOPI for ABORT :                FALSE
 10 Abort all programs by CSTOPI : FALSE
 11 PROD_START depend on PNSTROBE :FALSE
 12 Detect FAULT_RESET signal :        FALL
 13 Use PPABN signal :             <*GROUPS*>
 14 WAIT timeout :                  30.00 sec
 15 RECEIVE timeout :               30.000 sec
 16 Return to top of program :         TRUE
 17 Original program name (F1) : [PRG     ]
 18 Original program name (F2) : [MAIN    ]
 19 Original program name (F3) : [SUB     ]
 20 Original program name (F4) : [TEST    ]
 21 Original program name (F5) : [*******]
 22 Default logical command :  <*DETAIL*>
 23 Muximum of ACC instruction :   150
 24 Minimum of ACC instruction :     0
 25 WJNT for default motion :      ******
 26 Auto display of alarm menu :   FALSE
 27 Force Message :                ENABLE
 28 Reset CHAIN FAILURE detection : FALSE
 29 Allow Force I/O in AUTO mode  : TRUE
 30 Allow chg. ovrd. in AUTO mode : TRUE
 31 Signal to set in AUTO mode DOUT [  0]
 32 Signal to set in T1 mode DOUT   [  0]
 33 Signal to set in T2 mode DOUT   [  0]
 34 Signal to set if E-STOP DOUT    [  0]
 35 Hand broken :                  <*GROUPS*>
 36 Remote / Local setup :          Remote
 37 External I/O (ON : Remote) : DI [  0]

[TYPE]                  [CHOICE]
```

A.3 List of Program Instructions

Figure A–28. Motion (option) instruction

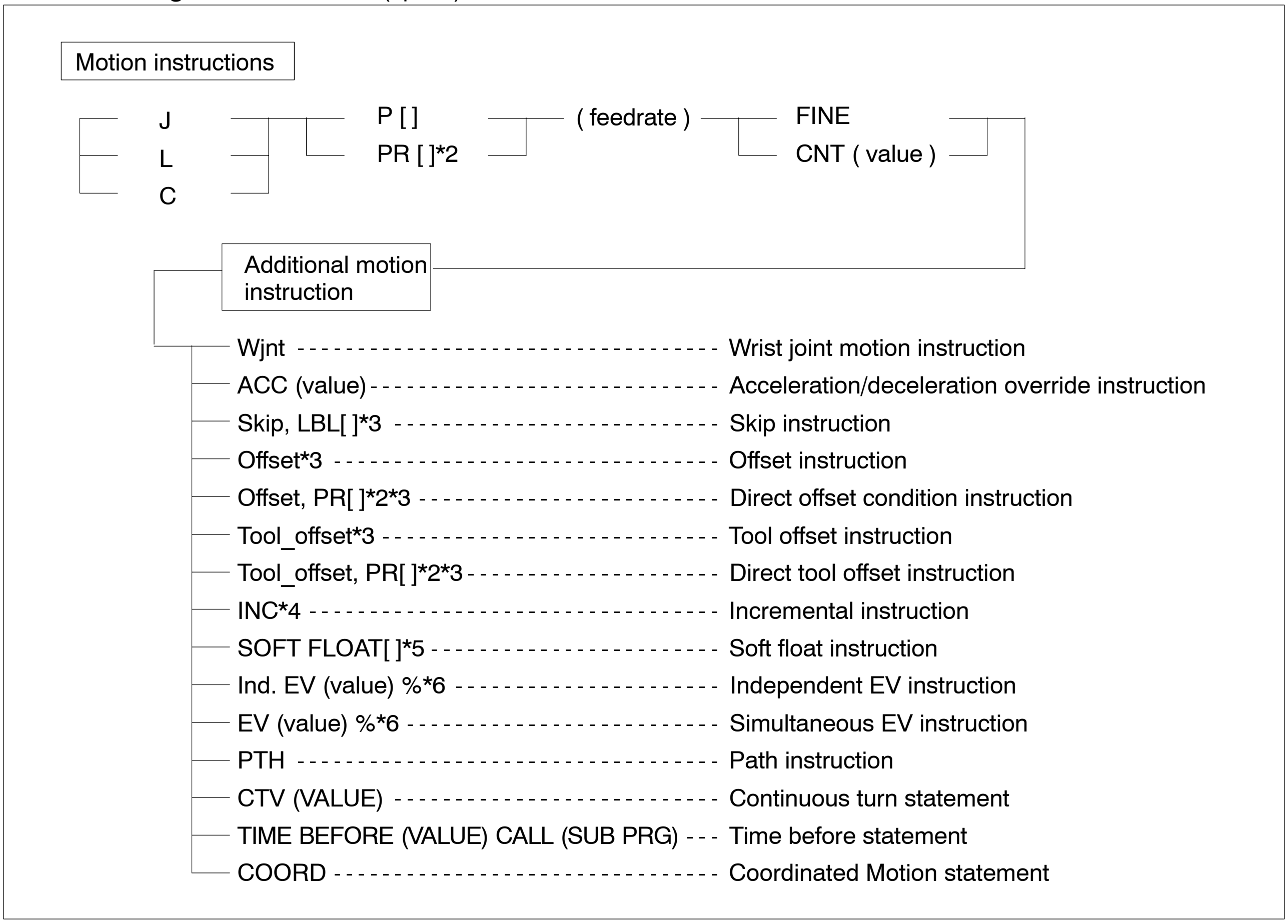


Figure A–29. Program instruction menu

Program instruction menu	Program instruction format	Program instruction
1 Register	R [ ] = / PR [i,j]*2 → R [ ]	Register instruction
	Const	
	SDI / O [ ]	
	RDI / O [ ]	
	GI / O [ ]	
	AI / O [ ]*9	
	PR [ ]*2 = ...	Position register instruction
	PL [ ] = ...	Palletizing register instruction
2 I / O	DO [ ] = ...	I/O instruction
	RO [ ] = ...	
	GO [ ] = ...	
	AO [ ]*9=	
3 IF / SELECT	IF → R [ ] = ...	Comparison
	I / O [ ] = ...	
	PL [ ] *1	
	SELECT R [ ] = ...	Selection condition instruction
4 WAIT	WAIT (time)	Time wait instruction
	WAIT → R [ ] = ...	Conditional wait instruction
	I / O [ ] = ...	
5 JMP / LBL	LBL [ ]	Label instruction
	JMP LBL [ ]	Jump instruction
6 CALL / END	CALL (Program)	Program call instruction
	END	Program end instruction

Figure A–30. Program instruction menu

Program instruction menu	Program instruction format	Program instruction
7 Palletizing*1	Palletizing – B	Palletizing instruction
	PALLETIZING – BX	Palletizing BX statement
	PALLETIZING – E	Palletizing E statement
	PALLETIZING – EX	Palletizing EX statement
	Palletizing – END	Palletizing end instruction
8 Program Control	PAUSE	Halt instruction
	ABORT	Abort instruction
9 Miscellaneous	RSR [ ] = ...	RSR statement
	UALM [ ]	User alarm instruction
	TIMER [ ]	Timer instruction
	OVERRIDE	Override instruction
	Remark	Comment instruction
	Message [ ]	Message instruction
	$ (Parameter) = ...	
10 Skip*3	SKIP CONDITION — I/O [ ] = ...	Skip condition instruction
11 Offset*1	OFFSET CONDITION PR [ ] *2=	Offset condition instruction
	UFRAME [ ] = ...	User frame setup instruction
	UFRAME_NUM = ...	User frame selection instruction
	UTOOL [ ] = ...	Tool frame setup instruction
	UTOOL_NUM = ...	Tool frame selection instruction
12 Tool_Offset*3	TOOL_OFFSET CONDITION PR [ ] *2=	Tool offset condition instruction
13 MACRO*3		Macro instruction
14 Multiple control	RUN (PRG)	Run program statement
	SEMAPHORE [ ]=....	Semaphore statement
	WAIT SEMAPHORE [ ]=....	Wait semaphore statement
15 SENSOR	SEND R[]	Send register statement
	RCV R[], LBL[]	Receive register statement
16 LOCK PREG*2	LOCK PREG	Position register lock instruction
	UNLOCK PREG	Position register unlock instruction
17 SOFTFLOAT*5	SOFTFLOAT [ ]	Soft float start instruction
	SOFT FLOAT END	Soft float end instruction
	FOLLOW UP	Follow–up instruction
18 MONITOR /MON.END	MONITOR	Monitor start statement
	MONITOR END	Monitor end statement

Program instruction menu	Program instruction format	Program instruction
18 Independent GP*12	Independent GP	Independent motion group statement
19 Simultaneous GP*12	Simultaneous GP	Simultaneous motion group statement
20 Payload	PAYLOAD [ ]	Payload setting statement
21 TORQ_LIMIT*13	TORQ_LIMIT (VALUE) %	Torque limit statement

Option

The items indicated by * (numeral) in Figure A–28 to A–30 are displayed when the corresponding option is added, as indicated below:

Table A–3. Option list

*	Option	Specification
1	Palletizing	A05B–2440–J500
2	Position register	A05B–2440–H510
3	Option command	A05B–2440–H510
4	Incremental input	A05B–2440–J510
5	Softfloat	A05B–2440–J612
6	Extended axis control	A05B–2440–J518
7	Continuous turn	A05B–2440–J613
8	Condition monitor function	A05B–2440–H510
9	Analog I/O	A05B–2440–H550
10	Multi task	A05B–2440–J600
11	Sensor interface	A05B–2440–J502
12	Multi motion group	A05B–2440–J601
13	Torque limit	A05B–2440–J611
14	Coordinated motion	A05B–2440–J619

A.4 Program Instructions

A.4.1 Motion instructions

Table A–4. Motion instructions

Motion format	J	Enables robot operation for each joint with interpolation.
	L	Moves the robot tool linearly.
	C	Enables the tool tip of the robot to make a circular motion.
Position variable	P[i : Comment]	Standard variable for storing position data.
	PR[i : Comment]	Register for storing position data. i: 1 – 10
Feedrate unit	%	Specify the rate of a feedrate to the highest feedrate of the robot.
	mm/sec, cm/min, inch/min, deg/sec	Specifies the speed with which the tool tip makes a linear or circular motion.
	sec	Specify the time required during a motion.
Positioning path	FINE	The robot stops at the specified position and starts the next motion.
	CNTn n (0 – 100):	The robot moves gradually from the specified position to the position at which the next motion starts.!Degree of gradual motion. The higher the specified number, the more gradual the robot moves.

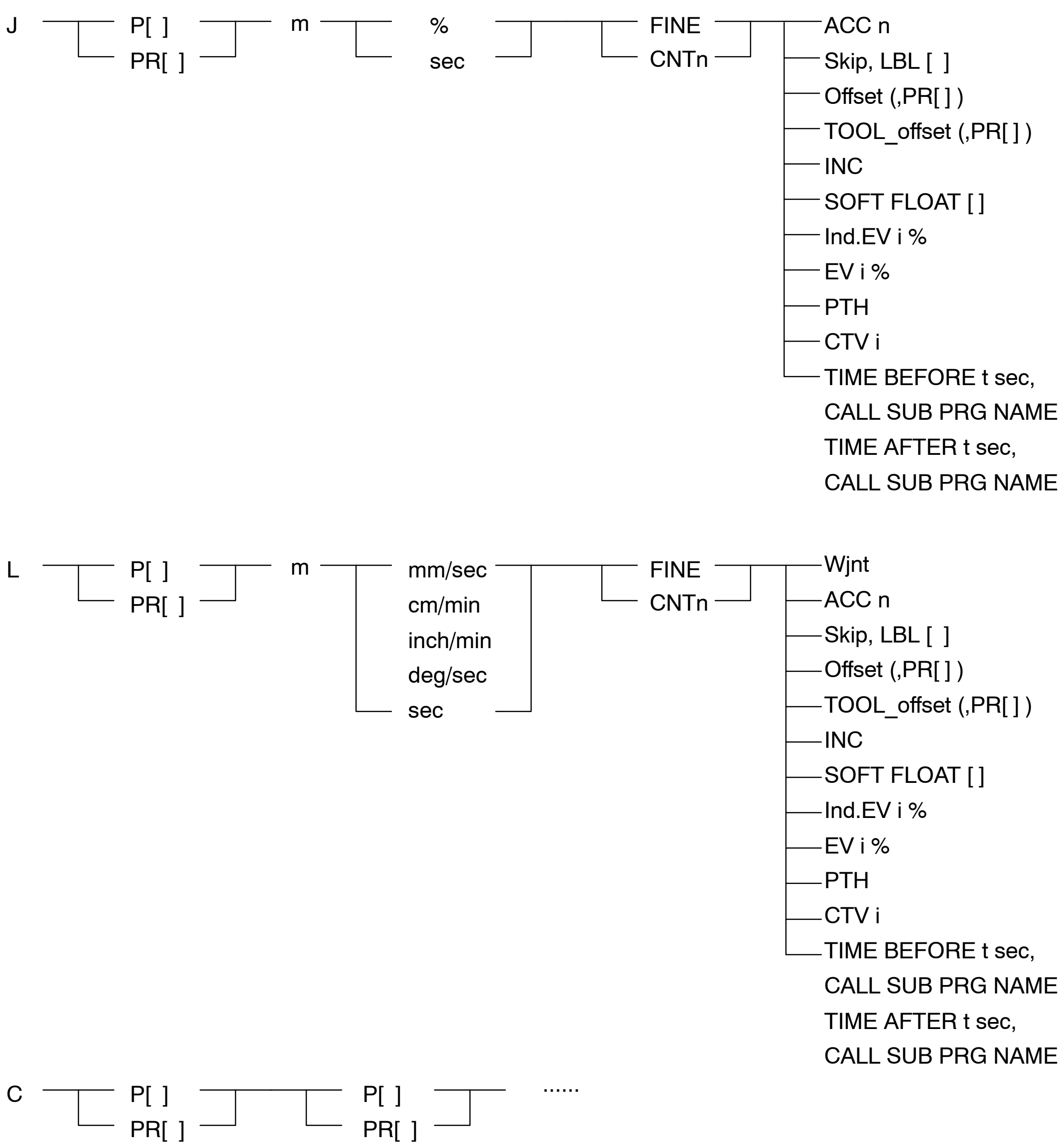
A.4.2 Additional motion instructions

Table A–5. Additional motion instructions

Wrist joint motion	Wjnt	On a linear or arc motion, the wrist axis moves with a joint motion, and the joint coordinates vary.
Acceleration/deceleration override	ACC a a=0 to 500(%)	Sets the rate of acceleration/deceleration when moving.
Skip	Skip, LBL[ ]	Causes a branch to the specified label when the condition specified in a skip condition instruction is not satisfied. When the condition is satisfied, cancels the motion and executes the next line.
Positional offset	Offset	Makes the robot move to the position where the value specified by the offset condition instruction is added to the positional variable.
	Offset,PR[i:comment]	Makes the robot move to the position where the value specified by the offset condition instruction and the value of position register are added to the positional variable.
Tool offset	Tool_offset	Moves the robot to the position corresponding to the value specified by the tool offset instruction, added to the position variable.
	Tool_offset,PR[(GPk:)i]	Moves the robot to the position corresponding to the position register value, added to the position variable.
Incremental	INC	Makes the robot move to the position where the value of the position variable is added to the current position.
Soft float	SOFT FLOAT[i]	Enables the soft float function.
Independent EV	Ind.EV(i)% i = 1 to 100 (%)	Moves the extended axis, independently of the robot motion.
Simultaneous EV	EV(value)% i = 1 to 100 (%)	Moves the extended axis, synchronized with the robot.
Path	PTH	Creates a motion plan, using the rate attainable in continuous operation.

Table A–5. (Cont'd) Additional motion instructions

Continuous turn	CTV i i = –100 to 100(%)	Start the execution of continuous turn.
Before execution	TIME BEFORE t CALL <prog> TIME AFTER t CALL <prog>	Before or after the specified end time, call a sub program and execute one. t=Excecution start time. <prog>=Name of sub program.



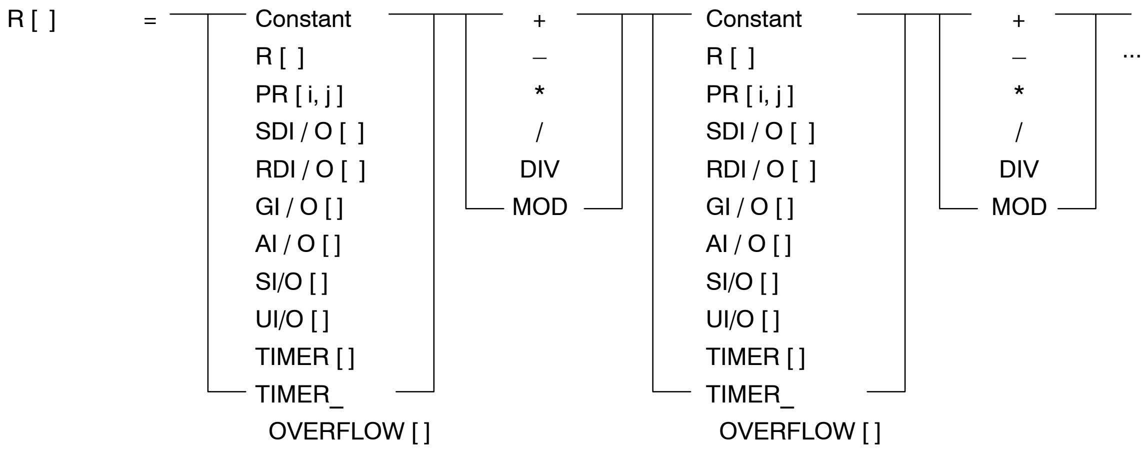
Example

```
1: J  P[1] 100% FINE
2: L  P[2:LINE] 500mm/sec CNT100 Wjnt
 : Offset, PR[1]
3: L  P[3] 3.5sec CNT100 INC
4: L  P[4] 100cm/min FINE Wjnt
 :    Skip,LBL[100]
5: C  P[5]
      P[6] 300mm/sec CNT50
```

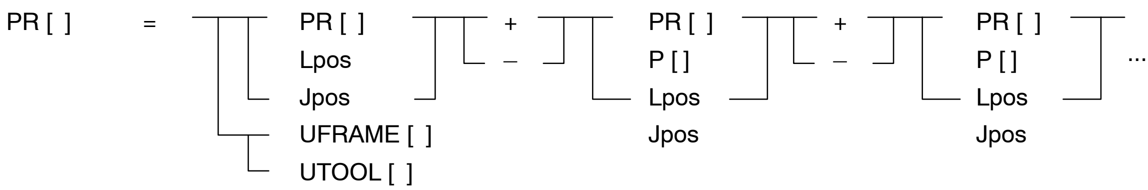
A.4.3 Register and I/O instructions

Table A–6. Register and I/O instructions

Register	R[i] i: 1 to 32	i: Register number.
Position register	PR[(GPk:) i] PR[(GPk:) i , j]	Fetches a position data element. i: Position register number. i: 1 to 10 j: Number of an element in a position register. j: 1 to 9 .
Position data	P[i :comment] Lpos Jpos UFRAME [i] UTOOL [i]	i: Position number. i: 1 to memory limit Cartesian coordinates of the current position Joint coordinates of the current position User coordinate system Tool coordinate system
Input/output signal	SDI[i], SDO[i] RDI[i], RDO[i] GI[i], GO[i] AI[i], AO[i]	System digital signals Robot digital signals Gourp signals Analog signals



Example
```
1: R[1]   = RI[3]
2: R[3:] = DI[4]*PR[1,2]
```



Example
```
1: PR[1] = PR[6]
2: PR[3] = PR[4]+Lpos
3: PR[8] = UFRAME[1]
4: PR[9] = UTOOL[2]
```

```
PR [ i, j ]  =  Constant              +    Constant              +   ...
                R [ ]                 –    R [ ]                 –
                PR [ i, j ]           *    PR [ i, j ]           *
                SDI / O [ ]           /    SDI / O [ ]           /
                RDI / O [ ]          MOD   RDI / O [ ]          MOD
                GI / O [ ]           DIV   GI / O [ ]           DIV
                AI / O [ ]                 AI / O [ ]
                SI/O [ ]                   SI/O [ ]
                UI/O [ ]                   UI/O [ ]
                TIMER [ ]                  TIMER [ ]
                TIMER_                     TIMER_
                 OVERFLOW [ ]               OVERFLOW [ ]
```

```
SDO [ ]  =  ON                       GO [ ]  =  Constant
SRO [ ]     OFF                      AO [ ]     R [ ]
            R [ ]
            Pulse ( , width )
```

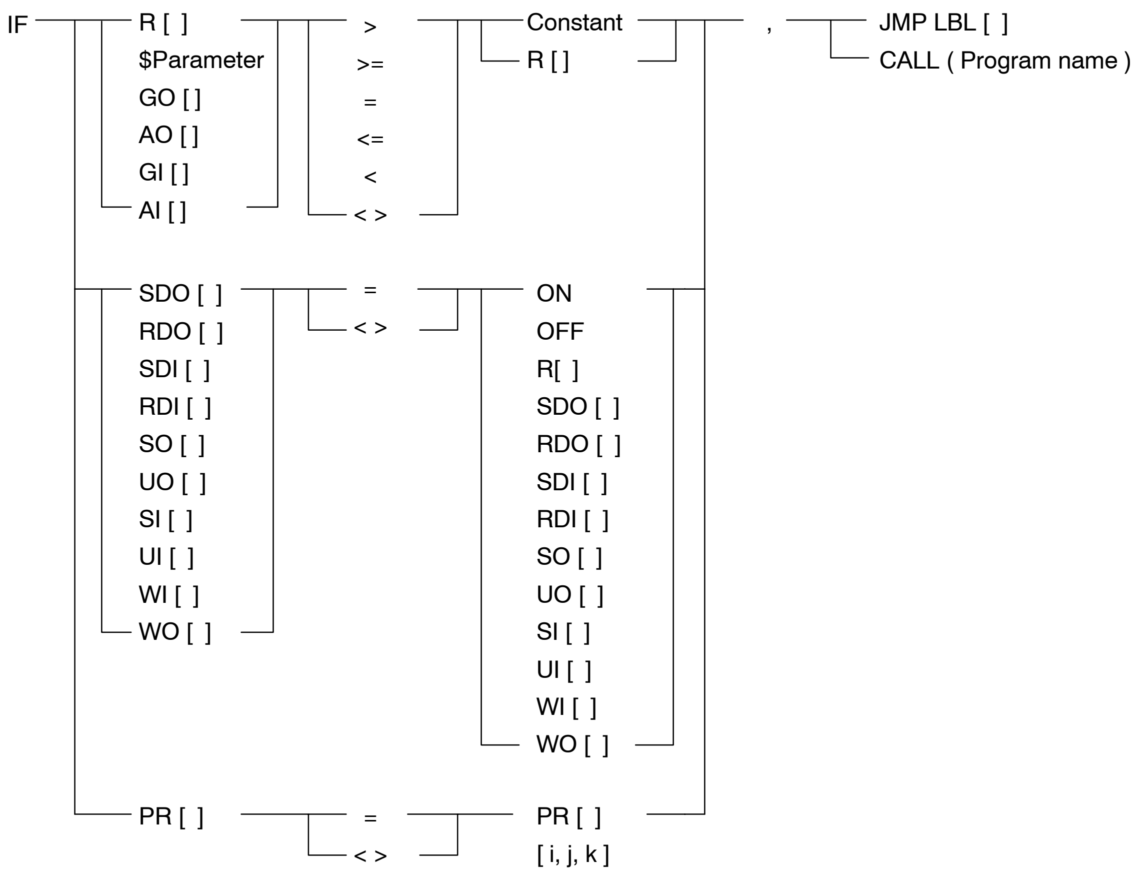
Example

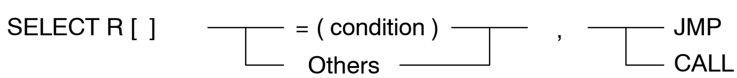
```
1: SDO[1]   = ON
2: RDO[3:]  = PULSE,1.0sec
3: RDO[4]   = R[1]
4: GO[9]    = R[2:control]
5: AO[10]   = 12.5
```

A.4.4 Conditional branch instructions

Table A–7. Conditional branch instructions

Comparison condition	IF (condition) (branch)	Specifies a comparison condition and an instruction or program to which the program branches to. You can link (Conditions) by using operators.
Selection condition	SELECT R[i] = (value) (branch)	Specifies a selection condition and an instruction or program to which the program branches to.





Example

```
1: IF R[2] >= R[3],LBL[1:HANDOPEN]
2: IF DI[2] = ON,CALL SUBPROGRAM
```

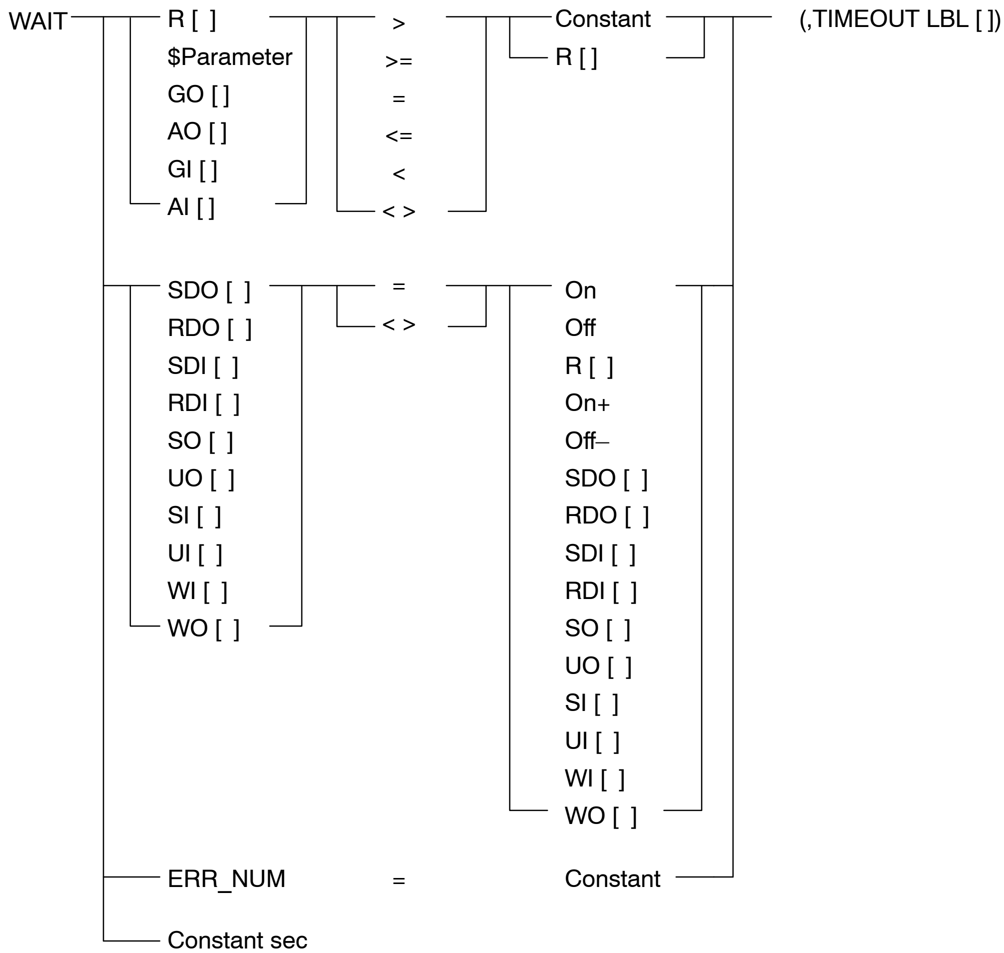
Example

```
3: SELECT R[2] = 1,JMP LBL[1]
4:    = 2,JMP LBL[2]
5:    = 3,JMP LBL[3]
6: ELSE,CALL MAINPROG
```

A.4.5 Wait instruction

Table A–8. Wait instruction

Wait	WAIT < condition > WAIT < time >	Waits until the specified condition is satisfied or until the specified time has elapsed. You can link (Conditions) by using operators.



Example

```
1: WAIT RDI[1] = ON
2: WAIT 10.5sec
3: WAIT R[2],TIMEOUT,LBL[1]
```

A.4.6 Unconditional branch instructions

Table A–9. Unconditional branch instructions

Label	LBL [i : COMMENT] JMP LBL[i]	Specifies the instruction which the program branches to. Causes a branch to the specified label.
Program call	CALL (program–name)	Causes a branch to the specified program.
Program end	END	Ends the program and returns control to the calling program.

JMP LBL[ ]

CALL (Program name) LBL[ ]

Example

```
1: LBL[1: HANDCLOSE]
2: JMP LBL[2]
3: CALL SAMPLE1
4: CALL PRG2: LBL[1]
```

A.4.7 Program control instructions

Table A–10. Program control instructions

Halt	PAUSE	Halts a program.
Abort	ABORT	Aborts a program.

Example
```
1: PAUSE
2: ABORT
```

A.4.8 Other instructions

Table A–11. Other instructions

RSR	RSR[i]	Enables or disables RSR signals (i = 1 to 4).
User alarm	UALM[i]	Displays a user alarm on the alarm line.
Timer	TIMER[i]	Sets the timer.
Override	OVERRIDE	Sets override.
Comment	!(comment)	Inserts a comment in a program.
Message	MESSAGE [message–text]	Displays a user message on a user screen.
Parameter	$(system variable name)	Changes the value of a system variable.
Maximum speed	JOINT_MAX_SPEED [ ] LINEAR_MAX_SPEED	Sets the maximum speed for operation statements in the program.

RSR [ ] = ENABLE / DISABLE

TIMER [ ] = START / STOP / RESET

OVERRIDE = m%

$(system variable name) = Constant / R [ ] / PR [ ]

R [ ] / PR [ ] = $ (Parameter)

JOINT_MAX_SPEED [ ] = Constant / R [ ]

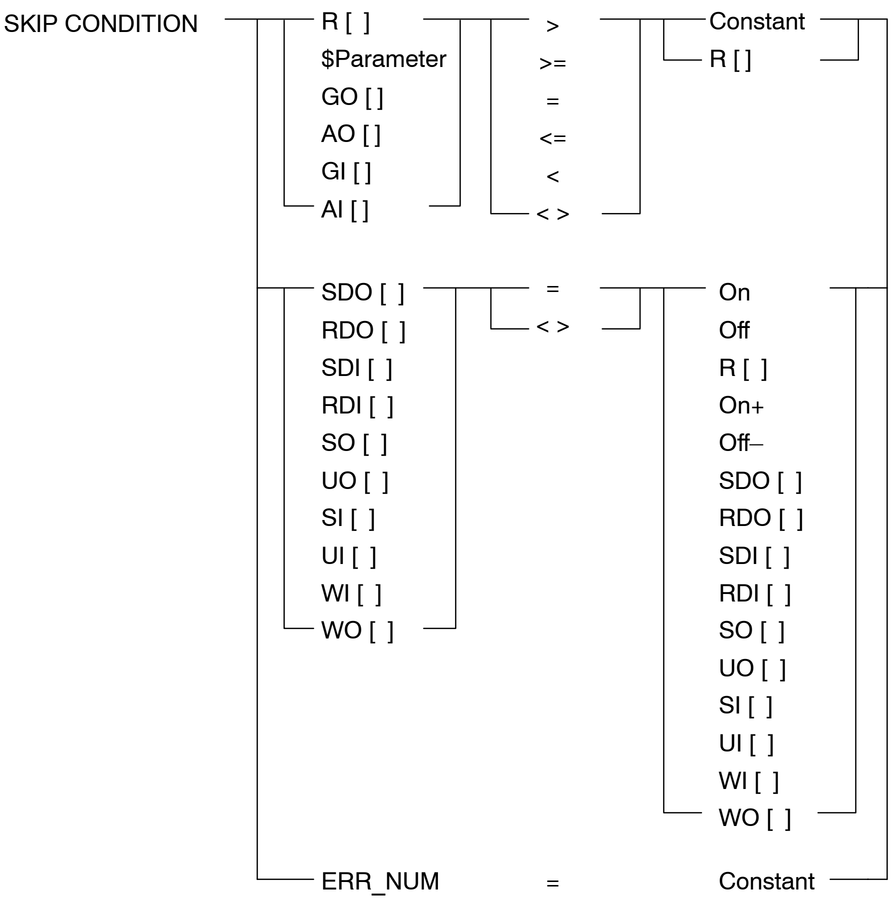
LINEAR_MAX_SPEED [ ] = Constant / R [ ]

Example
```
 3: OVERRIDE = 50%
 4: $NRM_TURN = 1
 5: $NRM_TURN = 1
 6: RSR[2] = ENABLE
 7: UALM [3:no work]
 8: ! STEP 2 START
 9: TIMER[1] = START
10: MESSAGE [STEP1 EXECUTION]
11: $NRM_TURN = 1
```

A.4.9 Skip and Offset condition instruction

Table A–12. Skip and Offset condition instruction

Skip condition	SKIP CONDITION (condition)	Specifies the skip execution condition for an additional motion instruction. You can link (Conditions) by using operators.
Offset condition	OFFSET CONDITION (offset amount)	Specifies the amount of offset used by the motion instruction.
Tool offset condition	TOOL_OFFSET CONDITION (offset amount)	Specifies the amount of tool offset used by the motion instruction.



OFFSET CONDITION PR[ ] (, UFRAME[ ])

TOOL OFFSET CONDITION PR[ ] (, UTOOL[ ])

Example

```
1: SKIP CONDITION  SDI[1] = ON
2: SKIP CONDITION  RDI[2] <> DI[3]
3: OFFSET CONDITION PR[1],UFRAME[1]
4: TOOL OFFSET CONDITION PR[2],UTOOL[1]
```

A.4.10 Frame setup instruction

Table A–13. Frame setup instruction

User frame User frame selection	UFRAME[i] UFRAME_NUM	User frame i=1 to 9 The number of current user frame.
Tool frame Tool frame selection	UTOOL[ ] UTOOL_NUM	Tool frame. i=1 to 9 The number of current tool frame

```
UFRAME[ ]   ─┬─  =  ─┬─ P[ ]
UTOOL[ ]    ─┘       └─ PR[ ]

UFRAME_NUM  ─┬─  =  ─┬─ Constant
UTOOL_NUM   ─┘       └─ R[ ]
```

Example

```
1: UFRAME[1] = P[12]
2: UTOOL[3]=PR[1]
3: UFRAME_NUM=3
```

A.4.11 Macro instruction

Table A–14. Macro instruction

Macro	(macro–instruction)	Executes a program defined on the macro instruction setting screen.

Example

```
1: HAND1 OPEN
2: HAND2 CLOSE
```

A.4.12 Multiaxis control instructions

Table A–15. Multiaxis Control Instructions

Program execution	RUN	Starts execution of a specified program in another motion group.
Semaphore variable	SEMAPHORE[i]	Defines a semaphore variable (i = 1 to 32). If the semaphore is off, other tasks cannot be executed.

```
SEMAPHORE [ ] = ─┬─ On
                 └─ Off
```

WAIT SEMAPHORE [ ]   (,TIMEOUT LBL [ ])

RUN (Program Name)

Example

```
PROGRAM 1                      PROGRAM 2
 1: SEMAPHORE[1] = OFF          1: J P[3] 100% FINE
 2: RUN PRG2                    2: J P[4] 100% FINE
 3: J P[1] 100% FINE            3: J P[5] 100% FINE
 4: J P[2] 100% FINE            4: J P[6] 100% FINE
 5: WAIT SEMAPHORE[1]           5: SEMAPHORE[1] = ON

Group Mask[1,*,*,*,*]          Group Mask[*,1,*,*,*]
```

A.4.13 Position register look–ahead execution instruction

Table A–16. Position register look–ahead execution instruction

Position register lock	LOCK PREG	Locks a position register to prevent the register contents from being changed.
Position register unlock	UNLOCK PREG	Unlocks a position register.

Example

```
 1: J P[1] 100% FINE
 2: PR[1] = PR[2]
 3: PR[2] = PR[3]
 4: LOCK PREG
 5: L P[2] 100mm/sec CNT100
 6: L P[3] 100mm/sec CNT100
 7: L PR[1] 100mm/sec CNT100
 8: L P[4] 100mm/sec CNT100 OFFSET PR[2]
 9: L P[5] 100mm/sec FINE
10: UNLOCK PREG
```

A.4.14 Soft float instruction

Table A–17. Soft float instruction

Soft float start	SOFTFLOAT[i]	Enables the soft float function.
Soft float end	SOFTFLOAT END	Disables the soft float function.
Follow–up	FOLLOW UP	Assumes the current robot position to be the taught position (follow–up) when the soft float function is used.

Example

```
1: J P[1] 100% FINE
2: SOFTFLOAT[1]
3: L P[2] 100mm/sec FINE
4: FOLLOWUP
5: L P[3] 100mm/sec FINE
6: SOFTFLOAT END
```

A.4.15 Status monitoring instructions

Table A–18. Status Monitoring Instructions

Status monitoring start instruction	MONITOR<Condition program name>	Starts monitoring under the conditions specified in the condition program.
Status monitoring end instruction	MONITOR END <Condition program name>	Ends monitoring under the conditions specified in the condition program.

Example

```
 1: MONITOR WRKFALL
 2: J P[1] 100% FINE
         .
         .
         .
 8: J P[7] 100% FINE
 9: MONITOR END WRKFALL
10: OPEN HAND
```

A.4.16 Motion group instructions

Table A–19. Motion Group Instructions

Independent motion group	Independent GP	Enables motion groups to operate independently of each other.
Simultaneous motion group	Simultaneous GP	Enables motion groups to operate simultaneously with the motion group that requires the longest travel time.

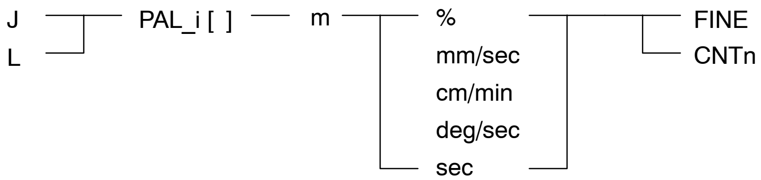
Example

```
1: Independent GP
 : GP1 L P[1] 90mm/sec CNT100
 : GP2 J P[1] 100% CNT50
2: Simultaneous GP
 : GP1 L P[2] 150mm/sec CNT20
 : CP2 J P[2] 70% CNT50
```

A.4.17 Palletizing instructions

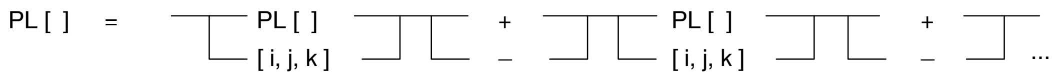
Table A–20. Frame setup instruction

Palletizing	PALLETIZING–B_i i: Palletizing number	Calculates palletizing.
Palletizing end	PALLETIZING–END_i i: 1 to 16	Increases or decreases the contents of a palletizing register.
Palletizing motion	L PAL_i[A–j] 300mm/s FINE	Specifies the palletizing position. i: Palletizing number (1 – 8). j: Number of the approach point.



Example

```
1: PALLETIZING-B_3
2: J  PAL_3[A_2] 50% CNT50
3: L  PAL_3[A_1] 100mm/sec CNT10
4: L  PAL_3[BTM] 50mm/sec FINE
5:    HAND1 OPEN
6: L  PAL_3[R_1] 100mm/sec CNT10
7: J  PAL_3[R_2] 50% CNT50
8:    PALLETIZING-END_3
```



Example

```
1: PL[1] = PL[3] + [1,2,1]
2: PL[2] = [1,2,1] + [1,R[1],*]
```

B. APPENDIX

This appendix summaries items necessary for using this model. It may also be used as an index.

☐ Contents of this appendix

B.1 Start Mode

B.1.1 Start up methods

R–J3 controller has the following four start up methods(start mode):

Initial start

When the unit is started in the initial start mode, all programs are deleted, and all settings are reset to their standard values. Upon the completion of the initial start, a controlled start is performed automatically.

Controlled start

When the unit is started in the controlled start mode, a controlled start menu, which is a simple system, starts up. The controlled start menu cannot be used to operate the robot. The controlled start menu can, however, be used to change a system variable which normally cannot be changed, to read a system file, and to set the robot. From the menu displayed by pressing the Fctn key on the controlled start menu, a cold start can be made.

Cold start

The cold start mode is used to perform normal power–up while power restoration is disabled. The program is aborted, and all output signals are turned off. Once the cold start has been completed, the robot can be operated.

A cold start can be performed while power restoration is enabled, provided the necessary setting is made at power–up.

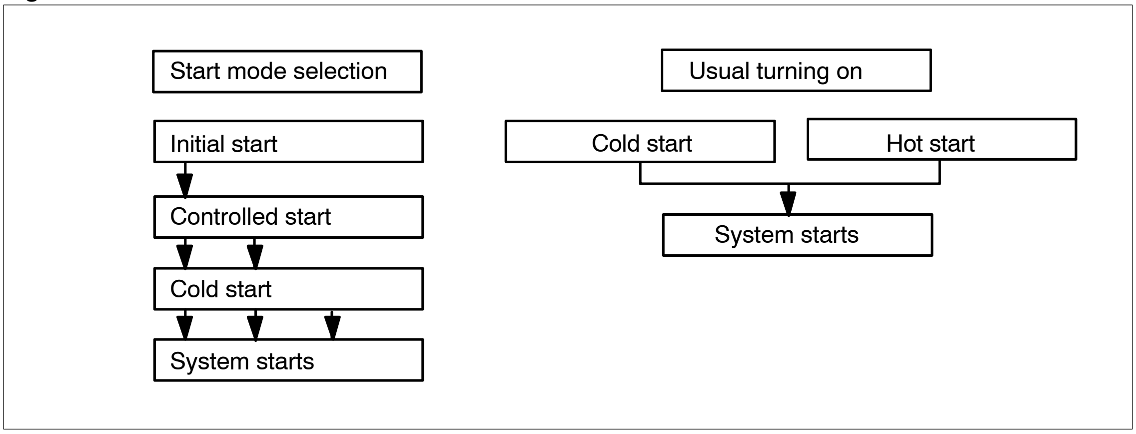
Hot start

The hot start mode is used to perform normal power–up while power restoration is enabled. When the unit starts, the program runs and output signals are restored to the state existing prior to the last power–down. Once the hot start has been completed, the robot can be operated.

The cold start or the hot start is started in usually operation. Which mode is used depends on whether the hot start is enable or disable.

The initial start and the controlled start will be used when maintaining it. These modes will not be used in usually operation.

Figure B–1. Start mode



B.1.2 Initial start

When the unit is started in the initial start mode, all programs are deleted, and all the settings are reset to their standard values.

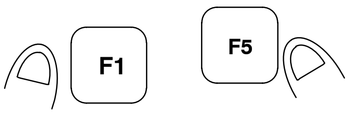
Once the initial start has been completed, a controlled start is performed automatically.

> **NOTE**
> At an initial start, programs and all data including settings will be lost. The factory–set mastering data is also erased. The initial start should be made only when the main printed circuit board or software is replaced. Before performing an initial start, therefore, make a backup copy of the necessary programs and system files.

Procedure B–1 Initial start

Step

1 Press and hold the F1 key and F5 key and then turn on the power on the operator panel, the boot monitor screen is displayed.




```
*** BOOT MONITOR for R-J3i CONTROLLER ***

Base system version (FRL)
Initializing file device ...done
*******  BMON MENU  *******
1. Configuration menu
2. All software installation
3. Init start
4. Controller backup/restore
5. Hardware diagnosis

Selece :
```

2 Select 3, Init start.

3 Enter 1, (YES) to the confirmation message of initial start.

```
CAUTION: INIT start is selected

Are you SURE? [Y=1 / N=else]
```

An initial start is performed. Upon the completion of the initial start, a controlled start is performed automatically, and the controlled start menu appears.

B.1.3 Controlled start

When the unit is started in the controlled start mode, a controlled start menu, which is a simple system, starts up. The controlled start menu cannot be used to operate the robot. The controlled start menu can, however, be used to change a system variable which normally cannot be changed, to read a system file, and to set the robot.

Press the Fctn key on the controlled start menu. A menu appears. From that menu, select 1 START (COLD). A cold start is performed.

The following screens can be displayed from the menu displayed by pressing the MENU key on the controlled start menu:

Setting screens
Settings can be made.

Software install screen
Optional software can be added or deleted.

System variables screen
System variables can be set. Even a system variable which cannot normally be changed (R0) can be changed.

On the file screen of the controlled start menu, F4 is displayed as [RESTORE]. When the F4 key is pressed, all files are read automatically. To switch F4 to [BACKUP] as on other file screens, press the Fctn key. A menu appears. From that menu, select RESTORE/BACKUP.

File screen
A program or system file can be saved and read. The system file can be read only from the controlled start menu.

Version ID Screen
The software edition is displayed.

Alarm history screen
The alarm history is displayed.

Port in:t screen
A serial port is set.

This screen is used to read a file from a Handy File or the like upon a controlled start.

Memory screen
The memory status is displayed.

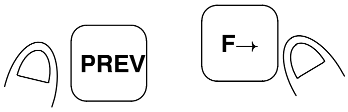
MAINTENANCE
A robot setting can be changed. An additional axis can be set. A motion group can be added or deleted.

Max. number setting screen
The numbers of resistor, macro, and user alarm can be changed.

Procedure B–2 Controlled start

Step

1 Press and hold the PREV key and F→ key and then turn on the power on the operator panel. The configuration manu is displayed.




```
System version:V6.1001      08/28/98

----------CONFIGURATION MANU----------

1. Hot start
2. Cold start
3. Controlled start
4. Maintenance



Select>
```

2 Select 3 CONTROLLED START. The setting screen for the controlled start menu appears.

3

ENTER

```
Tool Setup          CONTROLLED START MENUS
                                       1/1

FANUC Spot Tool


1 F Number                          F00000
[TYPE]
```

3 To operate the robot, a cold start must be performed. To do this, press the Fctn key. A menu appears. From that menu, select 1 START (COLD). A cold start is performed.

Fctn

```
1 START (COLD)
```

ENTER

B.1.4 Cold start

The cold start mode is used when normal power–up is performed while power restoration is disabled.

Upon a cold start, the following is performed:

- Each output signal of digital I/O, analog I/O, robot I/O, and group I/O is turned off or set to 0.
- The program is aborted, and the beginning of the program becomes the current line.
- The feedrate override is reset to the initial value.
- The manual feed coordinate system enters the JOINT state.
- The machine lock is released.

The cold start procedure depends on the power restoration setting.

Procedure B–3 Cold start

Condition ■ Hot start must be set to invalid.

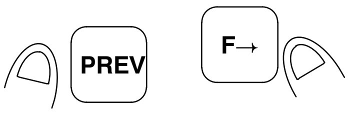
Step 1 Turn on the power. The following screen is displayed after the system stars by cold start.

```
UTILITIES Hints                  JOINT  30 %

FUNUC Handling Tool
V6,10P01

Copyright 1998
FANUC LTD
FANUC Robotics North America,Inc
All right Reserved

[ TYPE ]                              HELP
```

Procedure B–4 Cold start

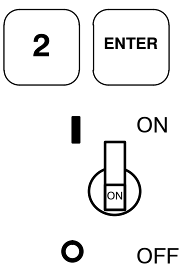
Condition ■ Hot start is set to disable

Step 1 Press and hold the PREV key and F→ key and then turn on the power on the operator panel. The configuration menu is displayed.




```
System version:V6.1001      08/28/98

----------CONFIGURATION MANU----------

1. Hot start
2. Cold start
3. Controlled start
4. Maintenance



Select>
```

2 Select “2. Cold start”. Cold start is executed and following is displayed.




```
UTILITIES Hints                  JOINT  30 %

FUNUC Handling Tool
V6,10P01

Copyright 1998
FANUC LTD
FANUC Robotics North America, Inc
All right Reserved

[ TYPE ]                              HELP
```

B.1.5 Hot start

The hot start mode is used when normal power–up is performed while power restoration is enabled. Upon a hot start, the following is performed:

– Each output signal of digital I/O, analog I/O, robot I/O, and group I/O is set in the same manner as it was prior to the last power–down.

– The program runs in the same way as it did prior to the last power–down. If the program was running up until the last power–down, the program enters the pause state.

– The feedrate override, manual feed coordinate system, and machine lock are set in the same manner as they were prior to the last power–down.

NOTE
When a hot start is performed in the following state, each output signal of the digital I/O, analog I/O, robot I/O, and group I/O is turned off or set to 0:
– When the I/O allocation is changed
– When an I/O unit is mounted or removed
– When the number of signals is changed on the I/O Link screen

Procedure B–5 Hot start

Condition ■ Hot start must be set to enable.

Step 1 Turn on the power. The screen which was being displayed at power off will be displayed on the screen of the teach pendant after a few seconds.

B.2 Mastering

Mastering associates the angle of each axis of the robot with the pulse count of the absolute pulse coder (APC) connected to the motor of each axis. More specifically, mastering is performed by obtaining the pulse count read at the zero–degree position.

The current position of the robot is determined by the pulse counts of the absolute pulse coders (APCs) for the axes.

Since mastering data is factory–set, mastering is unnecessary in normal operation. If one of the following events occurs, however, mastering must be performed:

- Mastering data is lost for some reason such as a drop in the voltage of the backup battery for C–MOS in the controller or memory erasing with an initial start.
- The APC pulse counts are lost for some reason such as a drop in the voltage of the backup battery for APC pulse counts backup in the mechanical unit or exchange of pulse coder.
- The pulse counts do not indicate the angles of the axes because the mechanical unit was hit bumped, etc.

CAUTION

The robot data including mastering data and the pulse coder data are maintained independently by backup batteries. If the batteries go empty, data is lost. To prevent this, replace both batteries periodically. When the battery voltage drops, an alarm 'BLAL' notifies the user.

There are five types of mastering as listed below.

Table B–1. Mastering types

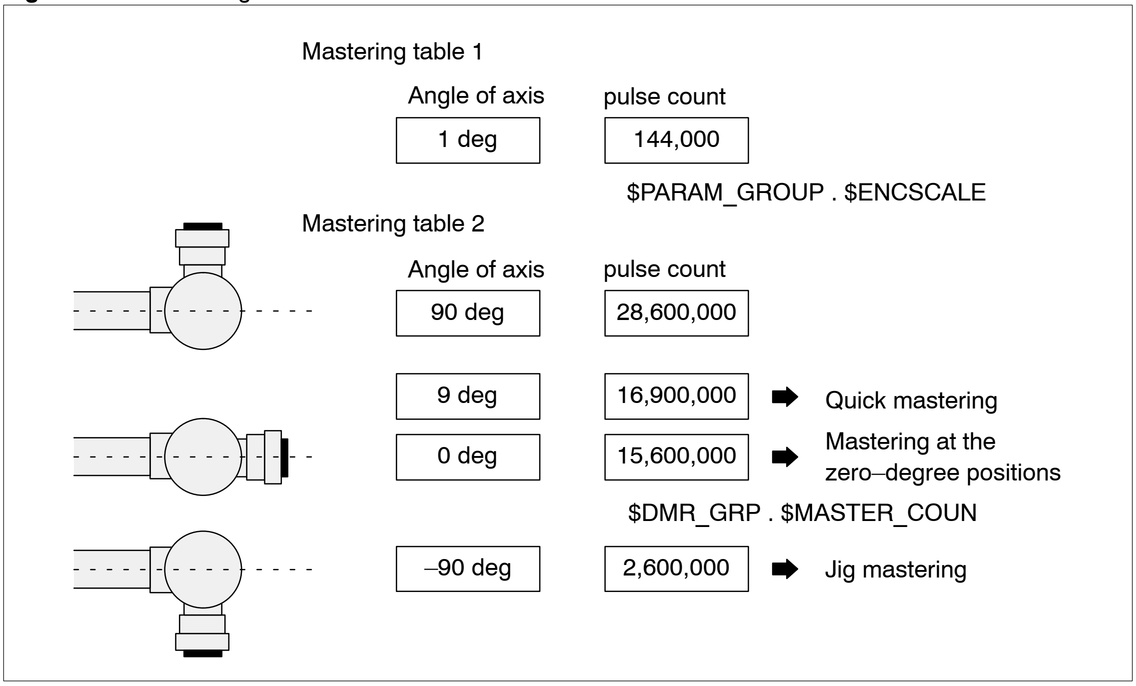
Type of mastering	Explanation
Jig mastering	Mastering is performed using a special jig. Jig mastering is performed at the factory.
Mastering at the zero–degree positions	Mastering is performed with each axis of the robot aligned with the zero–degree position. The zero–degree position mark attached to each axis of the robot is referenced.
Quick mastering	The mastering position can be set at any position. To do this, reference points must be set in advance.
Single axis mastering	Single axis mastering is that the mastering is performed every one axis.
Setting mastering data	Mastering data is set in mastering counters directly.

CAUTION

After the robot is installed, the quick mastering reference points should be stored in case the factory–adjusted settings are needed in mastering in the future.

After mastering, be sure to perform positioning (calibration). Positioning means that the controller reads the current pulse counts and recognizes the current position.

Figure B–2. Mastering



The current position of the robot is determined by the following data:

- Pulse count per degree (See mastering table 1.) This value is defined in system variable $PARAM_GROUP.$ENCSCALE.
- Pulse count at the zero–degree position (See mastering table 2.) This data is stored in $DMR_GRP.$MASTER_COUN by mastering.
  - In jig mastering, the pulse count at the jig position is received and converted to mastering data.
  - In quick mastering, the pulse count at the quick mastering reference position defined by the user is received and converted to mastering data.
- Current pulse count. The current pulse count is received from the by calibration.

Mastering and calibration are performed on the Master/Cal screen [6 SYSTEM, Master/Cal].

NOTE Mastering by mistake may cause the robot to move unexpectedly and it is very dangerous. Therefore, the Master/Cal screen will be displayed only when the system variable, $MASTER_ENB, is set to 1 or 2. Press F5,DONE,which is displayed in the Master/Cal screen after mastering. $MASTER_ENG is automatically set to 0 and then the Master/Cal screen can not be displayed. If you want to display the Master/Cal screen again, set $MASTER_ENB to 1 in the system variable screen again.

B.2.1 Jig mastering

Jig mastering is performed at the factory using a special jig. This mastering is performed at the mastering position set beforehand.

With this mastering, the accurate mastering can be performed by using the special jig.

This mastering is usually unnecessary to perform it in normal operation because this is used at shipment.

For details of jig mastering, refer to the maintenance manual.

Procedure B–6 Jig mastering

Condition ■ System variable $MASTER_ENB must be set to 1 or 2.

```
SYSTEM Variables                  JOINT 10%
                                    57/136
 57  $MASTER_ENB                     1
```

Step

1 Press the MENUS key. The screen menu is displayed.

2 Select ”0 — NEXT —” and then select ”6 SYSTEM”.

3 Press F1, [TYPE]. The screen change menu is displayed.

4 Select “Master/Cal” on the screen change menu. The positioning screen appears.

```
 9 USER
 0 -- NEXT --
```

MENUS

```
 5 POSITION
 6 SYSTEM
 7
```

```
  Master/Cal
   TYPE
```

F1

```
SYSTEM Master/Cal                 JOINT 30%

  1  FIXTURE POSITION MASTER
  2  ZERO POSITION MASTER
  3  QUICK MASTER
  4  SINGLE AXIS MASTER
  5  SET QUICK MASTER REF
  6  CALIBRATE


  Press 'ENTER' or number key to select.


  [TYPE]   LOAD RES_PCA           DONE
```

5 Move the robot by jog feed to the mastering position. Release the brake on the manual brake control screen if necessary.

6 Select “1 FIXTURE POSITION MASTER” and press the F4 key (yes). Mastering data is set.

```
 SYSTEM Master/Cal

   1 FIXTURE POSITION MASTER
   2 ZERO POSITION MASTER
   Master at master position?  [NO]

Master at master position?  [NO]
[ TYPE ]                    YES      NO
```

ENTER

F4

```
 SYSTEM Master/Cal                JOINT  30 %

   1 FIXTURE POSITION MASTER
   2 ZERO POSITION MASTER
   3 QUICK MASTER
   4 SINGLE AXIS MASTER
   5 SET QUICK MASTER REF
   6 CALIBRATE
 Robot Mastered! Mastering Data:
    <0> <11808249> <38767856>
    <9873638> <122000309> <2000319>


 [ TYPE ]  LOAD  RES_PCA           DONE
```

7 Select “6 CALIBRATE” and press the F4 key (yes). Calibration is performed.

```
 5 SET QUICK MASTER REF
 6 CALIBRATE
 Calibrate? [NO]
```

ENTER

```
Calibrate? [NO]
[ TYPE ]                 YES      NO
```

F4

```
SYSTEM Master/Cal            JOINT  30 %

  1 FIXTURE POSITION MASTER
  2 ZERO POSITION MASTER
  3 QUICK MASTER
  4 SINGLE AXIS MASTER
  5 SET QUICK MASTER REF
  6 CALIBRATE
Robot Calibrated! Cur Jnt Ang(deg):
   <10.000> <-25.000> <40.000>
   <5.000> <-15.000> <0.000>
[ TYPE ]  LOAD  RES_PCA          DONE
```

8 Press F5 “DONE”, after mastering.

```
                DONE
```

F5

9 Alternatively, to perform positioning, turn the power off, then turn it on again. Calibration is performed whenever the power is turned on.

B.2.2 Mastering at the zero–degree positions

Mastering at the zero–degree positions is performed for the robot with its all axes at the zero–degree positions. On each axis of the robot, a zero–degree position mark is attached. Using these marks as a reference, move the robot by jog feed to the zero–degree positions for all axes.

Mastering at the zero–degree positions cannot be performed as accurate by as other types of mastering because it relies on visual adjustment. Perform mastering at the zero–degree positions only as an emergency measure.

For details of mastering at the zero–degree positions, refer to the maintenance manual.

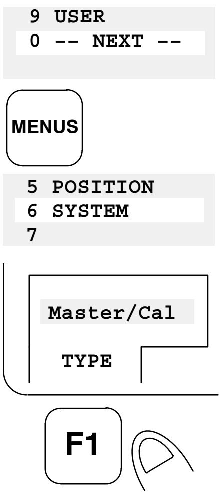
Procedure B–7 Mastering at the zero–degree positions

Condition

- System variable $MASTER_ENB must be set to 1 or 2.

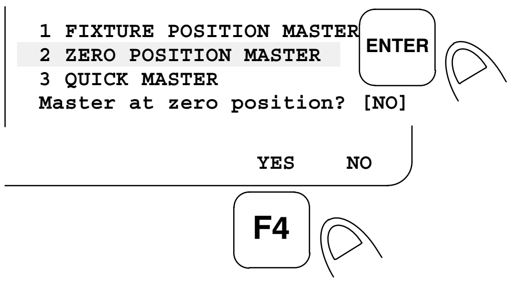
Step

1 Press the MENUS key. The screen menu is displayed.
2 Select "0 — NEXT —" and then select "6 SYSTEM".
3 Press F1, [TYPE]. The screen change menu is displayed.
4 Select "Master/Cal" on the screen change menu. The Master/Cal screen appears.



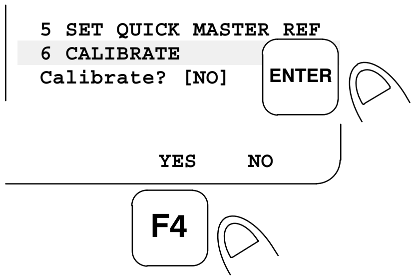

```
SYSTEM Master/Cal                 JOINT 30%

 1  FIXTURE POSITION MASTER
 2  ZERO POSITION MASTER
 3  QUICK MASTER
 4  SINGLE AXIS MASTER
 5  SET QUICK MASTER REF
 6  CALIBRATE

 Press 'ENTER' or number key to select.

 [TYPE]   LOAD RES_PCA           DONE
```

5 Move the robot by jog feed to the zero–degree positions for all axes. Set brake control to off, if necessary.
6 Select "2 ZERO POSITION MASTER" and press the F4 key (yes). Mastering data is set.




```
SYSTEM Master/Cal              JOINT  30 %

  1 FIXTURE POSITION MASTER
  2 ZERO POSITION MASTER
  3 QUICK MASTER
  4 SINGLE AXIS MASTER
  5 SET QUICK MASTER REF
  6 CALIBRATE
Robot Mastered! Mastering Data:
   <0> <11808249> <38767856>
   <9873638> <122000309> <2000319>

[ TYPE ]  LOAD  RES_PCA           DONE
```

7 Select “6 CALIBRATE” and press the F4 key (yes). Calibration is performed.




```
SYSTEM Master/Cal            JOINT  30 %

 1 FIXTURE POSITION MASTER
 2 ZERO POSITION MASTER
 3 QUICK MASTER
 4 SINGLE AXIS MASTER
 5 SET QUICK MASTER REF
 6 CALIBRATE
Robot Calibrated! Cur Jnt Ang(deg):
   <0.000> <0.000> <0.000>
   <0.000> <0.000> <0.000>

[ TYPE ]  LOAD  RES_PCA          DONE
```

8 Press F5 “DONE”, after mastering.

9 Alternatively, to perform calibration, turn the power off, then turn it on again. Calibration is performed whenever the power is turned on.

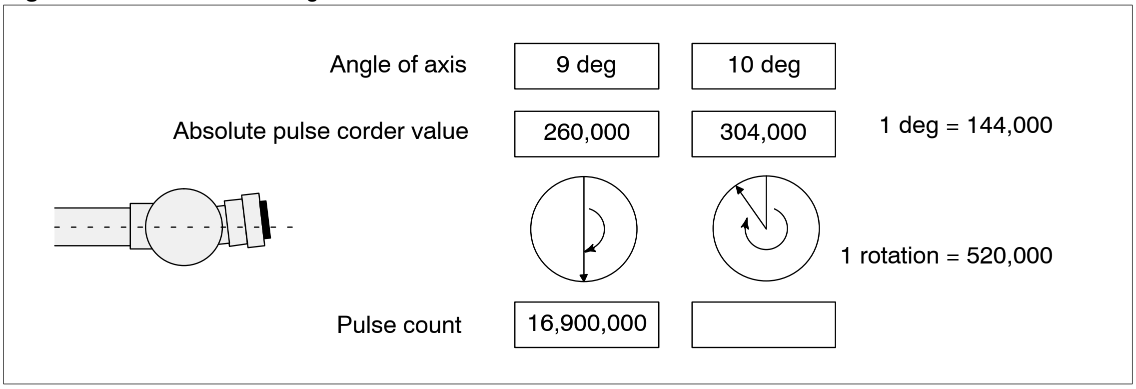
B.2.3 Quick mastering

Quick mastering allows mastering at any user–defined position. The pulse counts are calculated from the speed and angular displacement within one rotation of the APCs connected to the motors. Quick mastering uses the fact that the absolute angular displacement within one rotation is not lost.

- If mastering data is lost due to the empty of the backup battery for the pulse coder, quick mastering can be used.
- When the pulse coder is replaced or when mastering data in the robot controller is lost, quick mastering cannot be used.

To perform simple mastering, a reference point set after mastering is necessary (→ reference point setting). The reference point is factory–set to the zero position.

Figure B–3. Quick Mastering



Quick mastering uses the fact that the deviation of the angle of the axis from the reference point can accurately be compensated when it is within one rotation of the APC.

For details of quick mastering, refer to the maintenance manual.

CAUTION

If the robot is installed in such a way that the robot cannot be set to the 0° position, which is the reference point of initial simple mastering, the reference point of simple mastering should be stored after the installation. This must be done to store the factory–set mastering setting, providing for future mastering.

Procedure B–8 Quick mastering

Condition
- System variable $MASTER_ENB must be set to 1 or 2.
- Quick mastering reference position (reference position) must be set.

Step

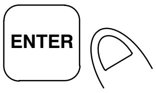
1 Display the Master/Cal screen.

```
SYSTEM Master/Cal                  JOINT 30%

 1  FIXTURE POSITION MASTER
 2  ZERO POSITION MASTER
 3  QUICK MASTER
 4  SINGLE AXIS MASTER
 5  SET QUICK MASTER REF
 6  CALIBRATE

 Press 'ENTER' or number key to select.

 [TYPE]   LOAD RES_PCA            DONE
```

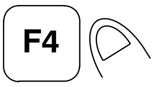
2 Jog the robot to the quick mastering position(reference position). If it is necessary,turn off the brake control.

3 Select “3 QUICK MASTER” and press the F4 key (yes). Mastering data is set.

```
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
```



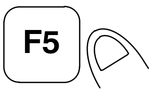

```
           YES    NO
```



4 Select “6 CALIBRATE” and press the F4 key (yes). Calibration is performed.

5 Press F5 “DONE” after mastering.

```
              DONE
```



Procedure B–9 Setting reference points for quick mastering (If the robot is installed in such a way that the robot cannot be set to the 0° position)

> **CAUTION**
> This operation cannot be executed if the mastering data is lost because of mechanical disassembly or maintenance. If that is the case, jig mastering or zero–degree positions mastering should be executed to restore the mastering data.

Condition ■ System variable $MASTER_ENB must be set to 1 or 2.

Step

1 Select “6 SYSTEM” on the screen menu.

2 Select “Master/Cal” on the screen change menu. The Master/Cal screen appears.

```
 5 POSITION
 6 SYSTEM
 7
```

MENUS

```
 Master/Cal

   TYPE
```

F1

```
SYSTEM Master/Cal                JOINT 30%

 1  FIXTURE POSITION MASTER
 2  ZERO POSITION MASTER
 3  QUICK MASTER
 4  SINGLE AXIS MASTER
 5  SET QUICK MASTER REF
 6  CALIBRATE

 Press 'ENTER' or number key to select.

 [TYPE]   LOAD RES_PCA           DONE
```

3 Move the robot by jog feed to the quick mastering reference position. Set brake control to off, if necessary.

4 Select “5 SET QUICK MASTER REF” and press the F4 key (yes). The reference points for quick mastering are stored in memory.

```
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE


            YES    NO
```

ENTER

F4

B.2.4 Single axis mastering

User can select the arbitrary position for the mastering of each axis.

Single axis mastering should be used when the mastering data of some axes is lost for some reason such as the drops of the voltage of the backup battery for pulse coder or exchanging of the pulse coder.

```
 SINGLE AXIS MASTER          JOINT  30 %
                                   1/9
    ACTUAL POS    (MSTR POS )  (SEL)[ST]
  J1     25.255   (     0.000)  (0)  [2]
  J2     25.550   (     0.000)  (0)  [2]
  J3    -50.000   (     0.000)  (0)  [2]
  J4     12.500   (     0.000)  (0)  [2]
  J5     31.250   (     0.000)  (0)  [2]
  J6      0.000   (     0.000)  (0)  [2]
  E1      0.000   (     0.000)  (0)  [2]
  E2      0.000   (     0.000)  (0)  [2]
  E3      0.000   (     0.000)  (0)  [2]

                           GROUP    EXEC
```

Table B–2 Settings for single axis mastering

ITEMS	DESCRIPTIONS
ACTUAL POS	The current position expressed by joint (degree) of the robot is displayed.
MSTR POS	Specifies the mastering position to the axis to be performed the single axis mastering. It is usually specified 0 degree.
SEL	For the axis to be performed mastering,set this item to 1. It is usually 0.
ST	Display the state of completion of the single axis mastering. The value displayed at this item can not be directly changed. The values of $EACHMST_DON[1 to 9] are displayed at this column. – 0 Specifies that the mastering data has been lost. The single axis mastering need to be performed. – 1 The mastering data has been lost.(Only other interactive axes is performed mastering.) This axis need to be performed mastering. – 2 The mastering has been completed.

Refer to the maintenance manual for an accurate method of single axis mastering.

Procedure B–10 Single axis mastering

Condition ■ System variable $MASTER_ENBL must be set to 1.

Step

1 Select "6 SYSTEM" on the screen menu.

2 Select "Master/Cal" on the screen change menu. The Master/Cal screen appears.

```
 5 POSITION
 6 SYSTEM
 7
```

MENUS

```
 Master/Cal

  TYPE
```

F1

```
SYSTEM Master/Cal                JOINT 30%

 1  FIXTURE POSITION MASTER
 2  ZERO POSITION MASTER
 3  QUICK MASTER
 4  SINGLE AXIS MASTER
 5  SET QUICK MASTER REF
 6  CALIBRATE

 Press 'ENTER' or number key to select.

 [TYPE]   LOAD RES_PCA           DONE
```

3 Select "4 SINGLE AXIS MASTER". The single axis mastering screen is displayed.

In the example on the mastering of J2 and the J3 axis needs to be executed.

```
SINGLE AXIS MASTER              JOINT  30 %
                                        1/9
   ACTUAL POS    (MSTR POS )   (SEL) [ST]
 J1     25.255   (     0.000)   (0)   [2]
 J2     25.550   (     0.000)   (0)   [0]
 J3    -50.000   (     0.000)   (0)   [0]
 J4     12.500   (     0.000)   (0)   [2]
 J5     31.250   (     0.000)   (0)   [2]
 J6      0.000   (     0.000)   (0)   [2]
 E1      0.000   (     0.000)   (0)   [2]
 E2      0.000   (     0.000)   (0)   [2]
 E3      0.000   (     0.000)   (0)   [2]
                            GROUP     EXEC
```

4 Enter 1 to SEL setting field of the axis that you want to perform the mastering. SEL can be specified every one axis or plural axes simultaneously.

```
R            JOINT   30 %
                      2/9
(     0.000)  (0)   [0]
(     0.000)  (0)   [0]
```

```
SINGLE AXIS MASTER              JOINT  30 %
                                        3/9
 J2     25.550   (     0.000)   (1)   [0]
 J3    -50.000   (     0.000)   (1)   [0]

                            GROUP     EXEC
```

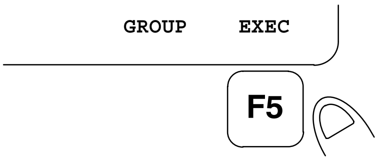
5 Jog the robot to the mastering position. Turn off the brake control if it is necessary.

6 Enter the axis data of the mastering position.

```
R            JOINT   30 %
                      2/9
(     0.000)  (1)   [0]
(     0.000)  (1)   [0]
```

```
SINGLE AXIS MASTER              JOINT  30 %
                                        3/9
 J2     25.550   (     0.000)   (1)   [0]
 J3    -50.000   (    90.000)   (1)   [0]

                            GROUP     EXEC
```

7 Press F5 ”EXEC.” The mastering is performed. This operation causes that SEL is set to 0 and ST is set to 2 or 1.




```
SINGLE AXIS MASTER              JOINT  30 %
                                        1/9
   ACTUAL POS    (MSTR POS )   (SEL) [ST]
 J1    25.255    (     0.000)   (0)   [2]
 J2    25.550    (     0.000)   (0)   [2]
 J3   -50.000    (    90.000)   (0)   [2]
 J4    12.500    (     0.000)   (0)   [2]
 J5     0.000    (     0.000)   (0)   [2]
 J6     0.000    (     0.000)   (0)   [2]
 E1     0.000    (     0.000)   (0)   [2]
 E2     0.000    (     0.000)   (0)   [2]
 E3     0.000    (     0.000)   (0)   [2]
                             GROUP    EXEC
```

8 When the single axis mastering is completed, press the PREV key to display the Master/Cal screen.


```
SYSTEM Master/Cal               JOINT 30%

 1  FIXTURE POSITION MASTER
 2  ZERO POSITION MASTER
 3  QUICK MASTER
 4  SINGLE AXIS MASTER
 5  SET QUICK MASTER REF
 6  CALIBRATE
 Press 'ENTER' or number key to select.
 [TYPE]   LOAD RES_PCA           DONE
```

9 Select ”6 CALIBRATE” and press F4 ”YES.” The calibration is performed.

10 Press F5 ”DONE”, after calibration.

B.2.5 Setting mastering data

Mastering data can directly be set to the system variable. Setting mastering data can be performed when the pulse counts are not changed.

- If C–MOS mastering data is lost for some reason such as an initial start, set the recorded mastering data.
- Setting mastering data cannot be performed when pulse count data is lost.

Procedure B–11 Directly setting mastering data

Step

1 Select “6 SYSTEM” on the screen menu.

2 Select “Variables” on the screen change menu. The system variable screen appears.

```
 5 POSITION
 6 SYSTEM
 7
```

MENUS

```
 Variables

  TYPE
```

F1

```
 SYSTEM Variables                     JOINT 10%
                                          1/98
   1 $AP_MAXAX             536870912
   2 $AP_PLUGGED           4
   3 $AP_TOTALAX           16777216
   4 $AP_USENUM            [12] Of Byte
   5 $AUTOINIT             2
   6 $BLT                  19920216

 [TYPE]
```

3 Change mastering data.
Mastering data is stored in system variable $DMR_GRP.$MASTER_COUN.

```
 13 $DMR_GRP             DMR_GRP_T
 14 $ENC_STAT            [2]of ENC_STAT_T


 [TYPE]
```

4 Select “$DMR_GRP.”

```
 DMR_GRP_T
 [2] of ENC_STAT_T
```

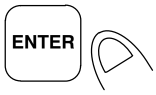
ENTER

```
 SYSTEM Variables                     JOINT 10%
 $DMR_GRP                                  1/1
  1  [1]                        DMR_GRP_T
```

```
 SYSTEM Variables                  JOINT  30 %
 $DMR_GRP                                  1/8
  1  $MASTER_DONE        FALSE
  2  $OT_MINUS           [9]of Boolean
  3  $OT_PLUS            [9]of Boolean
  4  $MASTER_COUN        [9]of Integer
  5  $REF_DONE           FALSE
  6  $REF_POS            [9]of Real
  7  $REF_COUNT          [9]of Integer
  8  $BCKLSH_SIGN        [9]of Boolean
 [ TYPE ]                   TRUE   FALSE
```

5 Select “$MASTER_COUN” and enter mastering data.

```
[9] of Boolean
[9] of integer
FALSE
```

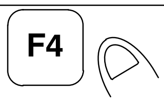



```
SYSTEM Variables                  JOINT 10%
$DMR_GRP[1].$MASTER_COUN               1/9
 1   [1]                        95678329
 2   [2]                        10223045
 3   [3]                        3020442
 4   [4]                        304055030
 5   [5]                        20497709
 6   [6]                        0

[TYPE]
```

6 Press the PREV key.

7 Set “$MASTER_DONE” to “TRUE.”

```
TRUE    FALSE
```

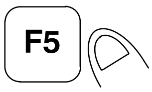



```
SYSTEM Valiables                  JOINT 10%
$DMR_GRP[1]                            1/8
 1   $MASTER_DONE          TRUE
 2   $OT_MINUS             [9]of Boolean
```

8 Display the Master/Cal screen and select “6 CALIBRATE.”

9 Press F5 ”DONE”, after calibration.

```
DONE
```



B.3 Software Version

Screens related to the software version display identification information of the controller. This information is to be reported to FANUC if a failure occurs in the controller. The following are the screens related to the software version:

```
[TYPE] SOFTWARE  MOT_ID  MOT_INF  SER_PAR
```

- F2 “SOFTWARE” : Displays the software version screen.
- F3 “MOT_ID”: Displays the motor ID screen.
- F4 “MOT_INF”: Displays the motor information screen.
- F5 “SER_PAR”: Displays the servo parameter information screen.

Software version screen

The software version screen displays the following information:

```
STATUS Version ID             JOINT  10 %
     SOFTWARE:               ID:    1/11
  1  FANUC LR Tool           M6.1011
  2  S/W Serial No.          007051
  3  Controller ID           F00000
  4  Default Personality (from FD)
  5  LRM100iB                M6.1011
  6  Servo Code              V10.09S
  7  Cart. Mot. Parameter    V3.00
  8  Joint Mot. Parameter    V3.00
  9  Software Edition No.    M6.1011
 10  Boot MONITOR            M6.1011

[ TYPE ] SOFTWARE CONFIG  MOTOR   SERVO
```

Software configuration

The software configuration screen displays the software installed.

```
STATUS Version ID             JOINT  10 %
     FEATURE:              ORD NO:  1/128
  1  FANUC LR Tool           H548
  2  English Dictionary      H521
  3  Multi Language (KANA)   H530
  4  Ext. DIO Config         EIOC
  5  External DI BWD         ESET
  6  SSPC error text         ETSS
  7  Kernel + Basic S/W      H510
  8  Robot Servo Code        H930
  9  MACROs, Skip/Offset     J503
 10  Program Shift           J505

[ TYPE ]SOFTWARE CONFIG  MOTOR   SERVO
```

Motor information screen

The motor information screen displays servo control information on each axis.

```
STATUS Version ID             JOINT  10 %
   GR: AX: MOTOR ID AND INFO:      1/16
 1  1    1  ACbM0.5/4000 20A  H1 DSP1-L
 2  1    2  ACbM0.5/4000 20A  H2 DSP1-M
 3  1    3  ACbM0.5/4000 10A  H3 DSP2-L
 4  1    4  ACbM0.4/4000 10A  H4 DSP2-M
 5  1    5  ACbM0.4/4000 10A  H5 DSP3-L
 6 **   **  ****************************
 7 **   **  ****************************
 8 **   **  ****************************
 9 **   **  ****************************
10 **   **  ****************************

[ TYPE ]SOFTWARE CONFIG  MOTOR   SERVO
```

Servo parameter information screen

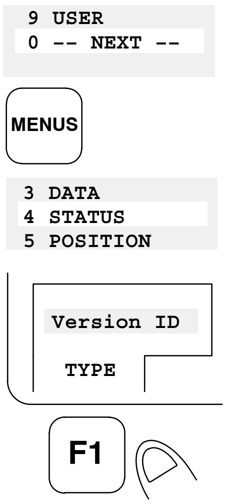
The servo parameter information screen displays the ID of the servo parameter for each axis.

```
STATUS Version ID           JOINT  10 %
    GROUP: AXIS:  SERVO PARAM ID:  1/16
 1   1      1     P00.11
 2   1      2     P00.11
 3   1      3     P00.11
 4   1      4     P00.11
 5   1      5     P00.11
 6  **     **     ********
 7  **     **     ********
 8  **     **     ********
 9  **     **     ********
10  **     **     ********

[ TYPE ]SOFTWARE CONFIG  MOTOR   SERVO
```

Procedure B–12 Software version screen

Step

1 Press MENUS key to display the screen menu.
2 Select ”0 — NEXT —” and then select ”4 STATUS” on the next page.
3 Press F1 ”TYPE” to display the screen change menu.
4 Select ”Version ID” . Software version screen is displayed.




```
STATUS Version ID           JOINT  10 %
     SOFTWARE:              ID:    1/11
  1  FANUC LR Tool          M6.1011
  2  S/W Serial No.         007051
  3  Controller ID          F00000
  4  Default Personality (from FD)
  5  LRM100iB               M6.1011
  6  Servo Code             V10.09S
  7  Cart. Mot. Parameter   V3.00
  8  Joint Mot. Parameter   V3.00
  9  Software Edition No.   M6.1011
 10  Boot MONITOR           M6.1011

[ TYPE ]SOFTWARE CONFIG  MOTOR   SERVO
```

– F2 “SOFTWARE” : Displays the software version screen.
– F3 “MOT_ID”: Displays the motor ID screen.
– F4 “MOT_INF”: Displays the motor information screen.
– F5 “SER_PAR”: Displays the servo parameter information screen.

B.4 Robot Axis Status

The robot axis status screens displays the status of each axis motor of the robot. The status of each axis is updated in real time. This status information is used during maintenance.

Status 1 screen

The status 1 screen displays the alarm status of the servo system. The status information consists of servo alarm status 1 (16 bits) and servo alarm status 2 (16 bits).

```
 STATUS Axis                    JOINT  30 %

                                  GRP[ 1 ]
       Flag Bits 1/2           History
  J1: 0000000000000000 (0000000000000000)
      0000000000000000 (0000000000000000)
  J2: 0000000000000000 (0000000000000000)
      0000000000000000 (0000000000000000)
  J3: 0000000000000000 (0000000000000000)
      0000000000000000 (0000000000000000)

 [ TYPE ]STATUS1 STATUS2  PULSE  [UTIL ]>
```

Flag 1	Servo alarm status 1
Flag 2	Servo alarm status 2

Table B–3. Servo Alarm Status 1 Address: FC80h (L–axis), FCC0h (M–axis)

MSB	B14	B13	B12	B11	B10	B9	B8
OHAL	LVAL	OVC	HCAL	HVAL	DCAL	FBAL	ALDF
B7	**B6**	**B5**	**B4**	**B3**	**B2**	**B1**	**LSB**
MCAL	MOFAL	EROFL	CUER	SSTB	PAWT	SRDY	SCRDY

OHAL	Amplifier overheat alarm
LVAL	It indicates a low voltage alarm.
OVC	It indicates an overcurrent (OVC) alarm.
HCAL	It indicates an high current alarm.
HVAL	It indicates an high voltage alarm.
DCAL	It indicates a regenerative discharge alarm.
FBAL	Disconnection alarm (ALDF indicates whether the disconnection is associated with the hardware or software.)
ALDF	Alarm distinction bit If an amplifier alarm (OHAL, LVAL, HCAL, FSAL, IPMAL, or DCLVAL) is raised while ALSF is set to 1, the alarm is detected by PSM. When both FBAL and ALDF are set to 1, the disconnection alarm is detected by the hardware.
MCAL	Amplifier MCC adhesion alarm
MOFAL	Move command overflow alarm When this bit is set to 1, it indicates that an overflow occurred when the move command was distributed.
EROFL	Error counter overflow alarm for line tracking When this bit is set to 1, it indicates that the error counter has overflowed.
CUER	Current offset error This bit is set to 1 when the current offset value of the A/D converter is higher than permitted.
SSTB	Servo standby signal After POWON, this signal is set to 1, and the system waits for ITP. When SSTB is set to 1, the host outputs ITPCON and generates ITP.
PAWT	Parameter change completion signal When the servo CPU finishes rewriting parameters, only 1ITP is set to 1.

SRDY	Servo ready signal While this flag is held to 1, a move command is accepted.
SCRDY	Servo communication flag The servo CPU sets this flag to 1 once data writing to the shared RAM is completed. After reading the data, the host CPU resets the flag to 0.

OVL	FBAL	ALDF	Alarm
1	0	1	Motor overload alarm (not used for a serial pulse coder)
1	0	0	Amplifier overload alarm
0	1	1	Pulse coder disconnection alarm (not used for a serial pulse coder)

Table B–4. Servo Alarm Status 2 Address: FC81h (L–axis), FCC1h (M–axis)

MSB	B14	B13	B12	B11	B10	B9	B8
SRCMF	CLALM	FSAL	DCLVAL	BRAKE	IPMAL	SFVEL	GUNSET
B7	**B6**	**B5**	**B4**	**B3**	**B2**	**B1**	**LSB**
FSSBDC	SCUCAL	AMUCAL	CHGAL	NOAMP			

SRCMF	Compensation warning flag When part of the position data is missing because of noise or some other reason, data compensation is performed. This data, however, should not be used for mastering or other purposes. To inform the host of this state, the flag is set to 1.
CLALM	It indicates a collision detection alarm. When the servo CPU detects a collision, the flag is set to 1. The host CPU starts alarm handling after a lapse of a predetermined period from when the flag is set to 1.
FSAL	Fan stop alarm
DCLVAL	Low DC Link voltage alarm
BRAKE	Brake alarm of 6–axis amplifier
IPMAL	IPM alarm IPM is an abbreviation for intelligent power module, which is a power component to replace IGBT. The IPM detects overheating and HC by itself.
SFVEL	Soft float start permission signal When the velocity feedback falls below the velocity specified in a parameter, this flag is set to 1 to allow soft float to be started.
GUNSET	Servo gun switch completion signal Once the resetting (initialization) of the pulse coder has been completed after the servo gun is switched, the signal is set to 1 only for 1ITP.
FSSBDC	FSSB disconnection alarm When a disconnection of FSSB is detected, this bit is set to 1. (Hardware detection by FSSBC)
SVUCAL	FSSB communication alarm When two consecutive alarms are detected in data communication between the slave and a servo module, this bit is set to 1. (Detected by the servo software)
AMUCAL	FSSB communication alarm When two consecutive alarms are detected in data communication between the servo module and a slave, this bit is set to 1. (Detected by the slave)
CHGAL	Amplifier charge alarm
NOAMP	No amplifier connection alarm This bit is set to 1 when an amplifier is not connected while the presence of the corresponding axis is specified (B3 of AXIS register set to 0).

Status 2 screen

The status 2 screen indicates the pulse coder alarm status (12 bits).

```
STATUS Axis                    JOINT  30 %

                                 GRP[ 1 ]
      Alarm Status      History
 J1: 0000000000000000 (0000000000000000)
 J2  0000000000000000 (0000000000000000)
 J3: 0000000000000000 (0000000000000000)
 J4  0000000000000000 (0000000000000000)
 J5: 0000000000000000 (0000000000000000)


[ TYPE ]STATUS1 STATUS2  PULSE  [UTIL ]>
```

Alarm Status	Pulse coder alarm status

Table B–5. Pulse Coder Alarm Status

				MSB	**B10**	**B9**	**B8**
				SPHAL	STBERR	CRCERR	DTERR
B7	**B6**	**B5**	**B4**	**B3**	**B2**	**B1**	**LSB**
OHAL	CSAL	BLAL	PHAL	RCAL	BZAL	CKAL	

SPHAL	When this bit is 1, it indicates a soft phase alarm (abnormal acceleration).
STBERR	When this bit is 1, it indicates a start/stop bit alarm.
CRCERR	When this bit is 1, it indicates a CRC alarm.
DTERR	When this bit is 1, it indicates a data alarm.
OHAL	When this bit is 1, it indicates a over heat alarm.
CSAL	When this bit is 1, it incicates a check sum alarm.
BLAL	When this bit is 1, it indicates the low voltage alarm of the battery.
PHAL	When this bit is 1, it indicates a phase alarm.
RCAL	When this bit is 1, it indicates a rotating speed counter abnormal alarm
BZAL	When this bit is 1, it indicates an exhausted battery alarm.
CKAL	When this bit is 1, it indicates a clock alarm.

Pulse screen

The pulse screen displays the servo delay, machine position, and status of the motion command.

```
STATUS Axis                         JOINT 30%
                                       GRP[1]
          Position    Machine       Motion
           Error       Pulse         Command
  J1:          0            0            0
  J2:          0            0            0
  J3:          0            0            0
  J4:          0            0            0
  J5:          0            0            0


[TYPE] STATUS1  STATUS2   PULSE [ UTIL ]
```

Position Error	Servo delay (pulses). Delay of the actual pulse to the command pulse
Machine Pulse	Machine position (pulses). Actual absolute pulses
Motion Command	Relative command pulses from the host (pulses)

Monitor screen

The monitor screen displays the current values, and the status of the position, overtravel, and servo amplifier. Load to the motor and thermal loss can be estimated using the root–mean–square current values.

```
 STATUS Axis                        JOINT  30 %
                                     GRP[ 1 ]
     Torque Monitor
           Ave. / Max.     Inpos OT VRDY
   J1:     0.000/    0.000    1    0  OFF
   J2:     0.000/    0.000    1    0  OFF
   J3:     0.000/    0.000    1    0  OFF
   J4:     0.000/    0.000    1    0  OFF
   J5:     0.000/    0.000    1    0  OFF



[ TYPE ]MONITOR TRACKING DISTURB[UTIL ]>
```

Ave.	Average of the root–mean–square current values (A)
Max.	Maximum of the root–mean–square current values (A)
Inpos	Position status (0 or 1)
OT	Overtravel status (0 or 1)
VRDY	Servo amplifier ready status (on or off)

Tracking screen

The tracking screen displays the status of the tracking servo system.

```
 STATUS Axis                           JOINT 30%
                                         GRP[1]
       Tracking Status
        Flag Bits1       Flag Bits2
   P1: 0000000000000000 0000000000000000
   P2: 0000000000000000 0000000000000000
        Alarm status     Counter Value
   P1: 000000000000              0
   P2: 000000000000              0


 [TYPE] MONITOR  TRACKING  DISTURB
```

Flag Bits 1	Servo alarm status 1
Flag Bits 2	Servo alarm status 2
Alarm Status	Pulse coder alarm status
Counter Value	Line tracking counter

For the servo and pulse coder alarm statuses, see Table B–3, Table B–4, and Table B–5

Disturbance torque screen

The disturbance torque screen displays the disturbance torque to each motor (current torque and maximum and minimum torque for each ITP). The disturbance torque is indicated with the current values estimated from the difference between the scheduled and actual values of the pulse coder. If the maximum or minimum value set for the disturbance torque is exceeded, the collision detection function of the servo system regards a collision as occurring and turns the servo power off.

```
STATUS Axis                          JOINT 30%
                                       GRP[1]
   Disturbance Torque
      Current       Max.           Min.
 J1:    0.000  /  0.0 (90.9)  / 0.0 (-90.9)
 J2:    0.000  /  0.0 (84.3)  / 0.0 (-84.3)
 J3:    0.000  /  0.0 (97.4)  / 0.0 (-97.4)
 J4:    0.000  /  0.0 (30.2)  / 0.0 (-30.2)
 J5:    0.000  /  0.0 (34.3)  / 0.0 (-34.3)


[TYPE]  MONITOR TRACKING  DISTURB [UTIL ]>
```

Current	Estimated disturbance torque to the servo motor (A)
Max.	Maximum value of the above estimated disturbance torque (A)
Min.	Minimum value of the above estimated disturbance torque (A)

Procedure B–13 Robot axis status screens

Step

1 Press the MENUS key to display the screen menu.

2 Select “0 —NEXT—” to display the next page, then select “4 STATUS.”

3 Press the F1 key, “[TYPE]” to display the screen change menu.

4 Select “Axis.” The robot axis status screens can be displayed.

```
 9 USER
 0 --NEXT--
```

MENUS

```
 3 DATA
 4 STATUS
 5 POSITION
```

```
 Axis

 TYPE
```

F1

```
STATUS Axis                      JOINT  30 %

                                   GRP[ 1 ]
      Flag Bits 1/2             History
 J1: 0000000000000000  (0000000000000000)
     0000000000000000  (0000000000000000)
 J2: 0000000000000000  (0000000000000000)
     0000000000000000  (0000000000000000)
 J3: 0000000000000000  (0000000000000000)
     0000000000000000  (0000000000000000)

[ TYPE ]STATUS1 STATUS2  PULSE   [UTIL ]>
```

```
[ TYPE ]MONITOR TRACKING DISTURB[UTIL ]>
```

– F2 “STATUS 1”: Displays the status 1 screen.
– F3 “STATUS 2”: Displays the status 2 screen.
– F4 “PULSE”: Displays the pulse screen.
– F2 “MONITOR” on the next page: Displays the monitor screen.
– F3 “TRACKING” on the next page: Displays the tracking screen.
– F4 “DISTURB” on the next page: Displays the disturbance torque screen.
– When F2 REG.DIS is selected on the next page, the regenerative discharge screen appears.

5 To change the group number, press F5 [UTIL]. A menu appears. On that menu, select 1 GROUP, then enter a desired group number.

B.5 Diagnosis Screen

B.5.1 Outline

This function is a function to show users very useful information at maintenance of the robot.
Each information has help that shows the description and the recommended action.
You can use the robot long time without trouble.
The following items are shown.

- Main (List)
- Reducer diagnosis
- Overheat diagnosis
- Torque diagnosis
- Disturbance diagnosis
- OVC diagnosis
- collision diagnosis
- Help

B.5.2 About Reducer Diagnosis

Servo diagnosis function includes reducer's recommended overhaul time diagnosis.
The overhaul time depends on the future motion of reducer. This function indicates the overhaul time in the motion of recent 50 hours.

When the exchange or overhaul of the reducer is done, you have to reset parameters.
Pleasae refer maintainance manual.

B.5.3 Procedure

Procedure B–14 Diagnosis screen

Step

1 Press the MENUS key to bring up the screen menu.
2 Select “4 STATUS” on the next page.
3 Press F1“[TYPE]” to display the pull–up menu.
4 Select “Axis”.
5 Press [next] key until “diag” is shown above function key.
6 Press F4“diag”. Diagnosis main screen is shown first.

```
 Diagnosis                          JOINT   10 %
                                              1/9
  group[ 1]
  reducer                    55542.2 hours
  overheat(motor)            11.31 %
  overheat(trans)            12.12 %
  current                    76.39 %
  disturbance                21.75 %
  OVC                        9.824 %
  collision detection        19 times
  discharge                  134 W



  [ TYPE ]  main reducer  ov.heat help   >
```

Procedure B–15 Change diagnosis screen

Step

1 Each item is allocated to the function key. Press function key to show the item.
For example, by pressing F3 key reducer diagnosis screen is shown.

```
 Diagnosis                          JOINT   10 %
                                              2/2
  reducer
  group[ 1]
  J1                         76863.3 hours
  J2                         57686.3 hours
  J3                         85768.6 hours
  J4                         93217.6 hours
  J5                         76876.8 hours
  **             ******************



  [ TYPE ]  main reducer  ov.heat help   >
```

2 You can change the allocation of function keys by pressing the [next] key.

```
Diagnosis                          JOINT   10 %
                                             2/2
 reducer
 group[ 1]
 J1                       76863.3 hours
 J2                       57686.3 hours
 J3                       85768.6 hours
 J4                       93217.6 hours
 J5                       76876.8 hours
 **           ******************




 torque  disturb  OVC    cl.det.
```

3 To show the Axis screen again, press [prev] key.

B.5.4 Each item

Main: Each item shows the value of the worst axis.

```
Diagnosis                          JOINT   10 %
                                             1/9
 group[ 1]
 reducer                  55542.2 hours
 overheat(motor)          11.31 %
 overheat(trans)          12.12 %
 current                  76.39 %
 disturbance              21.75 %
 OVC                      9.824 %
 collision detection      19 times
 discharge                134 W



 [ TYPE ]  main reducer  ov.heat help  >
```

Reducer: The time until the recomended overhaul of reducers

```
Diagnosis                          JOINT   10 %
                                             2/2
 reducer
 group[ 1]
 J1                       76863.3 hours
 J2                       57686.3 hours
 J3                       85768.6 hours
 J4                       93217.6 hours
 J5                       76876.8 hours
 **           ******************




 [ TYPE ]  main reducer  ov.heat help  >
```

Over heat: The ratio of root mean square current to the rated current

```
Diagnosis                         JOINT   10 %
                                           3/3
 over heat
 trans                       13.7 %
 motor  group[ 1]
        J1                   17.33 %
        J2                   23.63 %
        J3                   14.74 %
        J4                   21.65 %
        J5                   14.55 %
        **                  ****** %




 [ TYPE ]  main reducer  ov.heat help   >
```

Torque: The ratio of the current torque to the maximum.

```
Diagnosis                         JOINT   10 %
                                           2/2
 torque
 group[ 1]
        J1                   17.33 %
        J2                   23.63 %
        J3                   14.74 %
        J4                   21.65 %
        J5                   14.55 %
        **                  ****** %




 [ TYPE ]  main reducer  ov.heat help   >
```

Disturbance: The ratio of the force observed by the servo software to the alarm threshold.

```
Diagnosis                         JOINT   10 %
                                           2/2
 disturbance
 group[ 1]  current  max(%)  min(%)
 J1        23.63 %   32.33 / -23.52
 J2        26.37  %  45.21 / -23.43
 J3       -17.74  %  21.56 / -35.21
 J4        12.65  %  15.41 / -23.65
 J5       -16.55  %  19.37 / -16.55
 **       *******   ******   *******




 [ TYPE ]  main reducer  ov.heat help   >
```

OVC: The ratio of the temperature simulated by the software to the alarm threshold.

```
Diagnosis                              JOINT   10 %
                                                2/2
 OVC
 group[ 1]
        J1                17.33 %
        J2                23.63 %
        J3                14.74 %
        J4                21.65 %
        J5                14.55 %
        **               ****** %

 [ TYPE ]  main reducer  ov.heat help   >
```

Collision detection: The count of the collision and the data of the last collision detection.

```
Diagnosis                              JOINT   10 %
                                                3/3
 last detection
        2001/  4/  6, 16: 55: 26
 group[ 1]
 count / position
 J1      ***** times       -17.33  deg
 J2      ***** times        23.63  deg
 J3      ***** times       -14.74  deg
 J4      ***** times       121.65  deg
 J5         14 times       114.55  deg
 **                        ******  deg

 [ TYPE ]  main reducer  ov.heat help   >
```

Help: Information of the last shown item

```
Diagnosis                              JOINT   10 %
 INFORMATION:
 The count of the collision and the data
 of the last collision detection.
 REMEDY:
 If many collision detection occur,
 execute the overhaul more   frquently.
 DETAIL:
 You can watch the count of the    collisi-
 on detection ever   occured.You can also
 watch the information of the last de-
 tection(time, position).

 [ TYPE ]  main reducer  ov.heat help   >
```

B.6 World Frame Origin

This section describes the world frame origin of the each robot model (See Section 3.8,"Setting Coordinate Systems" for the world frame). When the user frame or tool frame is set, refer to this.

S series/LR Mate

(Other than S–450)
A crossing point between J1 axis and level plane which includes the J2 axis.

(S–450U/L)
Intersection of rotation axes U and γ when the robot is set to the zero positions on all linear axes

(S–450S)
Position at which rotation axis U, moved parallel to itself in the horizontal direction, intersects with rotation axis θ when the robot is set to the zero positions on all linear axes

M series

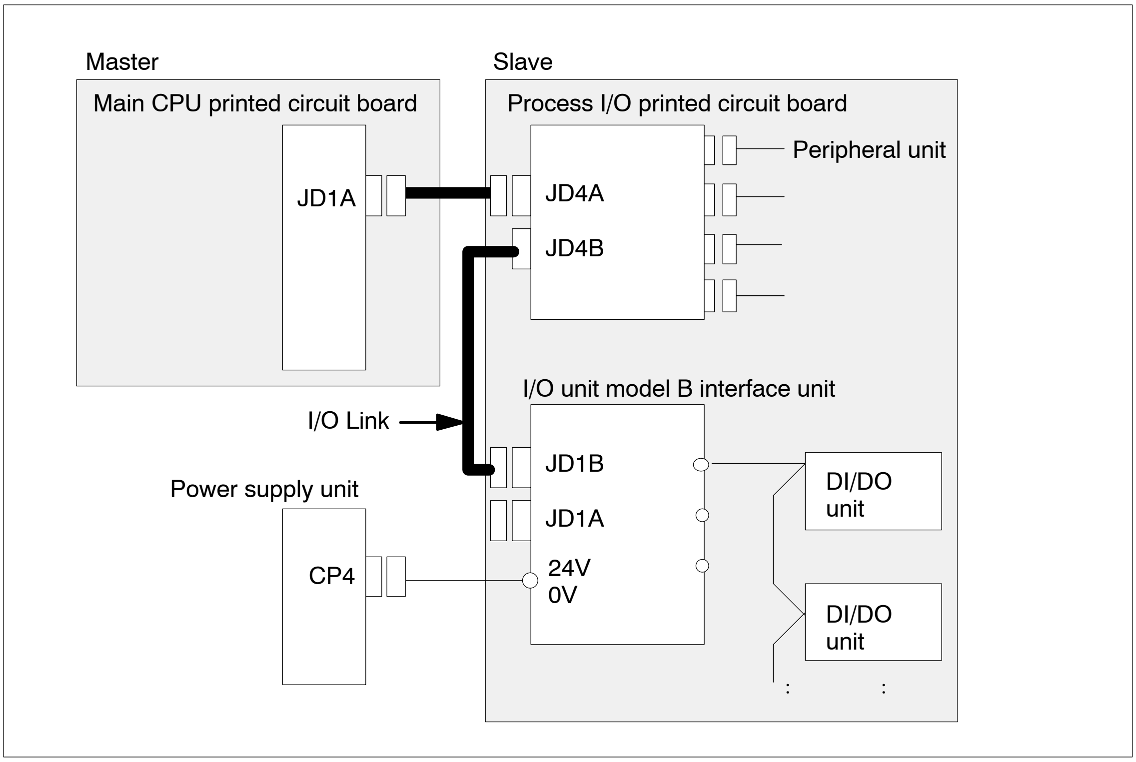
(M–410i/M–500)
Intersection of the J2–axis, moved parallel to itself, and the J1–axis

B.7 I/O Module Setting

FANUC I/O Link

The FANUC I/O Link is a serial interface used for high–speed I/O signal (bit data) transmission between the Robot controller and I/O modules, such as the process I/O printed circuit board and I/O Unit–MODEL A. Using the FANUC I/O Link, one master and multiple slaves can be connected. Generally, the Robot controller is used as the master, with the I/O modules connected to the controller being used as slaves. Up to 16 slave groups can be connected to one I/O Link.

Figure B–4. FANUC I/O Link configuration



I/O signals

The following I/Os are used for signal transmission between the Robot controller and system peripheral units, via the I/O modules connected to the FANUC I/O Link:

- Digital I/O SDI[i]/SDO[i]
- Group I/O GI[i]/GO[i]
- Analog I/O AI[i]/AO[i]
- Peripheral unit I/O UI[i]/UO[i] i = logical number

I/O modules

The following I/O modules can be connected to the Robot controller via the I/O Link:

Table B–6. I/O modules

	Abbreviation
Process I/O printed circuit board (CA, CB, DA, EA, EB, FA)	–
FANUC I/O Unit–MODEL A	I/O Unit – A
FANUC I/O Unit–MODEL B	I/O Unit – B
FANUC I/O Link connection unit	–
Programmable Controller SERIES 90–30A	–

Assignment

I/O logical number i is assigned to a physical number of I/O modules. I/O logical numbers can be redefined.

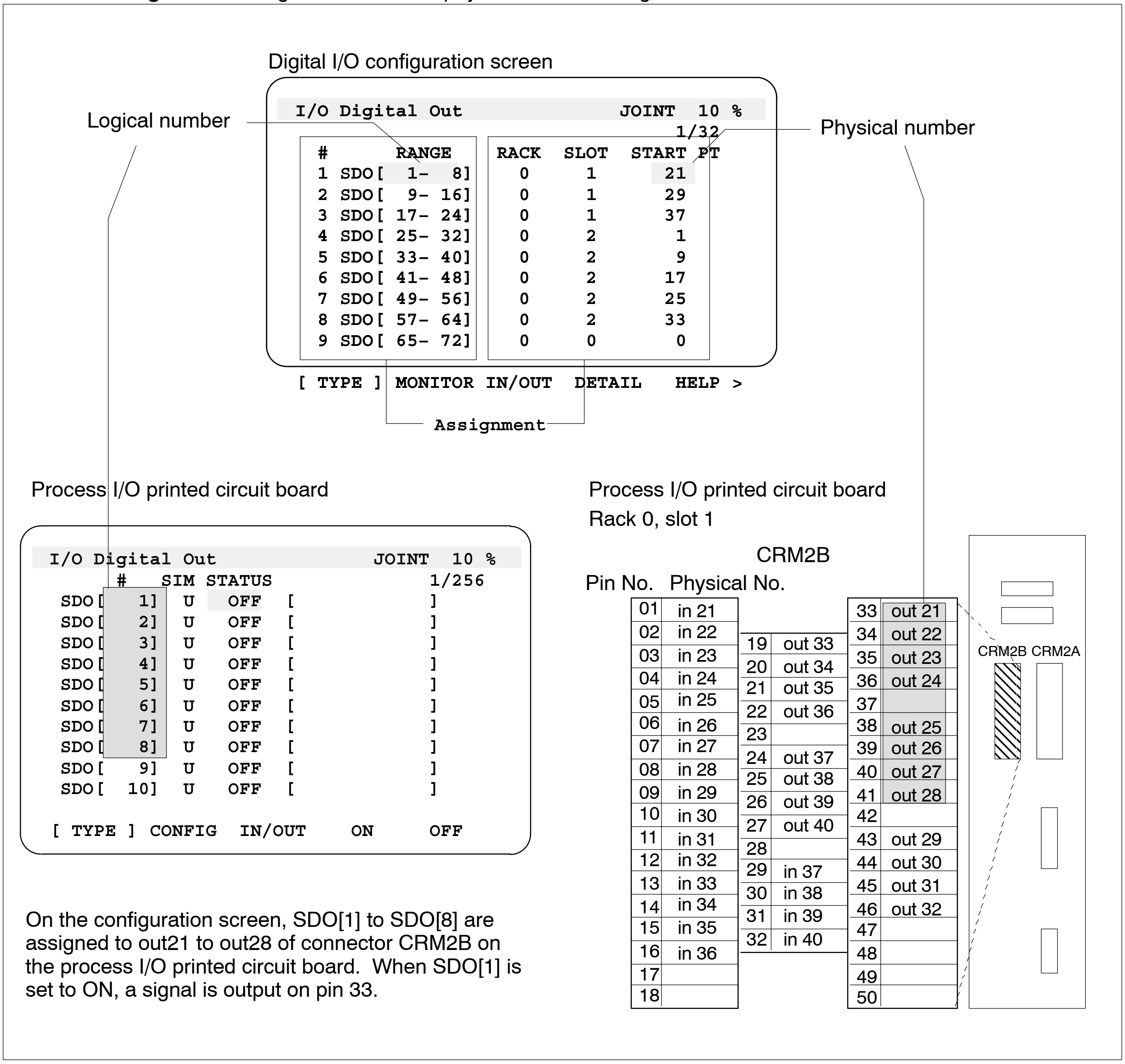
Logical number

I/O index used to reference an I/O in the Robot controller

Physical number

Number assigned to each signal pin of an I/O module. A specific signal pin of a particular I/O module can be specified with the rack, slot, and physical number.

Figure B–5. Logical number and physical number assignment



Rack number

Rack numbers indicate the hardware types and connection orders of I/O modules. I/O modules are classified into two major types: those having rack number 0, and those to which rack numbers are assigned in the order in which they are connected.

Slot number

I/O modules whose rack numbers are 0 are assigned slot numbers in the order in which they are connected. When the rack number of an I/O module is a non–zero value, indicating the order in which it is connected, a slot number is used to indicate the I/O module part of that I/O module. I/O module parts include, for example, modules of I/O Unit–A and DI/DO units of I/O Unit–B.

START PT (channel number)

Digital I/Os and peripheral I/Os are assigned in groups of eight signals. Specify the first physical number for eight sequential signals. For group I/Os, specify the first physical number for the sequential signals specified in NUM PTS. For an analog I/O, specify a channel number.

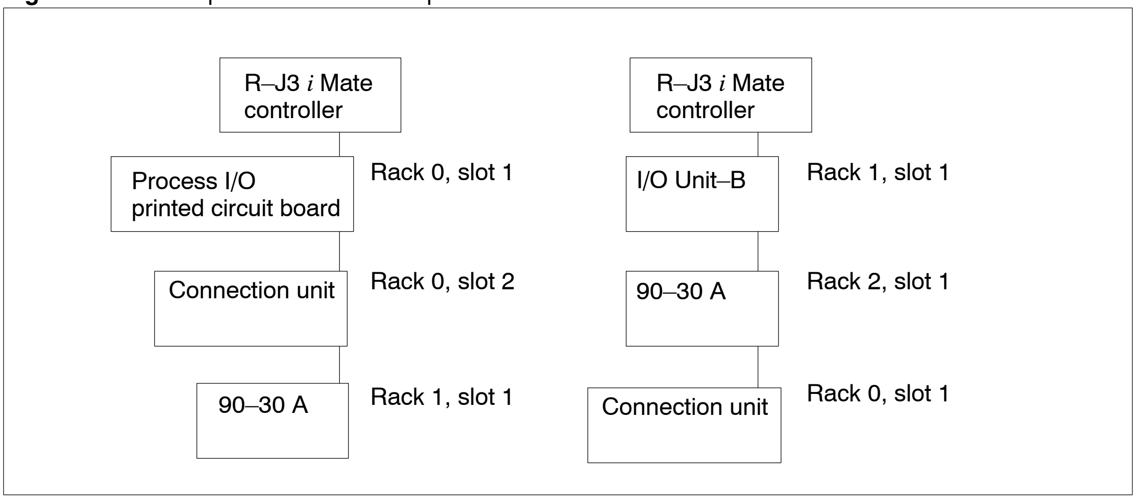
Table B–7. Specifying rack and slot numbers for each I/O module

I/O module	Rack	Slot
Process I/O printed circuit board	Always 0	(*2)
FANUC I/O Unit–MODEL A	(*1)	Number indicated on the base unit
FANUC I/O Unit–MODEL B	(*1)	Unit number (set with DIP switches)
FANUC I/O Link connection unit	Always 0	(*2)
Programmable Controller SERIES 90–30A	(*1)	1 (fixed)

NOTE *1 Numbers beginning with 1 are to be assigned to I/O modules, except those I/O modules having rack number 0, in the order in which they are connected.

NOTE *2 To those I/O modules having rack number 0, numbers beginning with 1 are to be assigned in the order in which they are connected.

Figure B–6. Example of rack and slot specification



I/O Link setting

When connected to the controller, some I/O modules require that the user make several additional specifications. Other I/O modules, however, do not require such specification.

When additional specification is not necessary

After connecting an I/O module to the Robot controller, via a cable, turn on the power. Data assignment is performed automatically.

When additional specification is necessary

Specify the system variables from the Robot controller.

	Specification
Process I/O printed circuit board (CA, CB, DA)	Unnecessary
FANUC I/O Unit–MODEL A	Unnecessary
FANUC I/O Unit–MODEL B	Necessary
FANUC I/O Link connection unit	Necessary
Programmable Controller SERIES 90–30A	Necessary

Number of available I/Os

Up to 16 slave groups can be connected to each I/O Link. Therefore, up to 16 I/O modules can be connected to the Robot controller.

The FANUC I/O Link supports 1024 inputs and 1024 outputs for a master. These I/Os are assigned to the slaves to enable the periodic transmission of I/O data between the master and slaves. The total number of I/Os used by the slaves connected to the FANUC I/O Link must satisfy the following:

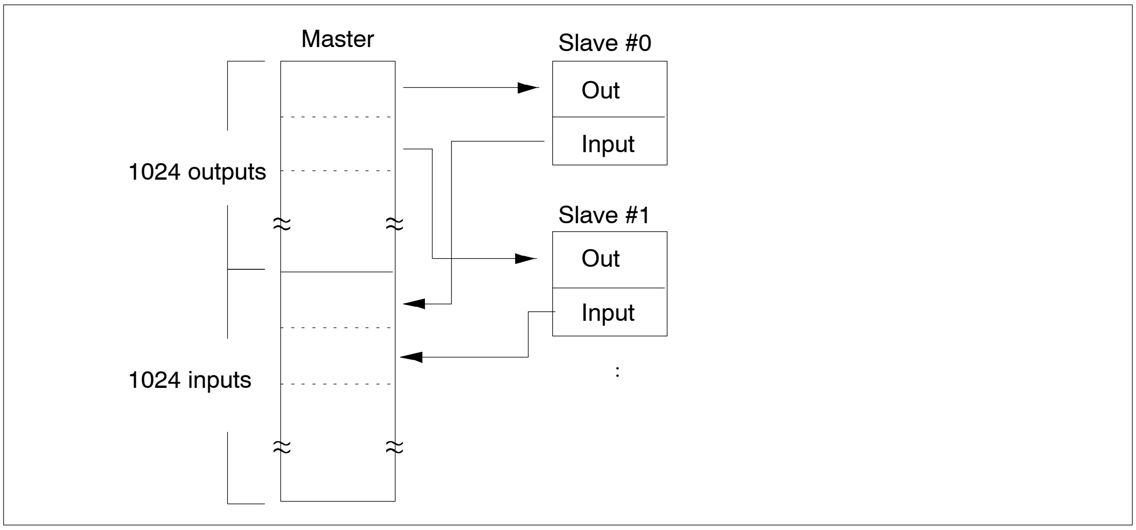
Number of inputs per I/O Link =< 1024

Number of outputs per I/O Link =< 1024

Therefore, I/Os can be expanded within the above range.

For details of the number of I/Os used for each I/O module that becomes a slave, refer to the relevant I/O module manual. The process I/O printed circuit board, however, always uses 128 inputs and 128 outputs, regardless of its type.

Figure B–7. Relation between master and slave in I/O signal points



Teach pendant display

On both the digital input and output screens, displayed on the teach pendant of the Robot controller, up to 256 signals can be displayed. Using these screens, a user can specify and change the assignment of up to 256 signals.

On both the analog input and output screens, up to 25 channels can be displayed. Using these screens, a user can specify and change the assignment of up to 25 channels, on a channel–by–channel basis.

I/O module manuals

For details of each I/O module, refer to the following manuals:

I/O module name	Manual name	Drawing number
Process I/O printed circuit board (CA, CB, DA, EA, EB, FA)	FANUC Robot series R–J3 *i* Mate Controller Maintenance Manual	B–81525EN
FANUC I/O Unit–MODEL A	FANUC I/O Unit–MODEL A Connection and Maintenance Manual	B–61813
FANUC I/O Unit–MODEL B	FANUC I/O Unit–MODEL B Connection Manual	B–62163
FANUC I/O Link connection unit	FANUC I/O Link Connection Unit Specifications	A–68806
Programmable Controller SERIES 90–30A	Programmable Controller SERIES 90–30A User's Manual	B–76014

B.8 Positioner setup

Step 1 Turn ON the controller with " PREV" key and F! key pressed.
Then select "3. Controlled start".

2 Press MENUS key and select "9. MAINTENANCE".

3 You will see similar screen to the following one.

```
Setup Robot System Variables

  Group    Robot      Library /Option    Ext Axs

   1     R–2000i/165F        *

   2     POSITIONER

 [Type]   ORD_NO    AUTO    MANUAL
```

Press arrow(↑,↓) keys and move the cursor to "POSITIONER". Then press F4,"MANUAL".

4 You will see similar screen to the following

```
— Hardware start axis setting —
enter Hardware start axis (1..16)?
Default value = 1
```

Enter axis number and press ENTER key.
* Which axis in the system is assigned to 1st axis of POSITIONER is set in this screen. For example, if the system has R–2000i and POSITIONER, start axis number of POSOTIONER is 7 because R–2000i has 6 axes.

5 You will see similar screen to the following

```
— Kinematics Type Setting —
1:Known Kinematics
2:Unknown Kinematics
Select Kinematics Type?
default value = 1
```

If the measurements of offset values between POSITONER axes are accurately known, item 1 should be selected. Otherwize item 2 should be selected.

6 You will see similar screen to following one.

Group number is displayed instead of "?" in following screen.
Total number of axes is displayed instead of "#" in following screen.
Initial value of number of axes is 0.

```
****Group ? Total POSITIONER Axis=#
1.Display/Modify POSITIONER axis
2.Add POSITIONER axis
3.Delete POSITIONER axis
4.Exit
Select item?
```

If you want to add POSITIONER axis, select "2. Add POSITIONER axes".
Then setup procedure starts.

If you want to delete POSITIONER axis, select "3. Delete POSITIONER axes".
Then following screen is displayed.

```
POSITIONER Axis ? Was Deleted
Press ENTER to Continue.
```

(* The deleted axis number is displayed instead of "?" in above screen.)
After this setup, please set values according to the specification of the mechanism.

7 Select the Motor size

```
****POSITIONER Axis 1 Initialization****
33.ACb0.5      38.ACa12       43.ACa100
34.ACal        39.ACa22       44.ACa150
:              :              :
:              :              :
0. Next page.
Select Motor size?
```

8 Select the motor type.

```
MOTOR TYPE
1./2000        6.F/3000
2./3000        7.F/2500
3.S/2000       8.L/3000
:              :
Select Moter Type?
```

9 Select Amplifier Current Limit.

```
CURRENT LIMIT FOR AMPLIFIER

1. 2A          6. 60A
2. 4A          7. 80A
3. 12A         8. 100A
:              :
Select Amplifier Current Limit?
```

10 Set amplifier number.

```
— Amplifier number Setting –
Enter Amplifier Number (1→16)?
```

11 Set amplifier type.

```
— Amplifier Type Setting –
Amplifier ? Type = #
Enter (1:Change, 2:No Change)?
```

* Amplifier number which is set in previous procedure is displayed instead of “?”.
* If 0 is displayed instead of “#”, this indicates that amp type is not set yet.

If you select “1: Change” in above screen, you will see following screen.

Select the amplifier type.

```
SELECT AMP TYPE
1. A06B–6100 series 6 axes amplifier
2. A06B–6093 Beta series (FSSB)
```

12 Select the axis type.

```
Axis Type Setting –
1: Linear Axis
2: Rotary Axis
Select Axis Type?
```

If the axis does linear motion, select item 1.

If the axis does rotary motion, select item 2.

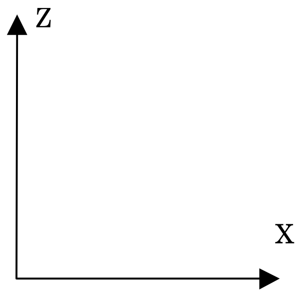
13 Select the direction of the axis.

```
— Direction Setting –
1: +X        3: +Y        5: +Z
2: –X        4: –Y        6: –Z
Select Direction?
```

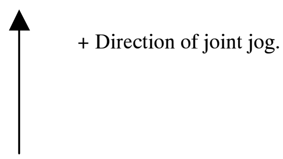
Directions in above screen indicate the directions of axes of the world coordinate system. The +/– direction must be considered in this setting.

Example)

World coordinate frame

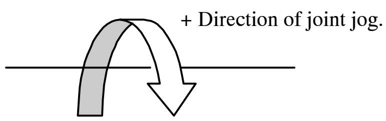


· Linear axis



In this case, the direction should be set to “+Z”.

· Rotary axis



In this case, the direction should be set to “+X”.

14 If you set the Kinematics Type to “Known Kinematics” in procedure 4., you will see the following screens. If the Kinematics is “Unknown Kinematics”, this procedure is skipped.

Enter the Offset value in X direction.

```
— Offset Setting –
Enter Offset X (mm)?
```

Enter the Offset value in Y direction.

```
Enter Offset Y (mm)?
```

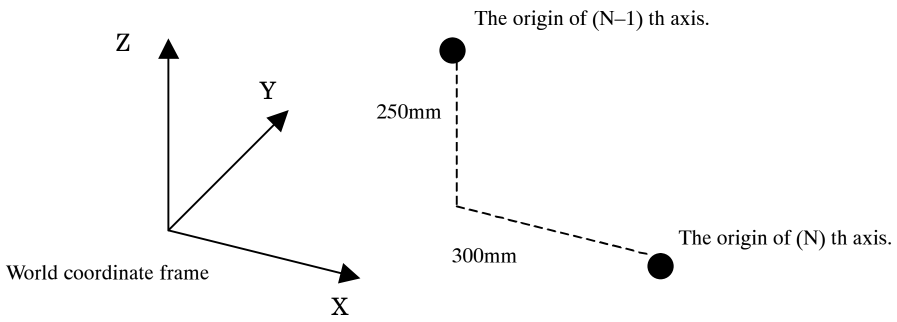
Enter the Offset value in Z direction.

```
Enter Offset Z (mm)?
```

For 1st axis, offset values between the origin of the world coordinate and that of the axis must be set.

For the 2nd or later axes, offset values between the origin of the axis and that of the previous axis must be set.

Example)



In this case, offset values for (N)th axis must be set as follows.

Offset X: 300mm

Offset Y: 0mm

Offset Z: – 250mm

15 Set Gear Ratio.

For a linear axis, enter the distance of the motion which corresponds to one revolution of the the axis of the motor.(UNIT: mm/rev)

Following screen is displayed for linear axes.

```
— Gear Ratio Setting —
Enter Gear Ratio (mm/rev)?
```

For a rotary axis, enter the number of revolution of the motor which corresponds to one revolution of the axis. (UNIT: motor_rev / axis–rev)

Following screen is displayed for rotary axes.

```
— Gear Ratio Setting —
Enter Gear Ratio (mot–rev/axs–rev)?
```

16 Set the maximum speed for the axis.

You will see the similar screen to following one.

```
— Maximum Speed Setting —
Suggested Speed = 150.000 (mm/sec)
(Calculated with Max Motor Speed)
Enter (1: Change, 2: No Change)?
```

If you want to change suggested value, select “1: Change”.

Then following screen will be displayed.

```
Enter Max Speed (mm/sec)?
```

Enter Max speed.

17 Set motion sign.

```
MOTOR DIRERCTION
Ext_axs 1 Motion Sign = TRUE
Enter ( 1: TRUE, 2: FALSE)?
```

18 Set the upper limit of the POSITIONER axis (UPPER LIMITS). Input unit of it by the “mm” in case of linear axis and by “degree” in case of rotary axis.

```
UPPER  LIMITS
Enter  Upper Limit  ( deg ) ?
```

warning) Determine upper limit of POSITIONER axis by user. So, the following condition must be consisted:

<upper limit> – <lower limit> = length of POSITIONER axis

For example, if the length of POSITIONER axis is 100 mm, you may set the upper limit.

<upper limit> = 50mm

<lower limit> = –50mm

19 81DSet the lower limit of the POSITIONER axis (LOWER LIMITS). Input unit of it by the "mm" in case of linear axis and by "degree" in case of rotary axis.

```
LOWER LIMITS
Enter Lower Limit ( deg ) ?
```

20 Set the mastering position data (MASTER POSITION).

```
MASTER POSITION
Enter Master Position ( deg ) ?
```

21 Set the constant of acceleration/deceleration time (ACC/ DEC TIME) .

Set the value when you change the constant of acceleration/deceleration time of the first joint. In case of changing it, input "1", or in case of using the recommending value, input "2".

```
ACC/DEC TIME
Default acc_time1=256(ms)
Enter ( 1: Change, 2: No Change ) ?
```

22 Set the value when you change the constant of acceleration/deceleration time of the second joint. In case of changing it, input "1". In case of using the recommending value, input "2".

```
Default acc_time2=128(ms)
Enter ( 1: Change, 2: No Change ) ?
```

23 Set the value when you change the constant of exponential acceleration/deceleration time of first joint. In case of changing it, input "1". In case of using the recommending value, input "2".

```
EXP_ACCEL TIME
Default exp_accel time =0(ms)
Enter (1: Change, 2: No Change)?
```

24 Set the "Minumum Access Time". This value is used when the real acceleration/deceleration time is smaller than specified time. In case of changing it, input "1". In case of not changing, input "2".

```
MIN_ACCEL TIME
Default min_accel time =384(ms)
Enter ( 1: Change, 2: No Change ) ?
```

25 Set the inertia ratio of all load inertia caluculated in moter axis to inertia (to rotary inertia). On setting inertia ratio, its value must be 1< the value<5. On not setting it, input "0".

```
Load Ratio is
Load Inertia ( Kg*cm*s*2)
Motor Inertia (Kg*cm*s*2)
Enter Load ratio ? ( 0:None 1→5: Valid)
```

26 Set the brake number (0–4) using the POSITIONER axis.

```
BRAKE SETTING
Enter Brake Number (0→4)?
```

27 Select the type of brake control. On valid of brack contrl, choose "1:Enable" and input the delay time of brake control. On invalid of it, choose "2:Disable".

```
SERVO  TIMEOUT
Servo  Off   is  Enable
Enter (1: Enable  2: Disable)?
Select?
```

(On choosing "1:Enable")

```
Enter  Servo  Off  Time ?  (0.0→30.0  Sec)
```

28 Come back screen of step 6.

```
**** Group ? Total POSITIONER Axis = # ****
1.Display/Modify POSITIONER Axis = #
2.Add POSITIONER axis
3.Delete POSITIONER axis
4.Exit
Select item ?
```

- In case of displaying/modifying the POSITIONER axis setting, select "1.Display/Modify Ext axis".
- When you set the POSITIONER axis successively, select the item "2" and text and continues after the step 7 in this text.
- In case of deleting the POSITIONER axis, select "3.Delete Ext axis".
- In case of finishing the setting, select "4.EXIT→0.EXIT".

B.9 Extended Axis Setup

Step 1 Turn ON the controller with “ PREV” key and “F→” key pressed.

Then select “3. Controlled start”.

2 Press MENU key and select “9. MAINTENANCE”.

3 You will see similar screen to the following one.

```
Setup  Robot  System  Variables
    Group     Robot        Library /Option      Ext Axs
      1     R–2000i/165F          *
              Extended     Axis   Control
   [Type]   ORD_NO      AUTO       MANUAL
```

Press arrow(↑,↓) keys and move the cursor to “Extended Axis Control”. Then press F4,”MANUAL”.

4 You will see similar screen to the following. Select the group of the extended axis and input its number.

```
**** EXTENDED  AXIS  SETTING  PROGRMA ****
SELECT   GROUP
0.              EXIT
1.              Group1
```

Display the information about extended axis of the selected group

```
                              E1 E2 E3
Group  1  Total  Ext  Axis =   *   *   *
1.Display/Modify  Ext  axis
2.Add  Ext  axes
3.Delete  Ext  axes
4.Exit
Select?
```

Select “2. Add Ext axes” in case of setting new extended axis.

5 Set the number of extended axis. For the first extended axis of the group, input “1”, for the second extended axis of the group, input ”2”, for the third axis of the group, input “3”. You should set the number from “1” in turn.

```
Enter  axis  to  add  (1→3) ?
```

6 You will see the initial setup screen of extended axis. Select the kind of motor used extended axis.

```
**** Ext  Axis  1  Initialization ****
33.ACb0.5      38.ACa12      43.ACa100
34.ACa1        39.ACa22      44.ACa150
:               :              :
:               :              :
0.              Next page.
Select ?
```

7 Select the type of motor on screen.

```
MOTOR TYPE
1. /2000      6.F/3000
2. /3000      7.F/2500
3. S/2000     8.L/3000
:             :
Select ?
```

8 Select the max current value of motor on screen.

```
CURRENT LIMIT FOR AMPLIFIER

1.  2A      6.  60A
2.  4A      7.  80A
3.  12A     8.  100A
:           :
Select ?
```

9 Select the type of extended axis through the four types to the following.

```
EXTENDED AXIS TYPE
1. Integrated Rail (Linear axis)
2. Integrated Arm (Rotary axis)
3. Auxiliary Linear Axis
4. Auxiliary Rotary Axis
Select ?
```

Warning 1)

Integrated: Robot coordinate is added to the distance of extended axis. The world coordinates are unchanged by the changed extended axis. So the current position changes only the distance transfered by the extended axis.
Auxiliary: Robot coordinate is NOT added to the distance of extended axis. World coordinate is transferred with changed extended axis, and remains the fixed robot coordinate.
Warning 2)
Integrated Rail (Linear axis):

Set the direction of attaced extended axis to the direction (X,Y,Z) of world coordinate

```
Direction 1:X 2:Y 3:Z
Enter Direction (1→3) ?
```

Integrated Arml (Rotary axis):

Set the offset length to Z direction between the origin of rotary center of extended axis and the origin of robot coordinate

```
Enter Off Set Length (mm) ?
```

Next, set the arm length of extended axis.

Correspondence of the X–axis of robot coordinate and the rotary axis of extended axis:

Set the offset length to the Y direction between the origin of rotary center of extended axis and the origin of robot coordinate.

Correspondence of the Y–axis of robot coordinate and the rotary axis of extended axis:

Set the offset length to the X direction between the origin of rotary center of extended axis and the origin of robot coordinate.

Correspondence of the Z–axis of robot coordinate and the rotary axis of extended axis:

Set the offset length to the X direction between the origin of rotary center of extended axis and the origin of robot coordinate.

```
Enter Arm Length (mm) ?
```

Finally, set the direction of attacehed extended axis. Input the direction of the rotary axis to the axis (X,Y,Z) of world coordinate

10 Set the gear ratio (GEAR RATIO). For linear axis, the gear ratio is in “mm” of travel per revolution of motor. For rotary axis, the gear ratio is in motor turns per single rotations of the rotary axis.
For linear axis:

```
GEAR RATIO

For a linear axis it is the number of
Mm’s traveled for one rotation of the
Motor
Enter Gear Ratio ? (mm)
```

For rotary axis:

```
GEAR RATIO
Enter Gear Ratio ?
```

11 Set the max joint speed. You will see the max rotary numbers and gear ratio on screen. With Changing, input “1”and the value. With using the recommending value, input “2”.

```
MAX JOINT SPEED SETTING
Suggested Speed = 150.000 ( deg / s )
( Calculated with Max motor speed)
Enter ( 1: Change, 2: No Change ) ?
```

In case of changing the max speed

```
Enter Max Speed (mm/sec) ?
```

12 Set the direction of extended axis to the motor axis. If the motion direction is positive to the positive rotation of motor, input “1”. If the motion direction is negative to the negative rotation of motor, input “2”.

```
MOTOR DIRERCTION
Ext_axs 1 Motion Sign = TRUE
Enter ( 1: TRUE, 2: FALSE) ?
```

13 Set the upper limit of the extended axis (UPPER LIMITS). Input unit of it by the “mm” degree in case of linear axis and by “degree” in case of rotary axis.

```
UPPER LIMITS
Enter Upper Limit ( deg ) ?
```

warning) Determine upper limit of extended axis by user. So, the following condition must be consisted:

<upper limit> – <lower limit> = length of extended axis

For example, if the length of extended axis is 100 mm, you may set the upper limit.

<upper limit> = 50mm

<lower limit> = –50mm

14 Set the lower joint orient area of the extended axis (LOWER LIMITS). Input unit of it by the “mm” degree in case of linear axis and by “degree” in case of rotary axis.

```
LOWER LIMITS
Enter Lower Limit ( deg ) ?
```

15 Set the mastering position data (MASTER POSITION).

```
MASTER POSITION
Enter Master Position ( deg ) ?
```

16 Set the constant of acceleration/deceleration time (ACC/ DEC TIME) . Set the value when you change the constant of acceleration/deceleration time of the first joint. In case of changing it, input “1”, or in case of using the recommending value, input “2”.

```
ACC/DEC TIME
Default acc_time1=256(ms)
Enter ( 1: Change, 2: No Change ) ?
```

Set the value when you change the constant of acceleration/deceleration time of the second joint. In case of changing it, input “1”, or in case of using the recommending value, input “2”.

```
Default acc_time2=128(ms)
Enter ( 1: Change, 2: No Change ) ?
```

17 Set the value when you change the constant of exponential acceleration/deceleration time of first joint. In case of changing it, input “1”, or in case of using the recommending value, input “2”.

```
EXP_ACCEL TIME
Default exp_accel time =0(ms)
Enter ( 1: Change, 2: No Change ) ?
```

18 Set the “Minumum Access Time”. This value is used when the real acceleration/deceleration time is smaller than specified time. In case of changing it, input “1”, or in case of not changing, input “2”.

```
MIN_ACCEL TIME
Default min_accel time =384(ms)
Enter ( 1: Change, 2: No Change ) ?
```

19 Set the inertia ratio of all load inertia calculated in motor axis to inertia (to rotary inertia). On setting inertia ratio, its value must be 1< the value<5. On not setting it, input “0”.

```
Load Ratio is
Load Inertia ( Kg*cm*s*2)
Motor Inertia (Kg*cm*s*2)
Enter Load ratio ? ( 0:None 1→5: Valid)
```

20 Set the amplifier number (AMP NUMBER).

```
SELECT AMP NUMBER
Enter amplifier number (1→16) ?
```

21 Set the type of amplifier (AMP TYPE).

```
SELECT AMP TYPE
1. A06B–6100 series 6 axes amplifier
2. A06B–6093 Beta series (FSSB)
```

22 Set the brake number (0–4) using the POSITIONER axis.

```
BRAKE SETTING
Enter Brake Number (0→4)?
```

23 Select the type of brake control. For brake control, choose “1:Enable” and input the delay time of brake control. To disable brake control, choose “2:Disable”.

```
SERVO  TIMEOUT
Servo  Off   is  Enable
Enter (1: Enable  2: Disable) ?
Select?
```

(On choosing “1:Enable”)

```
Enter  Servo  Off  Time ?  (0.0→30.0  Sec)
```

24 You will see a screen similar to the following.

```
**** Group  1  Total  Ext  Axis = ****
1.Display/Modify  Ext  axis
2.Add  Ext  axes
3.Delete  Ext  axes
4.EXIT
Select?
```

- In case of displaying/modifying the extended axis setting, select “1.Display/Modify Ext axis”.
- When you set the extended axis successively, select the item “2” and text and contines after the step 5 in this text.
- In case of deleting the extended axis, select “3.Delete Ext axis”.
- In case of finishing the setting, select “EXIT→4.EXIT”.

B.10 Extended Axis Setup

Step 1 Turn ON the controller with " PREV" key and "F→" key pressed.
Then select "3. Controlled start".
2 Press MENU key and select "9. MAINTENANCE".
3 You will see similar screen to the following one.

```
Setup Robot System Variables
    Group      Robot      Library /Option     Ext Axs
      1        S–430iF    Floor  Mnt             *
               Extended   Axis   Control
  [Type]     ORD_NO     AUTO         MANUAL
```

Press arrow(↑,↓) keys and move the cursor to "Extended Axis Control". Then press F4,"MANUAL".
4 You will see similar screen to the following. Select the group of the extended axis and input its number.

```
**** EXTENDED AXIS SETTING PROGRMA ****
SELECT  GROUP
0              EXIT
1              Group1
```

Display the information about extended axis of the selected group

```
                                E1 E2 E3
****Group 1 Total Ext Axis =  *  *  *
1.Display/Modify Ext axis
2.Add Ext axes
3.Delete Ext axes
4.Exit
Select?
```

Select "2. Add   Ext axes" in case of setting new extended axis.
5 Set the number of extended axis. For the first extended axis of the group, input "1", for the second extended axis of the group, input "2", for the third axis of the group, input "3". You should set the number from "1" in turn.

```
Enter axis to add (1→3) ?
```

6 You will see the initial setup screen of extended axis. Select the kind of motor used extended axis.

```
**** Ext Axis 1 Initialization ****
33.ACb0.5      38.ACa12      43.ACa100
34.ACa1        39.ACa22      44.ACa150
:              :             :
:              :             :
0.Next page.
Select ?
```

7 Select the type of motor on screen.

```
MOTOR TYPE
1. /2000      6.F/3000
2. /3000      7.F/2500
3. S/2000     8.L/3000
:              :
Select ?
```

8 Select the max current value of motor on screen.

```
CURRENT LIMIT FOR AMPLIFIER


1.  2A       6.  60A
2.  4A       7.  80A
3.  12A      8.  100A
:             :
Select ?
```

9 Select the type of extended axis through the four types to the following.

```
EXTENDED  AXIS TYPE
1. Integrated Rail (Linear axis)
2. Integrated Arm (Rotary axis)
3. Auxiliary Linear Axis
4. Auxiliary Rotary Axis
Select ?
```

Warning 1)

Integrated: Robot coordinate is added to the distance of extended axis. The world coordinate is unchanged by the changed extended axis. So current position changes the only distance transfered extended axis.
Auxiliary: Robot coordinate is NOT added to the distance of extended axis. World coordinate is transferred with changed extended axis, remain fixed robot coordinate.
Warning 2)
Integrated Rail (Linear axis):

Set the direction of attaced extended axis to the direction (X,Y,Z) of world coordinate

```
Direction 1:X 2:Y 3:Z
Enter Direction (1→3) ?
```

Integrated Arml (Rotary axis):

Set the offset length to Z direction between the origin of rotary center of extended axis and the origin of robot coordinate

```
Enter Off Set Length (mm) ?
```

Next, set the arm length of extended axis.

Correspondence of the X–axis of robot coordinate and the rotary axis of extended axis:

Set the offset length to the Y direction between the origin of rotary center of extended axis and the origin of robot coordinate.

Correspondence of the Y–axis of robot coordinate and the rotary axis of extended axis:

Set the offset length to the X direction between the origin of rotary center of extended axis and the origin of robot coordinate.

Correspondence of the Z–axis of robot coordinate and the rotary axis of extended axis:

Set the offset length to the X direction between the origin of rotary center of extended axis and the origin of robot coordinate.

```
Enter Arm Length (mm) ?
```

Finally, set the direction of attacehed extended axis. Input the direction of the rotary axis to the axis (X,Y,Z) of world coordinate

10 Set the gear ratio (GEAR RATIO). For linear axis, the gear ratio is in “mm” of travel per revolution of motor. For rotary axis, the gear ratio is in motor turns per single rotations of the rotary axis.
For linear axis:

```
GEAR RATIO

For a linear axis it is the number of
Mm’s traveled for one rotation of the
Motor
Enter Gear Ratio ? (mm)
```

For rotary axis:

```
GEAR RATIO
Enter Gear Ratio ?
```

11 Set the max joint speed. You will see the max rotary numbers and gear ratio on screen. With Changing, input “1”and the value. With using the recommending value, input “2”.

```
MAX JOINT SPEED SETTING
Suggested Speed = 150.000 ( deg / s )
( Calculated with Max motor speed)
Enter ( 1: Change, 2: No Change ) ?
```

In case of changing the max speed

```
Enter Max Speed (mm/sec) ?
```

12 Set the direction of extended axis to the motor axis. If the motion direction is positive to the positive rotation of motor, input “1”. If the motion direction is negative to the negative rotation of motor, input “2”.

```
MOTOR DIRERCTION
Ext_axs 1 Motion Sign = TRUE
Enter ( 1: TRUE, 2: FALSE) ?
```

13 Set the upper limit of the extended axis (UPPER LIMITS). Input unit of it by the “mm” degree in case of linear axis and by “degree” in case of rotary axis.

```
UPPER LIMITS
Enter Upper Limit ( deg ) ?
```

warning) Determine upper limit of extended axis by user. So, the following condition must be consisted:

<upper limit> – <lower limit> = length of extended axis

For example, if the length of extended axis is 100 mm, you may set the upper limit.

<upper limit> = 50mm

<lower limit> = –50mm

14 Set the lower joint orienta area of the extended axis (LOWER LIMITS). Input unit of it by the “mm” degree in case of linear axis and by “degree” in case of rotary axis.

```
LOWER LIMITS
Enter Lower Limit ( deg ) ?
```

15 Set the mastering position data (MASTER POSITION).

```
MASTER POSITION
Enter Master Position ( deg ) ?
```

16 Set the constant of acceleration/deceleration time (ACC/ DEC TIME) . Set the value when you change the constant of acceleration/deceleration time of the first joint. In case of changing it, input “1”, or in case of using the recommending value, input “2”.

```
ACC/DEC TIME
Default acc_time1=256(ms)
Enter ( 1: Change, 2: No Change ) ?
```

Set the value when you change the constant of acceleration/deceleration time of the second joint. In case of changing it, input “1”, or in case of using the recommending value, input “2”.

```
Default acc_time2=128(ms)
Enter ( 1: Change, 2: No Change ) ?
```

17 Set the value when you change the constant of exponential acceleration/deceleration time of first joint. In case of changing it, input “1”, or in case of using the recommending value, input “2”.

```
EXP_ACCEL TIME
Default exp_accel time =0(ms)
Enter ( 1: Change, 2: No Change ) ?
```

18 Set the “Minumum Access Time”. This value is used when the real acceleration/deceleration time is smaller than specified time. In case of changing it, input “1”, or in case of not changing, input “2”.

```
MIN_ACCEL TIME
Default min_accel time =384(ms)
Enter ( 1: Change, 2: No Change ) ?
```

19 Set the inertia ratio of all load inertia calculated in motor axis to inertia (to rotary inertia). On setting inertia ratio, its value must be 1< the value<5. On not setting it, input “0”.

```
Load Ratio is
Load Inertia ( Kg*cm*s*2)
Motor Inertia (Kg*cm*s*2)
Enter Load ratio ? ( 0:None 1→5: Valid)
```

20 Set the amplifier number (AMP NUMBER).

```
SELECT AMP NUMBER
Enter amplifier number (1→16) ?
```

21 Set the type of amplifier (AMP TYPE).

```
SELECT AMP TYPE
1. A06B–6100 series 6 axes amplifier
2. A06B–6093 Beta series (FSSB)
```

22 Set the brake number (0–4) using the POSITIONER axis.

```
BRAKE SETTING
Enter Brake Number (0→4)?
```

23 Select the type of brake control. On valid of brake control, choose “1:Enable” and input the delay time of brake control. On invalid of it, choose “2:Disable”.

```
SERVO  TIMEOUT
Servo  Off   is  Enable
Enter (1: Enable  2: Disable) ?
Select?
```

(On choosing “1:Enable”)

```
Enter  Servo  Off  Time ?  (0.0→30.0  Sec)
```

24 Come back screen of step 6 in this text.

```
**** Group  1  Total  Ext  Axis = ****
1.Display/Modify  Ext  axis
2.Add  Ext  axes
3.Delete  Ext  axes
4.EXIT
Select?
```

- In case of displaying/modifying the extended axis setting, select “1.Display/Modify Ext axis”.
- When you set the extended axis successively, select the item “2” and text and contines after the step 5 in this text.
- In case of deleting the extended axis, select “3.Delete Ext axis”.
- In case of finishing the setting, select “EXIT→4.EXIT”.

B.11 Independent Additional Axis Board (Nobot) Startup Procedure

Step

1 Execute control start:
Press and hold down the “PREV” and “F→” keys and turn the power switch ON.
Then, select 3. Control start.

2 On the teach pendant, select Screen selection, then 9. Robot setting.

3 The following screen appears:

```
Setup  Robot  System  Variables

   Group     Robot       Library /Option      Ext Axs

     1     S–430iF      Floor  Mnt              0

     2     NOBOT                                0


  [Type]    ORD_NO      AUTO       MANUAL
```

Position the cursor on 1NOBOT and press the F4 key, MANUAL.

4 The following screen appears. For this setting item, specify that the first axis of the independent additional axis board (NOBOT) should be the n–th axis in the entire system. For the S–430*i*F plus NOBOT, for example, because the S–430*i*F as the first group is a 6–axis robot, the first axis of the independent additional axis board (NOBOT) as the second group will be the seventh.

```
— Hardware start axis setting —
  enter Hardware start axis (1..16)?
Default value = 1
```

5 The following screen appears.
The “?” on the screen will be replaced by a group number.
The “#” on the screen will be replaced by the number of axes of the NOBOT currently set.

```
  **** Group  ?  Total  Nobot  Axis = #
     1.Display/Modify Nobot axis 1→6
     2.Add Nobot axis
     3.Delete Nobot axis
     4.Exit
Select item?
```

To add an axis of the independent additional axis board (NOBOT), select “2: Add Nobot axis.” To delete an axis, select “3: Delete Nobot axis.”
If 3 is selected, the following screen appears. To return to the above screen, press the Enter key.

```
Nobot Axis? Was Deleted
Press ENTER to Continue.
```

(The “?” on the screen will be replaced by the number of the axis just deleted.)

For the subsequent settings, use the values mentioned in the specifications of the mechanical unit of the robot.

6 From the screen, select the size of the motor used for an axis of the independent additional axis board (NOBOT).

```
**** Nobot Axis 1 Initialization ****
     33.ACb0.5      38.ACa12      43.ACa100
     34.ACa1        39.ACa22      44.ACa150
       :              :              :
       :              :              :
      0. Next page.
Select Motor size?
```

7 Select a motor type from the screen.

```
MOTOR TYPE
                1. /2000        6.F/3000
                2. /3000        7.F/2500
                3. S/2000       8.L/3000
                   :               :
Select Motor Type?
```

8 From the screen, select the maximum current control value of the motor (maximum permissible current value of the amplifier).

```
CURRENT LIMIT FOR AMPLIFIER

  1. 2A       6. 60A
  2. 4A       7. 80A
  3. 12A      8. 100A
   :            :
Select Amplifier Current Limit?
```

9 Set the amplifier number.

```
— Amplifier number Setting –
Enter Amplifier Number (1→16)?
```

10 Set the amplifier type.

```
— Amplifier Type Setting –
Amplifier ? Type = #
Enter (1: Change, 2: No Change)?
```

* The “?” on the screen will be replaced by the amplifier number set previously.
* If 0 is displayed in place of “#,” this indicates that no amplifier has been set.
  To change the amplifier type, select “1: Change.” The following screen appears:

```
SELECT AMP TYPE
     1. A06B–6100 series 6 axes amplifier
     2. A06B–6093 Beta series (FSSB)
```

Select an amplifier type.

11 Select the axis type of the independent additional axis board (NOBOT).

```
Axis Type Setting –
1: Linear Axis
2: Rotary Axis
Select Axis Type?
```

Linear Axis: Linear axis
Rotary Axis: Rotary axis

12 Enter the gear reduction ratio. For a linear axis, enter the distance of travel along the axis due to one rotation of the motor (in mm). For a rotary axis, enter the number of revolutions of the motor required for one rotation about the output axis.

```
— Gear Ratio Setting —
Enter Gear Ratio ?
```

13 Set the maximum axis speed. A suggested value is calculated with the maximum motor speed and the gear ratio and displayed on the screen. To change the value, enter 1 and enter a new value. To use the suggested value, enter 2.

```
— Maximum Speed Setting —
Suggested  Speed = 150.000 ( mm / sec )
 ( Calculated with Max Motor Speed)
Enter ( 1: Change, 2: No Change ) ?
```

To change the maximum speed, use the following screen:

```
Enter Max Speed (mm/sec) ?
```

Enter the maximum speed.

14 Set the axis direction in relation to the motor. If the direction of rotation about the axis due to the forward rotation of the motor is plus, enter 1 for TRUE; if minus, enter 2 for FALSE.

```
MOTOR DIRERCTION
Motion Sign = TRUE
Enter ( 1: TRUE, 2: FALSE) ?
```

15 Enter the upper limit (UPPER LIMITS) of the axis operation range in mm for a linear axis and in deg. for a rotary axis.

```
UPPER LIMITS
Enter Upper Limit ( deg ) ?
```

Note) The user must decide on the upper limit. The upper limit and the lower limit, to be entered next, must satisfy the following condition:

<Upper limit> – <Lower limit> = Axis length

For example, if the axis length is 100 mm, the following limits may be entered:

<Upper limit> = 50 mm
<Lower limit> = –50 mm

16 Enter the lower limit (LOWER LIMITS) of the axis operation range in mm for a linear axis and in deg. for a rotary axis.

```
LOWER LIMITS
Enter Lower Limit ( deg ) ?
```

17 Enter the mastering position.

```
MASTER POSITION
Enter Master Position ( deg ) ?
```

18 Set the acceleration/deceleration time constant.
To change the first acceleration/deceleration time constant for each axis, enter 1 and then a new value. To use the suggested value, enter 2.

```
ACC/DEC TIME
Default Value of acc_time1=256(ms)
Enter ( 1: Change, 2: No Change ) ?
```

To change the second acceleration/deceleration time constant for each axis, enter 1 and then a new value. To use the suggested value, enter 2.

```
Default Value of acc_time2=128(ms)
Enter ( 1: Change, 2: No Change ) ?
```

19 Set the minimum acceleration/deceleration time. The acceleration and other instructions will use this value if the actual acceleration/deceleration time is below the time specified here. To change the time, enter 1 and then a new value. To use the suggested value, enter 2.

```
MIN_ACCEL TIME
Default Value of min_accel time =384(ms)
Enter ( 1: Change, 2: No Change ) ?
```

20 Set the ratio of the motor phase conversion total load inertia to the inertia (rotor inertia ratio). The inertia ratio must be larger than 1 and less than 5. To set no ratio, enter 0.

```
Load Ratio is
   Load Inertia ( Kg*cm*s*2*)
   --------------------------
   Motor Inertia (Kg*cm*s*2*)
Enter Load ratio ? ( 0:None 1→5: Valid)
```

21 Set the brake number: Enter the number of the brake used for the axis, in the range of 0 to 4.

```
BRAKE  SETTING
Enter  Brake  Number (0→4)
```

22 Specify whether to enable or disable brake control. To enable it, select “1: Enable” and then the brake control delay time. To disable it, select “2: Disable.”

```
SERVO  TIMEOUT
  Servo  Off   is Enable
  Enter (1: Enable  2: Disable) ?
Select?
```

(If 1: Enable is selected)

```
Enter  Servo  Off  Time ?  (0.0→30.0  Sec)
```

23 The system returns to the screen in step 6.

```
**** Group  ?  Total  Nobot  Axis = #
   1.Display/Modify Nobot axis 1→6
   2.Add Nobot axis
   3.Delete Nobot axis
   4.Exit
Select item?
```

- To display/change the settings of the independent additional axis board (NOBOT), select 1. Display/Modify Nobot axis.
- To set up another axis of the independent additional axis board (NOBOT), select 2. and repeat the procedure starting at step 7.
- To delete an axis of the independent additional axis board (NOBOT), select 3. Delete Nobot axis.
- To exit from the screen, select 4. Exit, then 0. EXIT.

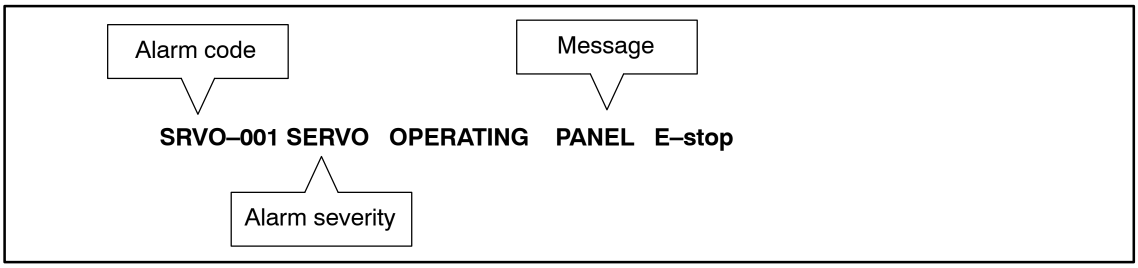
This is the end of the procedure.

C. Alarm codes

This part of this manual describes alarm codes, alarm severity, causes, and actions.

☐ Contents of this appendix

C.1 Description of an Alarm Code Table



Alarm code	Alarm ID: Alarm type	
	Alarm number	
Alarm message: Description of the alarm		

Alarm severity	Program	Robot	Power to the servo system	Range
NONE	Does not stop.	Does not stop.	Not turned off	———
WARN				———
PAUSE.L	Halts.	Decelerates and stops.		Local
PAUSE.G				Global
STOP.L				Local
STOP.G				Global
SERVO		Stops immediately.	Turned off	Global
ABORT.L	Terminates forcibly.	Decelerates and stops.	Not turned off	Local
ABORT.G				Global
SERVO 2		Stops immediately.	Turned off	Global
SYSTEM				Global

Range	Range in which the alarm is applied when multiple programs run simultaneously (multitask function)	
	Local	The alarm is applied only to a program which has caused the alarm.
	Global	The alarm is applied to all programs.

Alarm

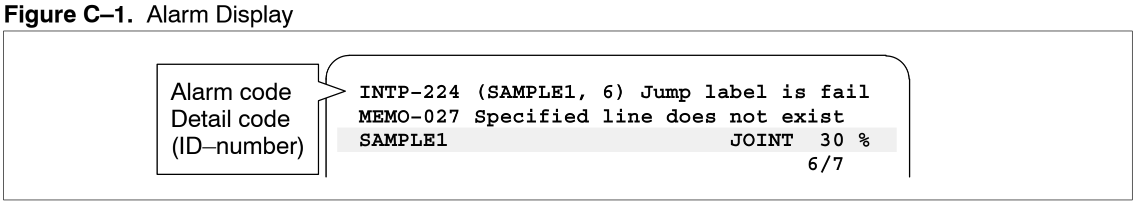
An alarm is issued when a failure occurs while the program is taught or played back or when the emergency stop signal or another alarm signal is input from a peripheral unit. The alarm is issued to notify the operator of the failure so the operator can halt processing for safety's sake.

NOTE If an alarm whose number is not described herein occurs, contact FANUC Robotics.

Alarm codes display or indication

When an alarm is issued, the alarm LED lights on the teach pendant and the alarm message is displayed in the first and second line of the screen. The operator can find out which alarm has been issued by looking at the LED and message.

Figure C–1. Alarm Display



Alarm severity

How to operate the program or the robot until the program or the robot stops depends on the seriousness of the cause of the alarm. This “seriousness” is called alarm severity. The degree of alarm severity is indicated as follows:

Table C–1. Alarm severity

Alarm severity	Description
WARN alarm	A WARN alarm warns the operator of a comparatively minor or unimportant failure. The WARN alarm does not affect the operation of the robot. When the WARN alarm occurs, no corresponding LED on the teach pendant or the machine operator’s panel lights. To prevent a possible failure, appropriate action should be taken.
PAUSE alarm	When a PAUSE alarm occurs, the execution of the program is halted, and the operation of the robot is stopped after the operation in progress is completed. Appropriate action must be taken for the alarm before the program is restarted.
STOP alarm	When a STOP alarm occurs, the execution of the program is halted, and the robot is decelerated until it is stopped. Appropriate action must be taken for the alarm before the program is restarted.
SERVO alarm	When a SERVO alarm occurs, the power to the servo system is turned off to halt the execution of the program and to stop the robot immediately. A SERVO alarm is issued for safety’s sake or when a failure occurs during robot operation.
ABORT alarm	When an ABORT alarm occurs, the execution of the program is forcibly terminated, and the robot is decelerated until it is stopped.
SERVO 2 alarm	When a SERVO 2 alarm occurs, the power to the servo system is turned off to forcibly terminate the program and to stop the robot immediately. A SERVO alarm is issued for safety’s sake or when a failure occurs during robot operation.
SYSTEM alarm	A SYSTEM alarm is issued when a major system failure occurs. When the SYSTEM alarm is issued, every robot in the system is disabled. After taking appropriate action for the alarm, turn the power off, the turn it on again.

Active alarm screen

The active alarm screen displays only active alarm(s). Once the alarms have been eliminated by alarm clear signal input, the active alarm screen reads ”THERE ARE NO ACTIVE ALARMS.”

The screen displays the alarm(s) output after the last alarm clear signal input. When the delete key (+ shift) is pressed on the alarm history screen, the corresponding alarm is cleared from the active alarm screen.

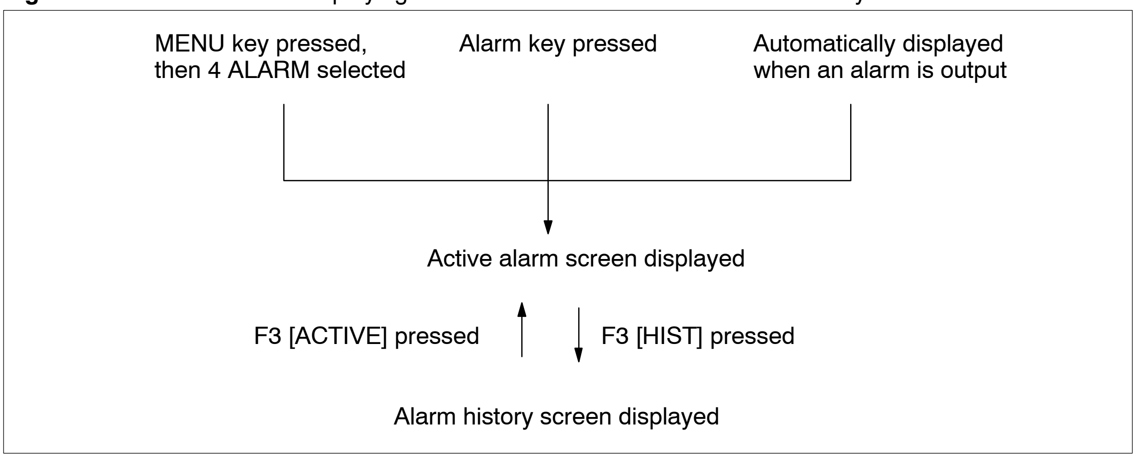
The screen shows alarms having a severity level of PAUSE or higher. WARN or NONE alarms or resets are not displayed. Some PAUSE alarms or severer alarms may not be displayed if system variables such as $ER_NOHIS are set accordingly.

If multiple alarms are detected, the screen displays the alarms in the opposite order to that in which they were detected.

Up to 100 lines can be displayed.

If an alarm has a cause code, the cause code is displayed below the alarm display line.

Figure C–2. Procedure for Displaying the Active Alarm Screen and Alarm History Screen



Automatic alarm display function

When an alarm which will cause the system to stop (PAUSE or severer alarm) is detected, the automatic alarm screen display function automatically displays the alarm screen. This function frees the operator of having to display the alarm screen and enables the direct cause of the system failure to be found quickly.

NOTE Once the display requirements are satisfied, the alarm screen is automatically displayed even if an alarm is detected at start–up. The automatic alarm display is performed irrespective of the start mode.

NOTE If an alarm is detected when a CRT is connected, the alarm screen is displayed on both the teach pendant and the CRT.

The requirements for automatic alarm screen display are as described below:

- When the flag of the automatic alarm screen display function is set
  On the system setting screen, select AUTO.DISPLAY OF ALARM MENU to enable or disable the automatic display function. The function is disabled by default. For this change to take effect, the power must be turned off and then on again. –> See Section 3.14.
- When the Auto. display of alarm menu flag for an alarm severity level is set
  $ER_SEV_NOAUTO[] sets whether automatic alarm screen display is enabled or disabled for each alarm severity level. There are seven levels of alarm severity. NONE and WARN alarms will not affect program execution or robot operation and will not instigate the automatic display. Automatic display is enabled for PAUSE and severer alarms by default. The setting can be changed on the system variable screen.

System variable	Corresponding alarm severity level	Standard setting
$ER_SEV_NOAUTO [1]	PAUSE	TRUE
$ER_SEV_NOAUTO [2]	STOP	TRUE
$ER_SEV_NOAUTO [3]	SERVO	TRUE
$ER_SEV_NOAUTO [4]	ABORT	TRUE
$ER_SEV_NOAUTO [5]	SYSTEM	TRUE

FALSE: Automatic alarm screen display is disabled.

TRUE: Automatic alarm screen display is enabled.

NOTE If a PAUSE alarm is detected, followed by an ABORT alarm, when the automatic display of a PAUSE alarm is disabled, automatic display is not performed during fault output.

- Automatic display of a particular alarm can be disabled.

The automatic alarm screen display function can be disabled for a particular alarm. Up to ten such alarms can be set on the system variable screen. If a specified alarm is detected while automatic alarm screen display is enabled, the alarm screen will not be automatically displayed.

This setting is made with the following system variables:

System variable	Description
$ER_NOAUTO. $NOAUTO_ENB	Enables or disables the function to suppress automatic alarm screen display for an alarm specified in $ER_NOAUTO.NOAUTO_CODE[] while the automatic alarm screen display is enabled. ● FALSE: The function to suppress automatic alarm screen display is disabled. ● TRUE: The function to suppress automatic alarm screen display is enabled.
$ER_NOAUTO. $NOAUTO_NUM	Sets the number of alarms set in $ER_NOAUTO.NOAUTO_CODE[].
$ER_NAOUTO. $NOAUTO_CODE [1 to 10]	Error number consisting of alarm ID and alarm number Example 11 002 (Servo 002 alarm) Alarm ID Alarm number Alarm ID → See Section C.2.

For the alarms listed below, which are caused by a user operation and which cause the system to stop, $ER_NOAUTO.$NOAUTO_ENB is set to TRUE by default. When the setting is changed to FALSE, the corresponding alarm screen is automatically displayed.

Servo 001	Operator's panel emergency stop
Servo 002	Teach pendant emergency stop
Servo 003	Deadman's switch
Servo 004	Fence open
Servo 005	Teach pendant released (mount/unmount button of the teach pendant pressed)
Servo 012	Power restoration

When an alarm set in $ER_NOAUTO.$NOAUTO_CODE[] and another alarm are detected in that order while $ER_NOAUTO.$NOAUTO_ENB is set to TRUE, automatic display is not performed.

- When fault output is in progress
  If the alarm screen is automatically displayed each time a PAUSE or severer alarm is detected, the alarm screen may be displayed while alarm recovery or setting check is being performed on another screen. The screens will be frequently switched, which can interfere with recovery and other operations. To prevent this from occurring, automatic display is not performed while an alarm is active.

  Whether there is an active alarm can be checked by the fault signal output.

  While the fault signal is output irrespective of the servo start–up, automatic display is not performed even if an alarm is detected.

NOTE When a PAUSE, STOP, or ABORT alarm is detected, the fault signal is output with the servo system started. Each time an alarm clear signal is input, the fault signal is reset. If continuous monitoring is performed to raise an alarm (NO ARC PROCESS I/O BOARD, for instance), automatic alarm display may be performed at each reset input.

NOTE When a SERVO or SYSTEM alarm is detected, the fault signal is reset after the servo system starts.

Automatic return function

The automatic return function displays the screen which was displayed until automatic screen display when an alarm clear signal is input. This function is used together with the automatic display function. The automatic return function operates as described below:

- When the automatic alarm screen display function is enabled, the alarm screen is automatically displayed if an alarm is raised. When the alarm is eliminated by the input of an alarm clear signal, the previous screen is automatically displayed.
- If the alarm screen is not displayed automatically because of an alarm but displayed by means of menu selection, the previous screen is not displayed even if an alarm clear signal is input.
- If another screen is displayed before an alarm clear signal is input, the automatic return function does not operate.
- The automatic return function operates when the fault signal output is turned off.
- If the power is turned off or on after the alarm screen is displayed by the automatic display function, the automatic return function does not work after the start–up. This is not affected by the start mode (cold start, hot start, etc.).

Procedure C–1 Displaying the alarm occurrence alarm history/alarm detail information

Step

1 Press the MENUS key to display the screen menu.

2 Select "4 ALARM".
The alarm occurrence screen is displayed.
If an alarm is detected, the active alarm screen is automatically displayed.

```
 3 MANUAL FCTNS
 4 ALARM
 5 I/O
```

MENUS

```
INTP-224 (SAMPLE1, 6) Jump label failed

Alarm:ACTIVE                   JOINT  30 %
                                       1/1



MEMO-027 Specified line. does not exist




[ TYPE ]                  HIST
```

Alarm cause code

3 To display the alarm history screen, press F3 [HIST]. When F3 [ACTIVE] is pressed, the active alarm screen appears again.

```
 3
 4 ALARM
 5 I/O
```

MENUS

```
INTP-224 (SAMPLE1, 7) Jump label is fail
MEMO-027 Specified line does not exist
Alarm                          JOINT  30 %
                                     1/25
  1 INTP-224 (SAMPLE1, 7) Jump label is
  2 SRVO-002 Teach pendant E-stop
  3 R E S E T
  4 SRVO-027 Robot not mastered(Group:1)
  5 SYST-026 System normal power up



[ TYPE ]                  CLEAR   HELP
```

NOTE The latest alarm is assigned number 1. To view messages that are currently not on the screen, press the F5, HELP, then press the right arrow key.

4 To display the alarm detail screen, press the F5 "HELP" key.

```
CLEAR     HELP
```

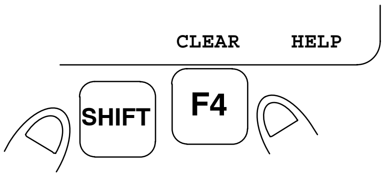
F5

```
INTP-224 (SAMPLE1, 7) Jump label is fail

INTP-224 (SAMPLE1, 7) Jump label is fail
MEMO-027 Specified line does not exist
30-MAY-44 07:15
STOP.L                        00000110
Alarm
                                     1/25
  1 INTP-224 (SAMPLE1, 7) Jump label is
  2 SRVO-002 Teach pendant E-stop


[ TYPE ]                  CLEAR   HELP
```

5 To return to the alarm history screen, press the PREV key.

PREV

6 To delete all the alarm histories, press and hold down the SHIFT key, then press the F4 ”CLEAR” key.

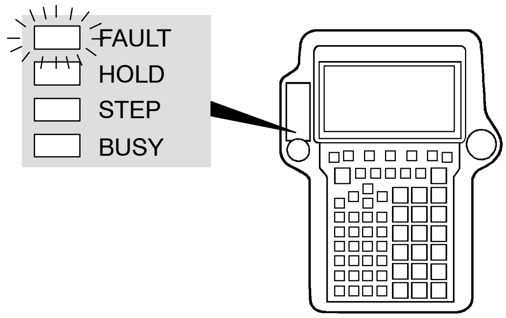


NOTE When system variable $ER_NOHIS = 1, NONE alarms or WARN alarms are not recorded. When $ER_NOHIS=2, resets are not recorded in the alarm history. When $ER_NOHIS=3, resets, WARN alarms, and NONE alarms are not recorded.

Procedure C–2 Halt caused by a major alarms and recoveries

Halt caused by a major alarm

Step 1 When an alarm is issued, the running program is halted, and PAUSED or END is displayed on the screen of the teach pendant.
An alarm message is also displayed on the screen of the teach pendant and the ALARM lamp lights.



Recovery from a deadman switch alarm (SERVO–003)

Step 1 Press and hold down the deadman switch, then press the RESET key.

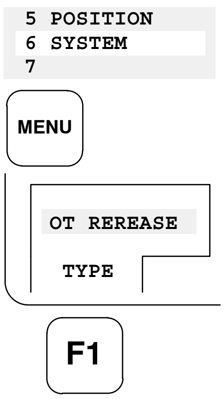
```
SRVO-003 Deadman switch released
SAMPLE1         LINE 2
SAMPLE1                          JOINT 30%
```

Eliminating an overtravel alarm (servo 005)

Step 1 Press the MENU key to display the screen menu.

```
SRVO-005 Robot Overtrabel
SAMPLE1         LINE 2           PAUSE
SAMPLE1                          JOINT 30%
```

2 Press 0 NEXT PAGE, then select 6 SYSTEM on the next page. Press F1 [TYPE], then select OT RELEASE. The OT Release screen appears.




```
System OT Release

AXIS      OTMINUS      OTPLUS
  1          --          OT
  2          --          --



[ TYPE ]  RELEASE
```

3 Press F2 [RELEASE] to release the overtravel axis.
4 While holding down the shift key, press the alarm clear button.
5 While holding down the shift key, press the jog key to move the tool along the overtravel axis into the movable range.

Recovery from a broken wrist alarm (SERVO–006)

Step 1 Press and hold down the SHIFT key, then press the RESET key.

```
SRVO-006 Hand broken
SAMPLE1          LINE 2
SAMPLE1                          JOINT 30%
```

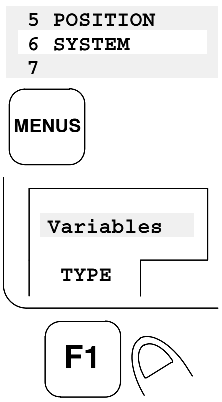
2 While pressing the SHIFT key, press the appropriate jog key to move the robot to a position where it can be repaired.

Recovery from a pulse mismatch alarm, a BZAL alarm and a RCAL alarm (SRVO–038, 062, 063)

Step 1 Press the MENUS key to display the screen menu.

```
SRVO-038 Pulse mismatch
SAMPLE1              LINE 2           PAUSED
SAMPLE1                          JOINT  30 %
```

2 Press ”0 — NEXT —” and then select ”6 SYSTEM” on the next page. Press F1 ”[TYPE]” and then select ”Variables”. A system variable screen is displayed.



3 Set TRUE to $MCR.$SPC_RESET. (This system variable is automatically set to FALSE soon again.)

4 Press RESET key to release a alarm.

NOTE The mastering data may be correct even if a pulse count mismatch alarm is detected. If the mastering data is correct, mastering need not be performed. Merely set $DMR_GRP.$MASTER_DONE to true, then select 6 MASTER/CAL on the positioning screen.

Recovery from other alarms

Step 1 Remove the cause of an alarm. For example, correct the program.

2 Press the RESET key to reset the alarm. Then, the alarm message disappears on the screen of the teach pendant. The ALARM LED goes off.

C.2 Alarm Codes

SRVO Error Codes (ID = 11)

SRVO–001 SERVO Operator panel E–stop

Cause: The operator panel emergency stop push button is pressed.
Remedy: Twist the operator panel emergency stop push button clockwise to release and press RESET.

SRVO–002 SERVO Teach pendant E–stop

Cause: The emergency stop button on the teach pendant was pressed.
Remedy: Release the emergency stop button on the teach pendant.

SRVO–003 SERVO Deadman switch released

Cause: The deadman's switch was not pressed when the teach pendant was enabled.
Remedy: Press the deadman's switch to enable operation of the robot.

SRVO–004 SERVO Fence open

Cause: On the terminal block on the printed circuit board of the operator's panel, no connection is established between the FENCE1 and FENCE2 signals. When a safety door is connected, it is open.
Remedy: Establish a connection between FENCE1 and FENCE2, then press the reset key. When a safety door is connected, close the door before starting work.

SRVO–005 SERVO Robot overtravel

Cause: A hardware limit switch on an axis was tripped. Usually, the movement of the robot is prevented from exceeding a limit beyond the maximum range of movement (software limits) for each axis. However, when the robot is shipped, the overtravel state is set for transit.
Remedy:
1 Check the fuse (F4) in the power supply unit. Replace the fuse if it has blown.
2 Release the overtravel axis by using the overtravel release screen [SYSTEM OT RELEASE].
3 While holding down the shift key, press the alarm release button to release the alarm.
4 Move the overtravel axis to within the movable range by holding down the shift key and performing jog feed.
5 For the model using the B cabinet, check the fuse (F2) on the printed circuit board for emergency stop control. Replace the fuse if it has blown.
6 Replace the printed circuit board for emergency stop control.

SRVO–006 SERVO Hand broken

Cause: A safety hand has broken. If no broken hand can be found, however, the most likely cause is the HBK signal of a robot connection cable being at the 0 V level.
Remedy:
1 Check the fuse (F4) in the power supply unit. Replace the fuse if it has blown.
2 While holding down the shift key, press the alarm release button to clear the alarm.
3 While holding down the shift key, position the tool to the workplace by performing jog feed.
  a Replace the safety hand.
  b Check the cable.

SRVO–007 SERVO External emergency stops

Cause: The external emergency stop push button is pressed.
On the terminal block of the printed circuit board for emergency stop control, no connection is established between EMGIN1 and EMGIN2.
Remedy: If using external emergency stop, clear source of fault and press RESET.
If not using external emergency stop, check wiring at EMGIN1, EMGIN2.

SRVO–008 SERVO Brake fuse blown

Cause: The brake fuse is blown on the EMG Control pcb.
Remedy: Replace the fuse. Check the LED (FALM) on the printed circuit board for emergency stop control to determine whether the fuse has blown.

SRVO–009 SERVO Pneumatic pressure alarm

Cause: The pneumatic pressure alarm indicates the presence of a defect. If the pneumatic pressure alarm is not detected, however, the most likely cause is the PPABN signal of a robot connection cable being at the 0 V level.
Remedy: If the pneumatic pressure alarm is not detected, check the cable.

SRVO–010 SERVO Belt broken

Cause: The belt broken robot digital input (RDI7) is asserted.

Remedy:
1. If the belt is found to be defective in any way, repair it and then press the reset key.
2. When the belt is found to be normal, signal RDI [7] in the robot connection cable may be abnormal. Check the cable.
3. Check system variable $PARAM_GROUP.$BELT_ENABLE.

SRVO–011 SERVO TP released while enabled

Cause: The teach pendant attachment switch on the operator's panel was operated while the teach pendant was enabled.

Remedy: Reconnect the teach pendant cable to continue operation.

SRVO–012 SERVO Power failure recovery

Cause: Normal power on (hot start).

Remedy: This is just a notification. You do not have to do anything for this warning message.

SRVO–013 SYSTEM Srvo module config changed

Cause: Upon power–up with power restoration enabled (hot start), the configuration of the DSP modules on the axis control printed circuit board and the multi–function printed circuit board has been changed.

Remedy: Turn on the power in cold start mode.

SRVO–014 WARN Fan motor abnormal

Cause: A fan motor in the control unit is abnormal.

Remedy: Check the fan motors and fan motor connection cables. Replace any faulty fan motor(s) and/or cable(s).

SRVO–015 SERVO System over heat

Cause: The temperature of the control unit is higher than the specified value.

Remedy:
1. If the ambient temperature is higher than the specified temperature (45øC), provide ventilation to reduce the ambient temperature to the specified value.
2. Check that the fans are operating normally. If not, check the fan motors and fan motor connection cables. Replace any faulty fan motor(s) and/or cable(s).
3. If the thermostat on the backplane printed circuit board is faulty, replace the backplane unit.

SRVO–016 SERVO Cooling water volume drop

Cause: Cooling water volume dropped.

Remedy: Consult our service representative.

SRVO–017 SERVO No robot internal mirror

Cause: No robot internal mirror.

Remedy: Consult our service representative.

SRVO–018 SERVO Brake abnormal

Cause: The current for brake exceeded the specification.

Remedy:
1. For the S–800 or S–900 robot, check the fuse (F1) on the printed circuit board for emergency stop control.
2. Check the brake cable.
3. Replace the amplifier.
4. Check the 100–VAC input voltage. If a voltage of 90 VAC or less is detected, check the input power supply voltage.

SRVO–021 SERVO SRDY off (Group:%d Axis:%d)

Cause: When HRDY is on, SRDY is off even though no other alarm cause is present. (HRDY is the signal sent from the host to the servo system to specify whether to turn the servo amplifier's MCC on or off. SRDY is the signal sent from the servo system to the host to indicate whether the servo amplifier's MCC is on or off. Generally, if a servo amplifier's MCC is not turned on despite the signal for turning the MCC on having been issued, an alarm is issued for the servo amplifier. The host does not issue this alarm (SRDY off) if an alarm for the servo amplifier is detected. So, this alarm indicates that the MCC is not turned on when no error can be found.)

Remedy:
1 Check whether the door is open. Also check the door switch.
2 Check the 200 VAC voltage applied to the servo amplifier. If the voltage is found to be 170 VAC or lower, check the input power supply voltage.
3 Replace the emergency stop control printed board.
4 Replace the main CPU printed circuit board.
5 Check the following cables. Replace them if necessary.
6 Replace the servo amplifier.

SRVO–022 SERVO SRDY on (Group:%d Axis:%d)

Cause: SRDY was already on when an attempt was made to turn on the MCC with HRDY. (HRDY is the signal sent from the host to the servo system to specify whether to turn the servo amplifier's MCC on or off. SRDY is the signal sent from the servo system to the host to indicate whether a servo amplifier's MCC is on or off.)

Remedy:
1 Replace the emergency stop control printed board.
2 Replace the main CPU printed circuit board.
3 Check the cable linking the servo amplifier and main CPU printed circuit board. If any abnormality is found, replace the cable.
4 Replace the servo amplifier.

SRVO–023 SERVO Stop error excess(Group:%d Axis:%d)

Cause: An excessive servo positional error occurred when the motor stopped.

Remedy:
1 Check whether the applied load exceeds the rating. If so, reduce the applied load. (If an excessive load is applied, the torque required for acceleration, deceleration, and so forth exceeds the maximum available torque of the motor. Therefore it may prove impossible to correctly respond to an issued command, resulting in the output of this alarm.)
2 Check each interphase voltage of the three–phase voltage (200 VAC) applied to the servo amplifier. If the voltage is found to be 170 VAC or below, check the input power supply voltage. (A sub–standard voltage, applied to a servo amplifier results in a lower–than–normal torque. Therefore, it may prove impossible to correctly respond to an issued command, thus resulting in the output of this alarm.)
3 If the input power supply voltage is found to be 170 VAC or higher, replace the servo amplifier.
4 Replace the motor.

SRVO–024 SERVO Move error excess(Group:%d Axis:%d)

Cause: When the robot moved, the servo positional error exceeded a previously specified value ($PARAM_GROU.$MOVER_OFFST or $PARAM_GROUP.$TRKERRLIM). For example, this error will occur if the feedrate of the robot differs from that specified.

Remedy: Perform the same action as that described for the previous item.

SRVO–025 SERVO Motn dt overflow (Group:%d Axis:%d)

Cause: The value entered with a command is too large.

Remedy: Perform a cold start :
1 Turn off the robot.
2 On the teach pendant, press and hold the SHIFT and RESET keys.
3 While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error.

SRVO–026 WARN Motor speed limit(Group:%d Axis:%d)

Cause: An attempt was made to exceed the maximum rated motor speed ($PARAM_GROUP.$MOT_SPD_LIM). The motor speed is clamped to its maximum rated value.

Remedy: This is just a notification. However, you should attempt to eliminate this error and not repeat the circumstances that led up to it.

SRVO–027 WARN Robot not mastered(Group:%d)

Cause: An attempt was made to perform calibration, but mastering has not yet been completed.

Remedy: Perform mastering from the calibration screen [6 SYSTEM CALIBRATION].

SRVO–030 SERVO Brake on hold (Group:%d)

Cause: When the temporary stop alarm function ($SCR.$BRKHOLD_ENB=1) is enabled, this alarm is issued whenever a temporary stop is made. When this function is not to be used, disable the function.

Remedy: Disable [TEMPORARY STOP/SERVO OFF] on the general item setting screen [6 GENERAL SETTING ITEMS].

SRVO–031 SERVO User servo alarm (Group:%d)

Cause: A user servo alarm was issued.This alarm is raised when system variable $MCR_GRP[i].$SOFT_ALARM is set to TRUE. Only KAREL users can use this variable, however.

Remedy: This is just a notification. You do not have to do anything for this warning message.

SRVO–033 WARN Robot not calibrated(Group:%d)

Cause: An attempt was made to set a reference point for simple mastering, but calibration has not yet been completed.

Remedy: Perform calibration by following the procedure below.

1 Turn on the power.
2 Execute [CALIBRATION] from the calibration screen [6 SYSTEM CALIBRATION].

SRVO–034 WARN Ref pos not set (Group:%d)

Cause: An attempt was made to perform simple mastering, but a required reference point has not yet been set.

Remedy: Set a reference point for simple mastering from the calibration screen.

SRVO–035 WARN Joint speed limit(Group:%d Axis:%d)

Cause: An attempt was made to exceed the maximum joint speed ($PARAM_GROUP.$JNTVELLIM). The joint speed is clamped to its maximum rated value.

Remedy: Every attempt should be made to eliminate this error.

SRVO–036 SERVO Inpos time over (Group:%d Axis:%d)

Cause: The in–position monitor time ($PARAM_GROUP.$INPOS_TIME) has elapsed, but the in–position state ($PARAM_GROUP.$STOPTOL) has not yet been set.

Remedy: Perform the same action as that specified for Servo – 023 (stop error excess).

SRVO–037 SERVO IMSTP input (Group:%d)

Cause: The *IMSTP signal, which is a peripheral device I/O signal, is applied.

Remedy: Turn on the *IMSTP signal.

SRVO–038 SERVO2 Pulse mismatch (Group:%d Axis:%d)

Cause: A pulse count detected at power–off differs from that detected at power–on.

Remedy: Contact our service center serving your locality.

SRVO–040 WARN Mastered at mark pos(Group:%d)

Cause: Zero position master is done with mark position (not with zero position).

Remedy: This is not an alarm.

SRVO–041 SERVO2 MOFAL alarm (Group:%d Axis:%d)

Cause: A value specified with a command is too large.

Remedy: Document the events that led to the error and contact our service center serving your locality.

SRVO–043 SERVO DCAL alarm(Group:%d Axis:%d)

Cause: The energy produced by the regenerative discharge is excessive. As a result, all the generated energy cannot be dissipated as heat. (When a robot is to be operated, a servo amplifier feeds energy to the robot. Along its vertical axis, however, the robot moves downword using potential energy. If the decrease in the potential energy exceeds the acceleration energy, the servo amplifier receives energy from the motor. This also occurs during deceleration even if the force of gravity has no effect. This energy is called regenerative energy. Normally, the servo amplifier dissipates this regenerative energy by converting it to heat. When the amount of regenerative energy exceeds the amount of energy that can be dissipated as heat, excess energy accumulates in the servo amplifier, thus triggering this alarm.)

Remedy: When the LED indicator of the servo amplifier PSM displays ”8” (DCOH alarm) (The DCOH alarm is issued when the thermostat detects overheating of the regenerative resistor.):

a This alarm may be raised when acceleration/deceleration is frequently performed or when a large amount of regenerative energy is generated in the vertical axis. In such cases, the robot should be used under less demanding conditions.

b Replace the regenerative resistor.

c Check the cable between the servo amplifier (CN8A) and the regenerative resistor. Replace it if necessary.

d Replace the servo amplifier.

SRVO–044 SERVO HVAL alarm(Group:%d Axis:%d)

Cause: The DC voltage (DC link voltage) of the main circuit power supply is abnormally high. The LED indicator of the servo amplifier PSM displays ”7.”

Remedy:

1 Check the three–phase input voltage applied to the servo amplifier. When the voltage is 253 VAC or higher, check the input power supply voltage. (If the motor is abruptly accelerated or decelerated while the three–phase input voltage exceeds 253 VAC, this alarm may be issued.)

2 Check whether the applied load is within the rated value. If the rated load is exceeded, reduce the applied load. (If a load exceeds the rated value, built–up regenerative energy may cause this alarm to be issued even when the three–phase input voltage satisfies the specifications.)

3 Check the cables (CN3 and CN4) in the amplifier. Replace them if necessary.

4 Check the cable between the main CPU printed circuit board (JRV1) and the printed circuit board for emergency stop control (JRV1).

5 Replace the servo amplifier.

SRVO–045 SERVO HCAL alarm(Group:%d Axis:%d)

Cause: An excessively high current flowed through the main circuit of a servo amplifier. The LED indicator on the servo amplifier PSM displays ”–.” One of the red LEDs (HC1 to HC6) above the 7–segment LED is lit, indicating the axis for which the HCAL alarm is detected.

Remedy:

1 Disconnect the motor power line from the terminal block of the servo amplifier, then turn on the power. If this alarm is still issued, replace the servo amplifier.

2 Remove the motor power line from the terminal block of the servo amplifier, then check the insulation between U, V, and W of the motor power line and GND. If a short circuit is found, check the motor, robot connection cable, or robot internal cable. If any abnormality is found, replace the faulty hardware.

3 Remove the motor power line from the terminal block of the servo amplifier, then check the resistance between U and V, V and W, and W and U of the motor power line using a measuring instrument capable of detecting very low resistances. If the measured resistances differ from each other, check the motor, robot connection cable, or robot internal cable. If any abnormality is found, replace the faulty hardware.

4 Replace the main CPU printed circuit board.

SRVO–046 SERVO2 OVC alarm (Group:%d Axis:%d)

Cause: This alarm is issued to protect the motor when there is a danger of thermal destruction when the root–mean–square current value, calculated internally by the servo system, exceeds the maximum permissible value.

Remedy:

1 Check the operating conditions of the robot. If the robot’s ratings, such as the rated duty cycle and load, are exceeded, modify the use of the robot such that the rated values are not exceeded.

2 Check each interphase voltage of the three–phase voltage (200 VAC) applied to the servo amplifier. If the applied voltage is found to be 170 VAC or less, check the input power supply voltage.

3 Replace the main CPU printed circuit board.

4 Replace the servo amplifier.

5 Replace the motor.

SRVO–047 SERVO LVAL alarm(Group:%d Axis:%d)

Cause: Despite the external magnetic contactor for a servo amplifier being on, the DC voltage (DC link voltage) of the main circuit power supply or the control power supply voltage (+5 V) is excessively low.

Remedy: 1 When the LED indicator on servo amplifier displays ”6” (This alarm is issued when the control power supply voltage (+5 V) is excessively low.):
a Check each interphase voltage of the three–phase voltage (200 VAC) applied to the servo amplifier. If the applied voltage is found to be 170 VAC or less, check the input power supply voltage.
b Replace the servo amplifier.

2 When the LED indicator on the servo amplifier displays ”4” (This alarm is issued when the DC voltage (DC link voltage) of the main circuit power supply is excessively low.):
a Check each interphase voltage of the three–phase voltage (200 VAC) applied to the servo amplifier. If the applied voltage is found to be 170 VAC or less, check the input power supply voltage. Check the servo amplifier's circuit breaker. Close the circuit breaker if it is found to be off.
b Replace the servo amplifier.

SRVO–049 SERVO OHAL1 alarm (Group:%d Axis:%d)

Cause: A servo amplifier's built–in thermostat was actuated. The LED indicator on the servo amplifier PSM displays”3.”

Remedy: 1 Check the operating conditions of the robot. If any of the ratings specified for the robot, such as its rated duty cycle or load, are exceeded, modify the use of the robot such that the ratings are not exceeded.

2 Check whether the fuse (F1) in the servo amplifier has blown.

3 Check the cable between the servo amplifier (CN8B) and the transformer. Replace it if necessary.

4 Check the cable (CN4) in the servo amplifier. Replace it if necessary.

5 Replace the servo amplifier.

SRVO–050 SERVO CLALM alarm (Group:%d Axis:%d)

Cause: An excessively large disturbance torque is estimated by the servo software. (A collision was detected.)

Remedy: 1 Check whether the robot has collided with an object. If so, reset the system, then move the robot away from the location of the collision by using jog feed.

2 Check that the applied load does not exceed the maximum rating. If the rated load is exceeded, reduce the applied load. (If the robot is used with an excessive load applied, the estimated disturbance may become excessively large, resulting in this alarm being output.)

3 Check each interphase voltage of the three–phase voltage (200 VAC) applied to the servo amplifier. If the applied voltage is found to be 170 VAC or less, check the input power supply voltage.

4 Replace the servo amplifier.

SRVO–051 SERVO2 CUER alarm(Group:%d Axis:%d)

Cause: The offset of a current feedback value is excessively large.

Remedy: 1 Replace the main CPU printed circuit board.

2 Replace the servo amplifier.

SRVO–053 WARN Disturbance excess(Group:%d Axis:%d)

Cause: Disturbance estimated in the software exceed the threshold value. There is a possibility that the load held in the wrist exceed the robot specification.

Remedy: If operation is allowed to continue, a detection error may result. On the status screen containing the disturbance value, specify a new value for the acceptable disturbance limit.

SRVO–054 SYSTEM DSM memory error (DSM:%d)

Cause: The DSP module program memory is defective.

Remedy: Replace the DSP module.

SRVO–055 SERVO2 FSSB com error 1 (Group:%d Axis:%d)

Cause: FSSB communication error from SRVO to SLAVE occured

Remedy: Check FSSB hardware connection.

SRVO–056 SERVO2 FSSB com error 2 (Group:%d Axis:%d)

Cause: FSSB communication error from SLAVE to SRVO occured

Remedy: Check FSSB hardware connection.

SRVO–057 SERVO2 FSSB disconnect (Group:%d Axis:%d)

Cause: FSS'B for communication from SLAVE to SRVO is disconnected.

Remedy: Check FSSB hardware connection.

SRVO–058 SYSTEM FSSB init error (N:%d)

Cause: FSSB communication error occured during initialization

Remedy: Check FSSB hardware connection.

SRVO–059 SYSTEM Servo amp init error

Cause: Servo amplifier initializing is failed.

Remedy: Check the servo amplifier and its wiring.
Refer to the maintenance manual.

SRVO–061 SERVO2 CKAL alarm(Group:%d Axis:%d)

Cause: The clock for the rotation counter in the pulse coder is abnormal.

Remedy: If this alarm occurs along with a SRVO–068 DTERR, SRVO–069 CRCERR, or SRVO–070 STBERR, disregard this alarm and refer to the other three alarm remedies. Replace the pulse coder or motor and master the robot.

SRVO–062 SERVO2 BZAL alarm(Group:%d Axis:%d)

Cause: This alarm is issued when the battery for backing up the absolute position data of the pulse coder is not connected. The battery cable inside the robot may have become disconnected.

Remedy: Correct the cause of the alarm, then turn on the power again after setting the system variable ($MCR.$SPC_RESET) to true. Mastering is required.

SRVO–063 SERVO2 RCAL alarm(Group:%d Axis:%d)

Cause: The built-in rotation counter on the pulse coder is abnormal.

Remedy:
1. Eliminate the cause of the alarm. Set system variable $MCR.$SPC_RESET to TRUE, and turn the power off and then on again. Mastering must be performed.
2. Replace the pulse coder. Mastering must be performed.

NOTE The RCAL alarm may be displayed when any of the "SERVO–068 DTERR," "SERVO–069 CRCERR," or "SERVO–070 STBERR" alarms is raised. In this case, however, the RCAL alarm is not actually raised.

SRVO–064 SERVO2 PHAL alarm(Group:%d Axis:%d)

Cause: This alarm is issued when the phase of a pulse signal generated by the pulse coder is abnormal.

Remedy: Replace the pulse coder. After replacing, perform mastering.

NOTE If the DTERR, CRCERR, or STBERR alarm is issued, this alarm may also be output at the same time. Should this occur, however, this alarm can be safely ignored.

SRVO–065 WARN BLAL alarm(Group:%d Axis:%d)

Cause: The battery voltage for the pulse coder has dropped below the allowable minimum.

Remedy: Replace the battery. (When this alarm is issued, immediately replace the battery while the system power is turned on. If the BZAL alarm is issued because the battery is not replaced in time, position data will be lost, thus necessitating robot mastering.)

SRVO–066 SERVO2 CSAL alarm(Group:%d Axis:%d)

Cause: The pulse coder ROM checksum data are abnormal.

Remedy: If this alarm occurs along with a SRVO–068 DTERR, SRVO–069 CRCERR, or SRVO–070 STBERR, disregard this alarm and refer to the other three alarm remedies. Replace the pulse coder or motor and master the robot.

SRVO–067 SERVO2 OHAL2 alarm (Group:%d Axis:%d)

Cause: The temperature inside the pulse coder has become too high, causing the built–in thermostat to actuate.

Remedy:
1. Check the operating conditions of the robot. If any of the rating specified for the robot, such as its rated duty cycle or load, are exceeded, modify the use of the robot such that the ratings are not exceeded.
2. If this alarm is issued, even when the power is turned on and the motor has not overheated, replace the motor.

SRVO–068 SERVO2 DTERR alarm (Group:%d Axis:%d)

Cause: A request signal was sent to the serial pulse coder, but no serial data was returned.

Remedy:
1. Check the cable between the main CPU printed circuit board (JRF2) and the printed circuit board for emergency stop control (JRF2). Replace it if necessary.
2. For a model with the robot connection cable, check the cable and replace it if necessary. Then, check connector P1 on the connector panel of the mechanical unit.
3. Replace the serial pulse coder.

SRVO–069 SERVO2 CRCERR alarm (Group:%d Axis:%d)

Cause: Serial data changed during transfer.

SRVO–070 SERVO2 STBERR alarm (Group:%d Axis:%d)

Cause: A serial data start bit or stop bit error occurred.

Remedy:
1. Check that the shields of the robot connection cable (for the pulse coder signal) and peripheral device cables are securely connected to a ground plate.
2. Check that each unit is securely grounded.
3. Replace the printed circuit board for emergency stop control.
4. Replace the cable between the printed circuit board for emergency stop control and the main CPU.
5. Replace the main CPU printed circuit board.
6. Replace the pulse coder, after which mastering must be performed.
7. Replace the robot connection cable (for the pulse coder signal).

SRVO–071 SERVO2 SPHAL alarm (Group:%d Axis:%d)

Cause: The feedback speed is abnormally high (3750 rpm or greater).

Remedy:
1. This alarm does not indicate the main cause of the problem if issued together with the PHAL alarm (alarm No. 064).
2. Check whether the load applied to the robot exceeds the maximum rating. If the rated load is exceeded, reduce the applied load. After replacing perform mastering.
3. Replace the pulse coder of the motor.

SRVO–072 SERVO2 PMAL alarm(Group:%d Axis:%d)

Cause: The pulse coder may be faulty.

Remedy: Replace the pulse coder, then perform mastering.

SRVO–073 SERVO2 CMAL alarm(Group:%d Axis:%d)

Cause: The pulse coder may be faulty, or noise may be causing the pulse coder to malfunction.

Remedy: Perform simple mastering and improve the shielding.

SRVO–074 SERVO2 LDAL alarm(Group:%d Axis:%d)

Cause: The LED on the pulse coder has become disconnected.

Remedy: Replace the pulse coder, then perform mastering.

SRVO–075 WARN Pulse not established(Group:%d Axis:%d)

Cause: The absolute position of the pulse coder has not yet been established.

Remedy: Using job feed, move the robot along each axis for which this alarm is issued, until the alarm is not re–issued after being cleared.

SRVO–076 Tip Stick Detection (Group:%d Axis:%d)

Cause: The servo software has detected an excessive disturbance torque at the beginning of operation.

Remedy: Press the reset key on the teach pendant to cause a reset, and separate the robot from all obstacles by jogging.
If neither a deposition nor a collision has occurred, it is likely that the load on the robot is heavier than the rating.
Check the input voltage of the servo amplifier. Each phase–to–phase voltage must be higher than 170 VAC.
Check the U–V, V–W, and U–W voltages. They must be the same voltage (210 VAC or below).
Refer to the maintenance manual.

SRVO–081 WARN EROFL alarm (Track enc:%d)

Cause: The line tracking pulse count overflowed.

Remedy: Contact our service center serving your locality.

SRVO–082 WARN DAL alarm(Track enc:%d)

Cause: Line tracking pulse coder disconnected.

Remedy:
1 Check the corresponding line tracking connection to the axis control printed circuit board.
2 Check the pulse coder cable.
3 Replace the SIF and DSM modules on the axis control printed circuit board.
4 Replace the pulse coder.

SRVO–083 WARN CKAL alarm (Track enc:%d)

Cause: The clock for the rotation counter in the line tracking pulse coder is abnormal.

Remedy: Refer to SRVO–061.

SRVO–084 WARN BZAL alarm (Track enc:%d)

Cause: This alarm is issued when the battery for backing up the absolute position data for the pulse coder is not connected.

Remedy: See the description for Servo – 062 BZAL alarm.

SRVO–085 WARN RCAL alarm (Track enc:%d)

Cause: The built-in rotation counter on the line tracking pulse coder is abnormal.

Remedy: Refer to SRVO–063.

SRVO–086 WARN PHAL alarm (Track enc:%d)

Cause: This alarm is issued when the phase of a pulse signal generated by the pulse coder is abnormal.

Remedy: See the description for Servo – 064 PHAL alarm.

SRVO–087 WARN BLAL alarm (Track enc:%d)

Cause: This alarm is issued when the battery voltage for backing up the absolute position data of the pulse coder has dropped.

Remedy: See the description for Servo – 065 BLAL alarm.

SRVO–088 WARN CSAL alarm (Track enc:%d)

Cause: The line tracking pulse coder ROM checksum data are abnormal.

Remedy: Refer to SRVO–066.

SRVO–089 WARN OHAL2 alarm (Track enc:%d)

Cause: The motor has overheated.

Remedy: See the description for Servo – 067 OHAL2 alarm.

SRVO–090 WARN DTERR alarm (Track enc:%d)

Cause: An error occurred during communication between the pulse coder and main CPU printed circuit board.

Remedy: See the description for Servo – 068 DTERR alarm.

SRVO–091 WARN CRCERR alarm (Track enc:%d)

Cause: An error occurred during communication between the pulse coder and main CPU printed circuit board.

Remedy: See the description for Servo – 069 CRCERR alarm.

SRVO–092 WARN STBERR alarm (Track enc:%d)

Cause: An error occurred during communication between the pulse coder and main CPU printed circuit board.

Remedy: See the description for Servo – 070 STBERR alarm.

SRVO–093 WARN SPHAL alarm (Track enc:%d)

Cause: This alarm is issued when the position data sent from the pulse coder is considerably greater than the previous data.

Remedy: See the description for Servo – 071 SPHAL alarm.

SRVO–094 WARN PMAL alarm (Track enc:%d)

Cause: The pulse coder may be faulty.
Remedy: See the description for Servo– 072 PMAL alarm.

SRVO–095 WARN CMAL alarm (Track enc:%d)

Cause: The pulse coder may be faulty. Or, noise may have caused the pulse coder to malfunction.
Remedy: See the description for Servo– 073 CMAL alarm.

SRVO–096 WARN LDAL alarm (Track enc:%d)

Cause: The LED on the pulse coder has become disconnected.
Remedy: See the description for Servo – 074 LDAL alarm.

SRVO–097 WARN Pulse not established(Enc:%d)

Cause: The absolute position of the pulse coder has not yet been established.
Remedy: See the description for Servo – 075 Pulse not established.

SRVO–101 SERVO Robot overtravel(Robot:%d)

Cause: A Robot overtravel limit switch is pressed.
Remedy: Refer to SRVO–005.

SRVO–102 SERVO Hand broken (Robot:%d)

Cause: The hand broken (*HBK) robot input is asserted.
Remedy: Refer to SRVO–006.

SRVO–103 SERVO Air pressure alarm(Rbt:%d)

Cause: The pneumatic pressure (PPABN) robot input is asserted.
Remedy: Refer to SRVO–009.

SRVO–105 SERVO Door open or E.Stop

Cause: Controller door is opened
Or the E.stop signals are detected for a short time,
Or a mis–wiring of hardware connection
Remedy: Close controller door
And press RESET
If reset is not effective, correct hardware connection

SRVO–106 SERVO Door open/E.Stop (Robot:%d)

Cause: The controller door was opened.
An emergency stop signal was detected temporarily.
A hardware disconnection occurred.
Remedy: Close the controller door, and press the reset key.
If no reset occurs, repair the hardware wiring.
Refer to the maintenance manual.

SRVO–108 Press RESET to enable robot

Cause: When the enable/disable switch is set to ”Enable,” it is necessary to cause a reset.
Remedy: To enable the robot, press the reset key.

SRVO–111 SERVO Softfloat time out(Group:%d)

Cause: Follow-up time is over when softfloat is OFF.
Remedy: Make $SFLT_FUPTIM larger.

SRVO–112 PAUSE.G Softfloat time out(Group:%d)

Cause: Follow-up time is over when softfloat is OFF.
Remedy: Make $SFLT_FUPTIM larger.

SRVO–121 SERVO Excessive acc/dec time(Group:%d)

Cause: Acceleration time is much longer.
Remedy: Contact our service center serving your locality.

SRVO–122 SERVO Bad last ang(internal)(Group:%d)

Cause: Last angle update request does not match current angle.
Remedy: Contact our service center serving your locality.

SRVO–122 Bad last ang(internal)(Group:%d)

Cause: Last angle update request does not match current angle
Motion speed is too high to perform quick stop.
Remedy: Report to hotline
Reduce the motion speed.

SRVO–126 SERVO Quick stop error (Group:%d)

Cause: The program was over in process of a quick stop.
Remedy: Press reset

SRVO–130 SERVO OHAL1(PSM) alarm (Group:%d Axis:%d)

Cause: The servo amplifier(PSM) overheated.
Remedy: Lower the duty cycle of operation.

SRVO–131 SERVO LVAL(PSM) alarm(Group:%d Axis:%d)

Cause: The DC voltage on the main power circuit of the servo amplifier is lower than the specification even though MCC is on.
Remedy: Refer to the maintenance manual.

SRVO–132 SERVO HCAL(PSM) alarm(Group:%d Axis:%d)

Cause: The current in the main power circuit of the servo amplifier exceeded specification.
Remedy: Refer to the maintenance manual.

SRVO–133 SERVO FSAL(PSM) alarm (Group:%d Axis:%d)

Cause: Cooling fan for Control circuit stops.
Remedy: Refer to the maintenance manual.

SRVO–134 SERVO DCLVAL(PSM) alarm (Group:%d Axis:%d)

Cause: Back–up charge circuit for amplifier have trouble.
Remedy: Check the cables and connections between amplifier(CN1) and MCC.
Check the fuse(F1,F3) in transformer.
If using B–cabinet
Replace the EMG Control printed circuit board.
Replace the amplifier.

SRVO–135 SERVO FSAL alarm (Group:%d Axis:%d)

Cause: Cooling fan for Control circuit stops.
Remedy: Check or Replace the fan.

SRVO–136 SERVO DCLVAL alarm (Group:%d Axis:%d)

Cause: Back–up charge circuit for amplifier have trouble.
Remedy: Check the cables and connections between amplifier(CN1) and MCC.
Check the fuse(F1,F3) in transformer.
If using B–cabinet
Replace the EMG Control printed circuit board.
Replace the amplifier.

SRVO–138 SERVO SDAL alarm (Group:%d Axis:%d)

Cause: A pulse coder signal error was detected by software.
The probable cause is noise or a failure (disconnection) in the pulse coder internal circuit.
Remedy: If a normal operation can be resumed by turning the controller power off and on again, it is likely that the cause is noise.
Enhance the shielding of the pulse coder cable.
If a normal operation cannot be resumed by turning the controller power off and on again, the pulse coder is defective.
Replace it, and perform mastering.

SRVO–148 HCAL(CNV) alarm (Group:%d Axis:%d)

Cause: The current of the main power supply circuit of the servo amplifier has exceeded the rating.

Remedy: Remove the motor power lines from the servo amplifier, and then turn the power on.
If HCAL still occurs, replace the servo amplifier and transistor module.
Measure the resistance between the grounding wire and each of the U, V, and W lines at the cable terminals.
If there is a short–circuit, check whether the cable or motor is defective.
Check the U–V, V–W, and U–W resistances with a measuring instrument that can detect a low resistance at the end of a cable.
If the same resistance is observed, replace the servo amplifier.
If a different resistance is observed, check whether the cable or motor is defective.
If the problem is not yet solved, replace the axis control SIF module for the axis of interest.
Refer to the maintenance manual.

SRVO–151 FSAL(INV) alarm (Group:%d Axis:%d)

Cause: The cooling fan for the control circuit has stopped.

Remedy: Check the fan, or remove it.
Refer to the maintenance manual.

SRVO–156 SERVO IPMAL alarm (Group:%d Axis:%d)

Cause: IPM module has trouble.

Remedy: Replace the IPM mudule. Refer to the maintenance manual for detail.

SRVO–157 SERVO CHGAL alarm (Group:%d Axis:%d)

Cause: Charge of the main circuit could not finish within specified time.

Remedy: DC link may short–circuit. Check the connections.
Electric resistance to restrict charge current may be defective.
Replace the wiring board.
Refer to the maintenance manual for deteil.

SRVO–160 SERVO Panel/External E–stop

Cause: The emergency stop button on the operator's panel was pressed, or the external emergency stop function was activated. (EMGIN1 and EMGINC are not strapped to each other. Or, EMGIN2 and EMGINC are not strapped to each other.)

Remedy: Release the emergency stop button. If the external emergency stop function has been activated, remove the cause. If no cause can be found, and no jumper is installed between EMGIN1 and EMGINC or between EMGIN2 and EMGINC on the terminal block of the emergency stop control printed circuit board, but cables are connected to the terminals, check the cables.

SRVO–171 WARN MotorSpd lim/DVC(Group:%d Axis:%d)

Cause: The actual motor speed was about to exceed the maximum allowable speed ($PARAM_GROUP[i].$MOT_SPD_LIM[j]). It was clamped at the maximum allowable speed.

Remedy: This is just a notification. You do not have to do anything for this warning message.

SRVO–172 WARN MotorSpd lim/DVC0(Group:%d Axis:%d)

Cause: Motor can not rotate as fast as the calculated speed required for the current motion.

Remedy: This is just a notification. You do not have to do anything for this warning message.

SRVO–173 WARN MotorSpd lim/DVC1(Group:%d Axis:%d)

Cause: Motor can not rotate as fast as the calculated speed required for the current motion.

Remedy: This is just a notification. You do not have to do anything for this warning message.

SRVO–174 WARN MotorAcc lim/DVC(Group:%d Axis:%d)

Cause: Motor can not accelerate as much as the calculated acceleration required to for the current motion

Remedy: This is just a notification. You do not have to do anything for this warning message.

SRVO–176 SERVO CJ/Illegal Mode %d,%d

Cause: Wrong CJ mode was used.

Remedy: Internal motion error.
Contact service immideately.

SRVO–177 WARN CJ error %d,%d,%d,%d

Cause: Wrong CJ mode was used.
Remedy: Contact service representative.

SRVO–178 SYSTEM CJ error %d,%d,%d,%d

Cause: Wrong CJ mode was used.
Remedy: Contact service representative.

SRVO–179 Motor torque limit (Group:%d Axis:%d)

Cause: The torque of the axis has exceeded the limit.
Remedy: This is a warning message.
No action is needed for this warning message.

SRVO–181 SERVO Mcmd input while estimating(Group:%d)

Cause: Robot was going to move while identifying the payload.
Remedy: Press RESET. Be careful not to move robot while identifying the payload.

SRVO–182 Needed init. has not been done

Cause: This is an error internal to the system.
A system variable or internal work memory has not been initialized normally.
Remedy: Turn the power off and on again.

1 Turn the power off.
2 Turn the power on.

If the alarm is still issued, take a note of what caused this alarm, and then contact the service personnel.

SRVO–183 ROBOT isn't ready

Cause: Servo activation is off.
Remedy: Remove the factor that turned servo activation off, and press the reset button.

SRVO–184 Other task is processing

Cause: The data area that this instruction tried to use had been locked by another task.
Remedy: Execute the instruction after the task that uses the data area ends.

SRVO–185 Data is for other group

Cause: The data the instruction tries to use belongs to another group.
Remedy: Collect the data of the desired group before executing the instruction.

SRVO–186 Needed Data has not been got

Cause: No data has been collected, or any collected data does not belong to the desired mode.
Remedy: Collect the necessary data before executing the instruction.

SRVO–187 Need specfing Mass

Cause: Estimating the load information of this type requires specifying the mass of the load.
Remedy: Specify the mass of the load before estimating load information.

SRVO–191 Illegal Joint Speed (Group:%d Axis:%d)

Cause: The motion command exceeded specification.
Remedy: Internal motion error.
Contact service immideately.

SRVO–194 Servo disconnect

Cause: Servo is disconnected.
Remedy: Connect servo, and then press reset.
Refer to the maintenance manual for detail.

SRVO–199 PAUSE Control Stop

Cause: Control Stop is detected.
Remedy: After this alarm, Fence open or SVON input alarm is detected.
See the remedy of the next alarm.

SRVO–201 SERVO Panel E–stop or SVEMG abnormal

Cause: The emergency stop button on the operator's panel was pressed, or the wiring of the SVEMG signal is incorrect.

Remedy: Release the emergency stop button on the operator's panel, and press the reset button.
If it is impossible to release, it is likely that the wiring of the SVEMG is incorrect. Check the wiring.
Refer to the maintenance manual for details.

SRVO–202 SERVO TP E–stop or SVEMG abnormal

Cause: The emergency stop button on the teach pendant was pressed, or the wiring of the SVEMG signal is incorrect.

Remedy: Release the emergency stop button on the teach pendant, and press the reset key.
If it is impossible to release, it is likely that the wiring of the SVEMG is incorrect. Check the wiring.
Refer to the maintenance manual for details.

SRVO–204 SYSTEM External(SVEMG abnormal) E–stop

Cause: An external emergency stop signal was input while the wiring of the SVEMG was abnormal.

Remedy: After turning the power off, correct the wiring of the SVEMG, and remove the cause of the external emergency stop, and then turn the power on again. Refer to the maintenance manual for details.

SRVO–205 SYSTEM Fence open(SVEMG abnormal)

Cause: The fence was opened while the wiring of the SVEMG was abnormal.

Remedy: After turning the power off, correct the wiring of the SVEMG, and close the fence, then turn the power on again. Refer to the maintenance manual for details.

SRVO–206 SYSTEM Deadman switch (SVEMG abnormal)

Cause: The deadman switch was released while the wiring of the SVEMG was abnormal.

Remedy: After turning the power off, correct the wiring of the SVEMG, and grip the deadman switch, then turn the power on again. Refer to the maintenance manual for details.

SRVO–207 SERVO TP switch abnormal or Door open

Cause: While the teach pendant was enabled, the deadman switch was gripped, and the fence was open, the controller door was opened, or the wiring of the SVEMG was incorrect.

Remedy: Close the controller door, and press the reset button. If the door was not open, the wiring of the SVEMG is incorrect. Correct it.
Refer to the maintenance manual for details.

SRVO–209 SERVO Robot–2 SVEMG abnormal

Cause: A disconnection of the SVEMG signal for robot 2 was detected.

Remedy: Turn the power off.
Rewire the SVEMG of the controller for robot 2.
Close the fence circuit, and press the reset button.
Refer to the maintenance manual.

SRVO–210 SERVO EX_robot SVEMG abnormal

Cause: A disconnection of the SVEMG signal for an additional robot (a third robot, such as a positioner or additional axis) was detected.

Remedy: Turn the power off.
Rewire the SVEMG of the controller for the additional robot.
Close the fence circuit, and press the reset button.
Refer to the maintenance manual.

SRVO–211 SERVO TP OFF in T1, T2

Cause: The teach pendant was disabled while the mode switch was set in the T1 or T2 position and robot 1 and 2 were disconnected.
Alternatively, there is a broken wire in the hardware.

Remedy: Set the teach pendant enable/disable switch to on, and press the reset key.
If a reset does not take effect, repair the hardware wiring.
Refer to the maintenance manual.

SRVO–213 SERVO Fuse blown (PanelPCB)

Cause: The fuse on the panel PBC board has blown.
Remedy: Replace the fuse on the panel PBC.
Refer to the maintenance manual.

SRVO–214 SERVO Fuse blown (Amp)

Cause: The fuse in the six–axis amplifier has blown.
Remedy: Replace the fuse in the six–axis amplifier.
Refer to the maintenance manual.

SRVO–215 SERVO Fuse blown (Aux axis)

Cause: The fuse for additional axis control in the six–axis amplifier has blown.
Remedy: Replace the fuse for additional axis control in the six–axis amplifier.
Refer to the maintenance manual.

SRVO–216 SERVO OVC (total) (%d)

Cause: The current flowing through the robot cable has exceeded its limit.
Remedy: Modify the program in such a way that the operating condition can be relaxed.

SRVO–221 SERVO Lack of DSP (Group:%d Axis:%d)

Cause: The DSP (servo control CPU) for this axis was not found, though it was specified in the system variable $AXISORDER.
Remedy: Check that the number of DSPs on the DSP board is sufficient for the quantity specified in $SCR_GRP[].$AXISORDER[]. Replace the DSP board with one having sufficient DSPs if necessary.
Alternatively, change the setting of $AXISORDER.

SRVO–222 SERVO Lack of Amp (Amp:%d)

Cause: FSSB indicates that there is no amplifier module.
Remedy: Check that the fiber cable is connected to the amplifier correctly.
Replace the fiber cable leading to the amplifier.
Check that the amplifier power is normal.
Check that $AXISORDER and $AMP_NUM are specified correctly.
Refer to the maintenance manual.

SRVO–230 SERVO Chain 1 (+24v) abnormal

Cause: A failure occurred in chain 1 (+24 V).
Remedy: If the failure occurred because of the deadman switch being released, grip it again.
Repair the chain 1 (+24 V) circuit in the hardware.
On the system setting screen, set whether to reset the chain failure to ”Yes.”
Press the reset key on the teach pendant.
Refer to the maintenance manual.

SRVO–231 SERVO Chain 2 (0v) abnormal

Cause: A failure occurred in chain 2 (0 V).
Remedy: If the failure occurred because of the deadman switch being released, grip it again.
Repair the chain 2 (0 V) circuit in the hardware.
On the system setting screen, set whether to reset the chain failure to ”Yes.”
Press the reset key on the teach pendant.
Refer to the maintenance manual.

SRVO–232 SERVO NTED input

Cause: The NTED (non–teacher enabling device) was released.
Remedy: Press the NTED (non–teacher enabling device), and then press the reset key.

SRVO–233 SERVO TP OFF in T1, T2/Door open

Cause: The mode switch is set in the T1 or T2 position, and the teach pendant is disabled.
Alternatively, the controller door is open.
Still alternatively, there is a disconnection in the hardware.
Remedy: After setting the teach pendant enable/disable switch to on, close the controller door, and press the reset key.
If a reset does not take effect, repair the hardware wiring.
Refer to the maintenance manual.

SRVO–234 WARN Deadman switch released

Cause: The deadman switch on the teach pendant was released.
Remedy: This is a warning message.

SRVO–235 SERVO Short term Chain abnormal

Cause: A temporary chain failure was detected.
Remedy: If this failure occurred simultaneously with the "deadman switch released" alarm, release the deadman switch, and press it again.
If this failure occurs simultaneously with any other safety–related error, cause the same error to occur again, and press the reset key.

SRVO–236 WARN Chain failure is repaired

Cause: A chain failure was removed.
Remedy: When the system checked for the chain failure again, the chain failure had been removed. Press the reset key.

SRVO–237 WARN Cannot reset chain failure

Cause: An attempt to reset the chain failure failed.
Remedy: Repair the chain 1 (+24 V) circuit in the hardware.
Press the emergency stop button on the teach pendant, and rotate it clockwise to release. Then, press the reset key.
Refer to the maintenance manual.

SRVO–240 SERVO Chain 1 (FENCE) abnormal

Cause: When the fence circuit was opened, a chain 1 (+24 V) failure occurred.
Remedy: Repair the chain 1 (+24 V) circuit in the fence hardware.
On the system setting screen, set whether to reset the chain failure to "Yes."
Press the reset key on the teach pendant.
Refer to the maintenance manual.

SRVO–241 SERVO Chain 2 (FENCE) abnormal

Cause: When the fence circuit was opened, a chain 2 (0 V) failure occurred.
Remedy: Repair the chain 2 (0 V) circuit in the fence hardware.
On the system setting screen, set whether to reset the chain failure to "Yes."
Press the reset key on the teach pendant.
Refer to the maintenance manual.

SRVO–242 SERVO Chain 1 (EXEMG) abnormal

Cause: When an external emergency stop signal was input, a chain 1 (+24 V) failure occurred.
Remedy: Repair the chain 1 (+24 V) circuit in the external emergency stop hardware.
On the system setting screen, set whether to reset the chain failure to "Yes."
Press the reset key on the teach pendant.
Refer to the maintenance manual.

SRVO–243 SERVO Chain 2 (EXEMG) abnormal

Cause: When an external emergency stop signal was input, a chain 2 (0 V) failure occurred.
Remedy: Repair the chain 2 (0 V) circuit in the fence hardware.
On the system setting screen, set whether to reset the chain failure to "Yes."
Press the reset key on the teach pendant.
Refer to the maintenance manual.

SRVO–244 SERVO Chain 1 abnormal (Rbt:%d)

Cause: A chain 1 (+24 V) failure occurred.
Remedy: Repair the chain 1 (+24 V) circuit in the hardware.
On the system setting screen, set whether to reset the chain failure to "Yes."
Press the reset key on the teach pendant.
Refer to the maintenance manual.

SRVO–245 SERVO Chain 2 abnormal (Rbt:%d)

Cause: A chain 2 (0 V) failure occurred.

Remedy: Repair the chain 2 (0 V) circuit in the hardware.
On the system setting screen, set whether to reset the chain failure to ”Yes.”
Press the reset key on the teach pendant.
Refer to the maintenance manual.

SRVO–246 SERVO Chain 1 abnormal (EX_robot)

Cause: A chain 1 (+24 V) failure occurred in an additional robot (a third robot, such as a positioner or additional axis).

Remedy: Repair the chain 1 (+24 V) circuit in the hardware.
On the system setting screen, set whether to reset the chain failure to ”Yes.”
Press the reset key on the teach pendant.
Refer to the maintenance manual.

SRVO–247 SERVO Chain 2 abnormal (EX_robot)

Cause: A chain 2 (0 V) failure occurred in an additional robot (a third robot, such as a positioner or additional axis).

Remedy: Repair the chain 2 (0 V) circuit in the hardware.
On the system setting screen, set whether to reset the chain failure to ”Yes.”
Press the reset key on the teach pendant.
Refer to the maintenance manual.

SRVO–250 SERVO SVEMG/MAINON1 abnormal

Cause: When the SVEMG became on, the MAINON1 signal remained off.
This is an emergency stop circuit failure.

Remedy: Repair the emergency stop circuit hardware.
Turn the power off and on again.

SRVO–260 SERVO Chain 1 (NTED) abnormal

Cause: A chain 1 (+24 V) failure occurred when the NTED (non–teacher enabling device) was released.

Remedy: Repair the chain 1 (+24 V) circuit in the NTED (non–teacher enabling device) hardware.
On the system setting screen, set whether to reset the chain failure to ”Yes.”
Press the reset key on the teach pendant.
Refer to the maintenance manual.

SRVO–261 SERVO Chain 2 (NTED) abnormal

Cause: A chain 2 (0 V) failure occurred when the NTED (non–teacher enabling device) was released.

Remedy: Repair the chain 2 (0 V) circuit in the NTED (non–teacher enabling device) hardware.
On the system setting screen, set whether to reset the chain failure to ”Yes.”
Press the reset key on the teach pendant.
Refer to the maintenance manual.

SRVO–262 SERVO Chain 1 (SVDISC) abnormal

Cause: When the servo power supply off signal was input, a chain 1 (+24 V) failure occurred.

Remedy: Repair the chain 1 (+24 V) circuit for the servo power supply off signal circuit.
On the system setting screen, set whether to reset the chain failure to ”Yes.”
Press the reset key on the teach pendant.
Refer to the maintenance manual.

SRVO–263 SERVO Chain 2 (SVDISC) abnormal

Cause: When the servo disconnect signal was input, a chain 2 (0 V) failure occurred.

Remedy: On the system setting screen, set whether to reset the chain failure to ”Yes.”
Press the reset key on the teach pendant.
Refer to the maintenance manual.

SRVO–264 SYSTEM E.STOP circuit abnormal 1

Cause: A deposition occurred in the emergency stop unit.

Remedy: Repair the MON3 circuit in the emergency stop unit.
Refer to the maintenance manual.

SRVO–265 SERVO E.STOP circuit abnormal 2

Cause: When the servo was activated, the MON3 was already on.
The MON3 is abnormal.

Remedy: Repair the MON3 circuit in the emergency stop unit.
Refer to the maintenance manual.

SRVO–266 SERVO FENCE1 status abnormal

Cause: When the fence signal was input, the FENCE1 remained on.

Remedy: Repair the FENCE1 circuit.
Refer to the maintenance manual.

SRVO–267 SERVO FENCE2 status abnormal

Cause: When the fence signal was input, the FENCE2 remained on.

Remedy: Repair the FENCE2 circuit.
Refer to the maintenance manual.

SRVO–268 SERVO SVOFF1 status abnormal

Cause: When the SVOFF signal was input, the SVOFF1 remained on.

Remedy: Repair the SVOFF1 circuit.
Refer to the maintenance manual.

SRVO–269 SERVO SVOFF2 status abnormal

Cause: When the SVOFF signal was input, the SVOFF2 remained on.

Remedy: Repair the SVOFF2 circuit.
Refer to the maintenance manual.

SRVO–270 SERVO EXEMG1 status abnormal

Cause: When an external emergency stop signal was input, the EXEMG1 remained on.

Remedy: Repair the EXEMG1 circuit.
Refer to the maintenance manual.

SRVO–271 SERVO EXEMG2 status abnormal

Cause: When an external emergency stop signal was input, the EXEMG2 remained on.

Remedy: Repair the EXEMG2 circuit.
Refer to the maintenance manual.

SRVO–272 SERVO SVDISC1 status abnormal

Cause: When the servo power off signal was input, the SVDISC1 remained on.

Remedy: Repair the SVDISC1 circuit.
Refer to the maintenance manual.

SRVO–273 SERVO SVDISC2 status abnormal

Cause: When the servo power off signal was input, the SVDISC2 remained on.

Remedy: Repair the SVDISC2 circuit.
Refer to the maintenance manual.

SRVO–274 SERVO NTED1 status abnormal

Cause: When the NTED signal was input, the NTED1 remained on.

Remedy: Repair the NTED1 circuit.
Refer to the maintenance manual.

SRVO–275 SERVO NTED2 status abnormal

Cause: When the NTED signal was input, the NTED2 remained on.

Remedy: Repair the NTED2 circuit.
Refer to the maintenance manual.

SRVO–276 SERVO Disable on T2 mode

Cause: The robot cannot operate in the T2 mode.

Remedy: Set the mode switch to the T1 or auto position.

SRVO–277 SYSTEM Panel E–stop (SVEMG abnormal)

Cause: When the emergency stop button on the operator's panel was pressed, the SVEMG signal was not input.

Remedy: The wiring of the SVEMG is incorrect. Correct it, and turn the power on again.

SRVO–278 SYSTEM TP E–stop (SVEMG abnormal)

Cause: When the emergency stop button on the teach pendant was pressed, the SVEMG signal was not input.

Remedy: The wiring of the SVEMG is incorrect. Correct it, and turn the power on again.

SRVO–280 SERVO SVOFF input

Cause: The SVOFF (servo off signal) was input.

Remedy: Find out what caused the SVOFF to be input, and remove the cause.

SRVO–281 SYSTEM SVOFF input (SVEMG abnormal)

Cause: The SVOFF input circuit was detected, and a disconnection of the SVEMG was detected.

Remedy: Turn the power off.
Repair the wiring of the SVEMG.
Close the SVOFF input circuit, and press the reset key.
Refer to the maintenance manual.

SRVO–282 SERVO Chain 1 (SVOFF) abnormal

Cause: When the SVOFF (servo off signal) was input, a chain 1 (+24 V) failure occurred.

Remedy: Repair the chain 1 (+24 V) circuit in the SVOFF hardware.
On the system setting screen, set whether to reset the chain failure to "Yes."
Press the reset key on the teach pendant.
Refer to the maintenance manual.

SRVO–283 SERVO Chain 2 (SVOFF) abnormal

Cause: When the SVOFF (servo off signal) was input, a chain 2 (0 V) failure occurred.

Remedy: Repair the chain 2 (0 V) circuit in the SVOFF hardware.
On the system setting screen, set whether to reset the chain failure to "Yes."
Press the reset key on the teach pendant.
Refer to the maintenance manual.

SRVO–290 SERVO DClink HC alarm (Group:%d Axis:%d)

Cause: An abnormal current flowed through the amplifier DC link circuit.

Remedy: It is likely that there is a short–circuit in a motor power line or the motor coil.
Refer to the maintenance manual.

SRVO–291 SERVO IPM over heat (Group:%d Axis:%d)

Cause: It was detected that the IPM element in the amplifier had overheated.

Remedy: Decrease the duty cycle of operation.
If this symptom occurs frequently, replace the amplifier.

SRVO–292 SERVO EXT.FAN alarm (Group:%d Axis:%d)

Cause: A fan for cooling the amplifier heat–release fins is defective.

Remedy: Replace the cooling fan.
Refer to the maintenance manual.

SRVO–293 SERVO DClink (PSM) HCAL (Group:%d Axis:%d)

Cause: An abnormal current flowed through the DC link circuit of the PSM amplifier.

Remedy: It is likely that there is a short–circuit in a motor power line or the motor coil.
Refer to the maintenance manual.

SRVO–294 SERVO EXT.FAN (PSM) alarm (Group:%d Axis:%d)

Cause: A fan for cooling the PSM amplifier heat–release fins is defective.

Remedy: Replace the cooling fan.
Refer to the maintenance manual.

SRVO–295 SERVO

Cause: A communication error occurred between the PSM and SVM.

Remedy: Replace the cable between the PSM and SVM. Alternatively, replace the SVM or PSM. Refer to the maintenance manual for details.

SRVO–296 SERVO

Cause: The power regenerated in the PSM is too much.

Remedy: Check whether a fan cooling the regenerative resistor for the PSMR is running. If the fan is running, it is likely that the operating condition is severe. Lower the teaching speed set in the program. Refer to the maintenance manual for details.

SRVO–297 SERVO

Cause: The voltage of the PSM control power supply has dropped.

Remedy: Check that the three–phase input voltage is low. Replace the PSM or the PSMR if necessary. Refer to the maintenance manual for details.

SRVO–298 SERVO

Cause: The speed calculated in the servo circuit is abnormal.

Remedy: Contact the service personnel. To reset the alarm condition requires turning the power off and on again.

SRVO–300 SERVO Hand broken/HBK disabled

Cause: When the HBK setting is disabled, a hand–broken signal was detected.

Remedy: To remove the alarm condition, press the reset button.
Check whether the hand–broken signal circuit is connected to the robot.
If the circuit is connected to the robot, enable the hand–broken setting.
Refer to the maintenance manual.

SRVO–301 SERVO Hand broken/HBK dsbl (Rbt:%d)

Cause: When the HBK setting is disabled, a hand–broken signal was detected.

Remedy: To remove the alarm condition, press the reset button.
Check whether the hand–broken signal circuit is connected to the robot.
If the circuit is connected to the robot, enable the hand–broken setting.
Refer to the maintenance manual.

SRVO–302 SERVO Set Hand broken to ENABLE

Cause: When the HBK setting was disabled, a hand–broken signal was input.
The hand–broken setting is incorrect.

Remedy: Enable a hand broken.
To remove the alarm condition, press the reset button.
Refer to the maintenance manual.

SRVO–303 SERVO Set HBK to ENABLE (Rbt:%d)

Cause: When the HBK setting was disabled, a hand–broken signal was input.
The hand–broken setting is incorrect.

Remedy: Enable a hand broken.
To remove the alarm condition, press the reset button.
Refer to the maintenance manual.

SRVO–310 SERVO

Cause: Internal alarm

Remedy: Contact the service personnel.

SYST Error Codes (ID = 24)

SYST–001 PAUSE.G HOLD button is being pressed

Cause: You attempted an operation while the hold button (input) is pressed.

Remedy: Clear the hold button (input), and try the same operation.

SYST–002 PAUSE.G HOLD is locked by program

Cause: The condition that the robot is being held in is locked by the program and cannot be cleared. If a HOLD statement is executed in a KAREL program, the held condition can only be cleared by the same program using the UNHOLD statement/action, or by aborting the program. If you attempt a motion in such a condition, this error message is displayed.

Remedy: Wait until the UNHOLD statement is executed by the karel program, or abort the KAREL program.

SYST–003 WARN TP is enabled

Cause: The attempted operation could not be done because the teach pendant is enabled.

Remedy: Disable the teach pendant, and try the same operation again.

SYST–004 WARN SOP is enabled

Cause: The attempted operation could not be done because the System Operator Panel is enabled.

Remedy: Turn the REMOTE switch on the SOP to REMOTE side, and try the same operation again.

SYST–005 WARN UOP is the master device

Cause: The attempted operation could not be done because the User Operator Panel is enabled.

Remedy: Turn the REMOTE switch to local (if the operation is attempted from the SOP), or set the $RMT_MASTER system variable correctly.

SYST–006 WARN CRT is the master device

Cause: The attempted operation could not be done because CRT is the master device.

Remedy:
1 To perform the operation from the operator's panel, set the remote switch to the local position.
2 To perform the operation from the remote unit, set an appropriate value for $RMT_MASTER.

SYST–007 WARN NETWORK is the master device

Cause: The attempted operation could not be done because the NETWORK command processor is the master device.

Remedy:
1 To perform the operation from the operator's panel, set the remote switch to the local position.
2 To perform the operation from the remote unit, set an appropriate value for $RMT_MASTER.

SYST–008 WARN Nothing is the master device

Cause: The system variable $RMT_MASTER is set to disable all devices. Therefore, no remote device can issue motion.

Remedy:
1 To perform the operation from the operator's panel, set the remote switch to the local position.
2 To perform the operation from the remote unit, set an appropriate value for $RMT_MASTER.

SYST–009 WARN Safety Fence open

Cause: The attempted operation could not be done because the safety fence is open.

Remedy: Close the safety fence and try the same operation again.

SYST–010 WARN Max num task reached

Cause: The number of tasks has reached the maximum allowed.

Remedy: Abort one of the running task.

SYST–011 WARN Failed to run task

Cause: The system has failed to run the program.

Remedy: Determine the cause of the alarm on the alarm cause screen. Then, eliminate the cause.

SYST–012 WARN Not in remote

Cause: Remote condition is not satisfied.

Remedy: Turn the remote switch on.

SYST–013 WARN Invalid program number

Cause: The specified PNS number is not within its valid range.

Remedy: Specify a program number that is within the valid range of 1 to 9999.

SYST–014 WARN Program select failed

Cause: PNS operation has failed.
Remedy: Determine the cause of the alarm on the alarm cause screen. Then, eliminate the cause.

SYST–015 WARN Robot Service Request failed

Cause: RSR operation has failed.
Remedy: Determine the cause of the alarm on the alarm cause screen. Then, eliminate the cause.

SYST–016 WARN ENBL signal is off

Cause: ENBL signal on the User Operator Panel is off.
Remedy: Set ENBL signal ON.

SYST–017 WARN Single step operation effective

Cause: Single step operation is effective.
Remedy: Disable single step switch.

SYST–018 WARN Continuing from different line

Cause: You attempted to continue program execution from a line different that the paused line.
Remedy: Respond YES or NO in the prompt box on at the teach pendant.

SYST–019 WARN Program not selected

Cause: Program has not been selected.
Remedy: Select a program from the program select menu on the teach pendant, or by using PNS.

SYST–020 WARN Program not verified by PNS

Cause: The program specified by PNS is different then the program currently selected.
Remedy: Select a correct program from the program select menu on the teach pendant.

SYST–021 WARN System not ready, press RESET

Cause: Because program verification failed, program start–up is disabled.
Remedy: Press RESET to clear the error condition.

SYST–022 WARN PNS not zero, cannot continue

Cause: A paused program cannot continue if PNS input ports are not zero.
Remedy: Input an error clear signal to set all PNS inputs to 0, then input a start signal.

SYST–023 SYSTEM Teach Pendant communication error

Cause: A communication cable is broken.
Remedy: Check the communication cable. Replace the cable if necessary.

SYST–024 WARN PNSTROBE is OFF. Cannot start exec

Cause: Prod_start could not be processed because PNSTROBE is off.
Remedy: Set PNSTROBE input to ON.

SYST–025 WARN Teach Pendant is different type

Cause: The type of teach pendant being connected, is different from the one that was disconnected.
Remedy: Connect the same type of teach pendant, as that which was disconnected.

SYST–026 System normal power up

Cause: System has executed normal power startup.
Remedy: This is just a notification. You do not have to do anything for this warning message.

SYST–027 PAUSE.G HOT start failed (Error:%d)

Cause: HOT start has failed for one of the following reasons:

1 Power failed during system start up.
2 Flash ROM module was changed.
3 A run-time error occurred.
4 System internal error 1.
5 System internal error 2.

Remedy: COLD start is selected automatically.

SYST–028 WARN (%s) Program timed out

Cause: $PWR_HOT, $PWR_SEMI program has been aborted by the system due to time out (40sec).
Remedy: Decrease program size so that it can be executed within the time out limit.

SYST–029 PAUSE.G Robot was connected (Group:%d)

Cause: The connect/isolate key was turn to the connect side.
Remedy: This is just a notification. You do not have to do anything for this warning message.

SYST–030 PAUSE.G Robot was isolated (Group:%d)

Cause: The connect/isolate key was turn to the isolate side
Remedy: This is just a notification. You do not have to do anything for this warning message.

SYST–031 SYSTEM F–ROM parity

Cause: A parity error has been detected in the system FROM memory.
Remedy: Reload system software.

SYST–032 WARN ENBL signal from UOP is lost

Cause: ENBL input signal from the User Operator Panel is lost.
Remedy: Restore input signal.

SYST–033 WARN SFSPD signal from UOP is lost

Cause: SFSPD input signal from User Operator Panel is lost.
Remedy: Restore input signal.

SYST–034 WARN HOLD signal from SOP/UOP is lost

Cause: HOLD input signal from System Operator Panel/User Operator Panel is lost.
Remedy: Restore input signal.

SYST–035 WARN Low or No Battery Power in PSU.

Cause: Battery power in the PSU board is low.
Remedy: Replace the old battery with a new battery of the same kind.

SYST–036 WARN Semi power failure recovery

Cause: System did a semi-hot start.
Remedy: This is just a notification. You do not have to do anything for this warning message.

SYST–037 ABORT.G CE Sign key switch broken

Cause: Improper input from CE Sign key switch.
Remedy: Fix the CE Sign key switch.

SYST–038 PAUSE.G Operation mode T1 Selected

Cause: Operation mode T1 Selected

SYST–039 PAUSE.G Operation mode T2 Selected

Cause: Operation mode T2 Selected

SYST–040 PAUSE.G Operation mode AUTO Selected

Cause: Operation mode AUTO Selected

SYST–041 Ovrd Select could not ENABLED

Cause: DI index is invalid
Remedy: Please set valid DI index

SYST–042 DEADMAN defeated

Cause: The mode switch was changed from T1 or T2 mode to AUTO mode and the DEADMAN was already pressed.
The DEADMAN must be released when switching to AUTO mode
Remedy: Release the DEADMAN and press RESET.

SYST–043 TP disabled in T1/T2 mode

Cause: The mode selector is in T1 or T2 and the TP ON/OFF switch is in the OFF position
Remedy: Turn the TP ON/OFF switch to ON.Press RESET.

SYST–044 (Abnormal) TP disabled in T1/T2 mode

Cause: The mode selector is in T1 or T2 and the TP ON/OFF switch is in the OFF position and SVON is ON. This is an abnormal condition.
Remedy: Call your FANUC technical representative.

SYST–045 TP enabled in AUTO mode

Cause: The mode selector is in AUTO and the TP ON/OFF switch is in the ON position
Remedy: Turn the TP ON/OFF switch to OFF.
Press RESET.”

SYST–046 Control Reliable config mismatch

Cause: Either 1. Control Reliable hardware exists but the option has not been loaded, or 2. The Control Reliable option has been loaded but the hardware is not available.
Remedy: Consult our service representative.

SYST–047 Continuing from distant position

Cause: Attempt to continue program from distant position from stopped position.
Remedy: Respond ABORT or CONTINUE in the prompt box on at the teach pendant

SYST–048 NECALC couldn’t get work memory

Cause: The OS could not allocate work memory to the NUCALC software part.
The memory may be insufficient.
Remedy: Increase the controller memory.

SYST–049 SFCALC couldn’t get work memory

Cause: The OS could not allocate work memory to the SFCALC software part.
The memory may be insufficient.
Remedy: Increase the controller memory.

SYST–067 Panel HSSB disconnect

Cause: Communication with the panel board is disabled.
Remedy: Check the cable of the panel HSSB.

SYST–095 Remote diagnose internal error

Cause: An internal error occurred with the remote diagnosis function.
Remedy: Internal error

SYST–096 Designated task is not valid

Cause: A task specified by the PC in remote diagnosis is invalid.
Remedy: Check the remote diagnosis software of the PC.

SYST–097 Fail to initialize Modem

Cause: Modem initialization failed.
Remedy: Check if a modem is installed.
Check the modem type setting.

SYST–098 Card Modem is removed

Cause: The modem card was removed during communication.
Remedy: Reinsert the modem card, then restart the remote diagnosis function.
Check if the modem card is inserted into the PCMIA slot correctly.

SYST–099 Card Modem is not responded

Cause: There is no response from the modem card.
Remedy: Check if a modem card is inserted correctly.
Check the modem card.

SYST–100 DSR in Modem OFF

Cause: DSR was turned off during communication.

Remedy: Check the connection between R–J3 and the modem.
If a modem card is used, check if the modem card is not destroyed and if the modem card is inserted correctly.

SYST–101 Connection is stopped

Cause: The line was disconnected.

Remedy: Check the telephone line.

SYST–144 Bad DO specfied by %s

Cause: An invalid or unassigned SDO was allocated by a system variable.

Remedy: Change the value of the system variable to 0 (for no use) or a valid number.
Check that a specified SDO is allocated.

SYST–148 Dynamic Brake is Disabled

Cause: The dynamic brake release request signal SDI[$DYN_BRK.$DI_IDX] was turned on, so that the dynamic brake was released.

Remedy: IMSTP is generated while the dynamic brake release request signal is on.

SYST–149 Dynamic Brake is Enabled

Cause: The dynamic brake release request signal was turned off, so that the dynamic brake was actuated.

Remedy: This is not an alarm.

SYST–150 Cursor is not on line 1

Cause: The program was started on a line other than the first line.

Remedy: Reply Yes/No in response to the inquiry displayed on the screen.
Then, restart the program.

SYST–151 Start again (%s, %d)

Cause: After the program was started on a line other than the first line, Yes was replied to the inquiry displayed on the screen.

Remedy: Restart the program.

SYST–152 Cannot force DO's in AUTO mode

Cause: An attempt was made to output a signal in the AUTO mode.

Remedy: Before performing this operation, exit from the AUTO mode.

SYST–153 Cannot SIM/UNSIM DO's in AUTO mode

Cause: An attempt was made to simulate signal output in the AUTO mode.

Remedy: Before performing this operation, exit from the AUTO mode.

SYST–156 Unknown hard ware

Cause: The PCB does not match the control unit.

Remedy: Replace the PCB with a correct PCB.

SYST–157 CE/RIA software does not exist

Cause: The CE/RIA option is not installed.

Remedy: Install the CE/RIA option.

SYST–158 Robot cannot move in T2 mode

Cause: The tri–mode switch is set to the T2 mode.
In the T2 mode, the robot cannot be moved.

Remedy: Set the switch to the T1 or AUTO mode.

INTP Error Codes (ID = 12)

INTP–000 ABORT.G.G Req has not been processed yet

Cause: Internal system error.

Remedy: Contact our service center serving your locality.

INTP–001 PAUSE.G Cannot lock the motion grp

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

INTP–002 ABORT.G Program manager internal error

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

INTP–003 ABORT.G Invalid request

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

INTP–004 PAUSE.G Cannot ATTACH with TP enabled

Cause: The ATTACH statement requires the teach pendant to be disabled.
Remedy: Disable the teach pendant.

INTP–005 PAUSE.G Cannot release motion control

Cause: Motion control cannot be released.
Remedy: Abort the running or paused program.

INTP–100 to 102 ABORT.L (%s^4, %d^5) Internal error (PXnn)

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

INTP–103 ABORT.L (%s^4, %d^5) Program error

Cause: An error occurred while the program was running.
Remedy: Refer to the error cause code.

INTP–104 ABORT.L (%s^4, %d^5) Single step failed

Cause: Single step cannot be executed
Remedy: Refer to the error cause code.

INTP–105 ABORT.L (%s^4, %d^5) Run request failed

Cause: The program cannot be started.
Remedy: Refer to the error cause code.

INTP–106 PAUSE.L (%s^4, %d^5) Continue request failed

Cause: Program cannot be resumed.
Remedy: Refer to the error cause code.

INTP–107 ABORT.L (%s^4, %d^5) Pause request failed

Cause: An error occurred when program execution was held.
Remedy: Refer to the error cause code.

INTP–108 ABORT.L (%s^4, %d^5) Abort request failed

Cause: An error occurred when program execution was aborted.
Remedy: Refer to the error cause code.

INTP–109 WARN (%s^4, %d^5) BWD motion request failed

Cause: Backward motion cannot be executed.
Remedy: Refer to the error cause code.

INTP–110 (%s^4, %d^5) Get task status request failed

Cause: The specified task attribute is not found or is not read accessible.
Remedy: Check the attribute.

INTP–111 WARN (%s^4, %d^5) Skip statement request failed

Cause: The currently executing line cannot be changed.
Remedy: Refer to the error cause code.

INTP–112 PAUSE.L Cannot call interrupt routine

Cause: The interrupt routine cannot be executed.
Remedy: Refer to the error cause code.
Cause: If this alarm is raised together with the ”MEMO–004 WARN SPECIFIED PROGRAM IS IN USE” alarm, the conditions in the condition program are satisfied, and the desired action program is currently being edited, executed, or suspended.
Remedy: Select another program from the program list. Terminate the current action program.
Cause: If this alarm is raised together with the ”PROG–020 TASK IS ALREADY ABORTED” alarm, it is possible that the program that executed the monitoring start instruction has already been terminated when the conditions in the condition program are satisfied.
Remedy: When program monitoring is enabled, an action program can run only if the program that executed the monitoring start instruction is running.

INTP–113 PAUSE.L (%s^4, %d^5) Stop motion request failed

Cause: An error occurred when motion was stopped.
Remedy: Refer to the error cause code.

INTP–114 PAUSE.L (%s^4, %d^5) Cancel motion request failed

Cause: An error occurred when motion was canceled.
Remedy: Refer to the error cause code.

INTP–115 PAUSE.L (%s^4, %d^5) Resume motion request failed

Cause: An error occurred when motion was resumed.
Remedy: Refer to the error cause code.

INTP–116 PAUSE.L (%s^4, %d^5) Hold motion request failed

Cause: An error occurred when motion was held.
Remedy: Refer to the error cause code.

INTP–117 PAUSE.L (%s^4, %d^5) Unhold motion request failed

Cause: An error occurred when motion was unheld.
Remedy: Refer to the error cause code.

INTP–118 to 123 PAUSE.L (%s^4, %d^5) System error

Cause: Internal error of software.
Remedy: Refer to the error cause code.

INTP–124 ABORT.L (%s^4, %d^5) Invalid ITR routine

Cause: Internal error of software
Remedy: Refer to the error cause code.

INTP–125 ABORT.L Failed to convert position

Cause: The conversion of one position type to another failed.
Remedy: Refer to the error cause code.

INTP–126 ABORT.L Vision built–in return failed

Cause: The vision built-in failed to return.
Remedy: Refer to the error cause code.

INTP–127 WARN Power fail detected

Cause: Power failure was detected.
Remedy: Resume the program after hot start is complete.

INTP–128 PAUSE.L Pos reg is locked

Cause: Pos register is locked.
Remedy: Wait a moment. The error should resolve itself.

INTP–129 ABORT.L Cannot use motion group

Cause: You tried to lock the motion group even though this program cannot use motion groups.
Remedy: Clear the motion group mask in the program detail screen.

INTP–130 ABORT.L (%s^4, %d^5) Exec status recovery failed

Cause: Failed to recover execution status.
Remedy: Refer to the error cause code.

INTP–131 ABORT.L Number of stop exceeds limit

Cause: Too many stop data is created at one time.
Remedy: Decrease the number of stop data.

INTP–132 Unlocked groups specified

Cause: The specified motion groups are already unlocked.
Remedy: Change the specify of motion group.

INTP–133 Motion is already released

Cause: Some specified motion groups are already unlocked.
Remedy: Change the specify of motion group.
Lock the motion group.

INTP–134 Over automatic start Max counter

Cause: The automatic start was done the defined times but the alarm was not fixed.
Remedy: Manually fix the alarm.

INTP–135 Recovery DO OFF in auto start mode

Cause: The error recovery DO status is OFF in the automatic start feature
So the resume program cannot be exeucted automatically.
Remedy: Please check the condition of error recovery DO status

INTP–136 Can not use motion group for dry run function

Cause: In $PAUSE_PROG and $RESUME_PROG, a program using a motion group is specified.
Remedy: Specify a program not specifying a motion group.

INTP–137 Program specified by $PAUSE_PROG doesn't exist.

Cause: $PAUSE_PROG does not include a specified program.
Remedy: Check $PAUSE_PROG.

INTP–138 Program specified by $RESM_DRYPROG doesn't exist.

Cause: $RESUME_PROG does not include a specified program.
Remedy: Check $RESUME_PROG.

INTP–139 (%s^4, %d^5) Local variable request failed

Cause: Execution failed.
Remedy: Check the alarm history screen to see if another alarm is output.

INTP–200 PAUSE.L (%s^4, %d^5) Unimplemented TP instruction

Cause: This instruction cannot be used.
Remedy: Make sure that the appropriate option is loaded.

INTP–201 PAUSE.L (%s^4, %d^5) Untaught element encountered

Cause: The program contains a portion without teaching data.
The specified condition program contains an error (statement without teaching data).
Remedy: Teach the instruction.

INTP–202 PAUSE.L (%s^4, %d^5) Syntax error

Cause: Instruction syntax error.
Remedy: Reteach the instruction.

INTP–203 PAUSE.L (%s^4, %d^5) Variable type mismatch

Cause: The variable type is not correct.
Remedy: Check the variable type.

INTP–204 PAUSE.L (%s^4, %d^5) Invalid value for index

Cause: The index value is invalid.
Remedy: Check the index value.

INTP–205 PAUSE.L (%s^4, %d^5) Analog port access error

Cause: Analog I/O is not functioning properly.
Remedy: Refer to the error cause code.

INTP–206 PAUSE.L (%s^4, %d^5) Digital port access error

Cause: Digital I/O is not functioning properly.
Remedy: Refer to the error cause code.

INTP–207 PAUSE.L (%s^4, %d^5) Group I/O port access error

Cause: Group I/O is not functioning properly.
Remedy: Refer to the error cause code.

INTP–208 PAUSE.L (%s^4, %d^5) Divide by 0

Cause: Division by 0 was executed.
Remedy: Check the value.

INTP–209 PAUSE.L (%s^4, %d^5) SELECT is needed

Cause: A CASE instruction was executed before a SELECT instruction.
Remedy: Add a SELECT instruction before the CASE instruction.

INTP–212 PAUSE.L (%s^4, %d^5) Invalid value for OVERRIDE

Cause: The indicated value cannot be used for the OVERRIDE instruction.
Remedy: Check the value.

INTP–213 PAUSE.L %s^7 (%s^4, %d^5) UALM[%d^9]

Cause: A user alarm occurred.
Remedy: Refer to the user alarm code.

INTP–214 PAUSE.L (%s^4, %d^5) Specified group not locked

Cause: The position register or frame setup instructions were executed in a program without a motion group.
Remedy: Set up the motion group in the program DETAIL screen. Refer to the user alarm code.

INTP–215 PAUSE.L (%s^4, %d^5) Group mismatch

Cause: The position data is invalid.
Remedy: Check the position data.

INTP–216 PAUSE.L (%s^4, %d^5) Invalid value for group number

Cause: The indicated value is invalid for the motion group number.
Remedy: Check the value.

INTP–217 PAUSE.L (%s^4, %d^5) SKIP CONDITION needed

Cause: The SKIP instruction was executed before a SKIP CONDITION instruction.
Remedy: Add a SKIP CONDITION instruction.

INTP–218 PAUSE.L (%s^4, %d^5) Skip failed

Cause: The SKIP instruction or SKIP CONDITION instruction cannot be executed.
Remedy: Refer to the error cause code.

INTP–219 ABORT.L (%s^4, %d^5) Pause task failed

Cause: The pause instruction cannot be executed.
Remedy: Refer to the error cause code.

INTP–220 ABORT.L (%s^4, %d^5) Abort task failed

Cause: The ABORT instruction cannot be executed.
Remedy: Refer to the error cause code.

INTP–221 PAUSE.L (%s^4, %d^5) Application failed

Cause: The application instruction cannot be executed.
Remedy: Refer to the error cause code.

INTP–222 PAUSE.L (%s^4, %d^5) Call program failed

Cause: The program CALL instruction cannot be executed.
Remedy: Refer to the error cause code.

INTP–223 PAUSE.L (%s^4, %d^5) Delay time failed

Cause: The WAIT instruction cannot be executed.
Remedy: Refer to the error cause code.

INTP–224 PAUSE.L (%s^4, %d^5) Jump label failed

Cause: The BRANCH instruction cannot be executed.
Remedy: Refer to the error cause code.

INTP–225 PAUSE.L (%s^4, %d^5) Motion statement failed

Cause: The MOTION instruction cannot be executed.
Remedy: Refer to the error cause code.

INTP–226 PAUSE.L (%s^4, %d^5) Read position register failed

Cause: The position register cannot be read.
Remedy: Refer to the error cause code.

INTP–227 PAUSE.L (%s^4, %d^5) Write position register failed

Cause: The position register cannot be written.
Remedy: Refer to the error cause code.

INTP–228 PAUSE.L (%s^4, %d^5) Read register failed

Cause: The register cannot be read.
Remedy: Refer to the error cause code.

INTP–229 PAUSE.L (%s^4, %d^5) Write register failed

Cause: The register cannot be written.
Remedy: Refer to the error cause code.

INTP–230 PAUSE.L (%s^4, %d^5) Wait condition failed

Cause: A condition WAIT instruction cannot be executed.
Remedy: Refer to the error cause code.

INTP–231 PAUSE.L (%s^4, %d^5) Read next line failed

Cause: The next line cannot be read.
Remedy: Refer to the error cause code.

INTP–232 PAUSE.L (%s^4, %d^5) Invalid frame number
Cause: The frame number is invalid.
Remedy: Check the frame number.

INTP–233 PAUSE.L (%s^4, %d^5) Read frame value failed
Cause: The specified frame cannot be read.
Remedy: Refer to the error cause code.

INTP–234 PAUSE.L (%s^4, %d^5) Write frame value failed
Cause: The specified frame cannot be written.
Remedy: Refer to the error cause code.

INTP–235 PAUSE.L (%s^4, %d^5) Read pos item failed
Cause: The position variable cannot be read.
Remedy: Refer to the error cause code.

INTP–236 PAUSE.L (%s^4, %d^5) Write pos item failed
Cause: The position variable cannot be written.
Remedy: Refer to the error cause code.

INTP–237 WARN (%s^4, %d^5) No more motion for BWD
Cause: Backward execution cannot be executed any further because the current program line is at the top.
Remedy: Stop using backward execution at this point.

INTP–238 WARN (%s^4, %d^5) BWD execution completed
Cause: Backward execution was completed.
Remedy: Do not use backward execution from this point.

INTP–239 WARN (%s^4, %d^5) Cannot execute backwards
Cause: This instruction cannot be executed backwards.
Remedy: Set the cursor to execute at the next line.

INTP–240 PAUSE.L (%s^4, %d^5) Incompatible data type
Cause: The specified data type in the PARAMETER instruction is invalid for the parameter type.
Remedy: Check the data type.

INTP–241 PAUSE.L (%s^4, %d^5) Unsupported parameter
Cause: This type of parameter cannot be used.
Remedy: Check the parameter type.

INTP–242 PAUSE.L (%s^4, %d^5) Offset value is needed
Cause: An OFFSET instruction was executed before an OFFSET CONDITION instruction. A position register was not taught in the OFFSET PR[] instruction.
Remedy: Add an OFFSET CONDITION instruction before the OFFSET instruction. Teach the position register.

INTP–243 ABORT.G (%s^4, %d^5) Def grp is not specified
Cause: This program has no motion group defined. The MOTION instruction cannot be executed.
Remedy: Remove the MOTION instruction or set up the motion group in the program DETAIL screen.

INTP–244 PAUSE.L (%s^4, %d^5) Invalid line number
Cause: The input line number is incorrect.
Remedy: Check the line number.

INTP–245 PAUSE.L (%s^4, %d^5) RCV stmt failed
Cause: The RECEIVE R[] instruction cannot be executed.
Remedy: Refer to the error cause code.

INTP–246 PAUSE.L (%s^4, %d^5) SEMAPHORE stmt failed

Cause: The SEMAPHORE instruction cannot be executed.
Remedy: Refer to the error cause code.

INTP–247 PAUSE.L (%s^4, %d^5) Pre exec failed

Cause: Internal error of software.
Remedy: Contact our service center serving your locality.

INTP–248 PAUSE.L (%s^4, %d^5) MACRO failed

Cause: The MACRO instruction cannot be executed.
Remedy: Refer to the error cause code.

INTP–249 PAUSE.L Macro is not set correctly

Cause: The MACRO setup was invalid.
Remedy: Check the MACRO setup.

INTP–250 PAUSE.L (%s^4, %d^5) Invalid uframe number

Cause: The user frame number is invalid.
Remedy: Refer to the error cause code.

INTP–251 PAUSE.L (%s^4, %d^5) Invalid utool number

Cause: The tool frame number is invalid.
Remedy: Refer to the error cause code.

INTP–252 PAUSE.L User frame number mismatch

Cause: The user frame number in the positional data is not the same as the currently selected user frame number.
Remedy: Check the user frame number.

INTP–253 PAUSE.L Tool frame number mismatch

Cause: The tool frame number in the positional data is not the same as the currently selected tool frame number.
Remedy: Check the tool frame number.

INTP–254 PAUSE.L (%s^4, %d^5) Parameter not found

Cause: The specified parameter name cannot be found.
Remedy: Check the parameter name.

INTP–255 PAUSE.L (%s^4, %d^5) CAL_MATRIX failed

Cause: The CAL_MATRIX instruction cannot be executed.
Remedy: Refer to the error cause code.

INTP–256 PAUSE.L (%s^4, %d^5) No data for CAL_MATRIX

Cause: The origin 3 points or destination 3 points are not taught.
Remedy: Teach the origin 3 points or destination 3 points.

INTP–257 PAUSE.L (%s^4, %d^5) Invalid delay time

Cause: The wait time value is negative or exceeds the maximum value of 2147483.647 sec.
Remedy: Input a correct value.

INTP–258 PAUSE.L (%s^4, %d^5) Weld port access error

Cause: The weld is not functioning properly.
Remedy: Refer to the error cause code.

INTP–259 PAUSE.L (%s^4, %d^5) Invalid position type

Cause: The data type of the position register was taught using joint type.
Remedy: Change position register data to Cartesian.

INTP–260 PAUSE.L (%s^4, %d^5) Invalid torque limit value

Cause: The specified torque limit is not in the range of 0.0 to 100.0.
Remedy: Specify a torque limit in the range of 0.0 to 100.0.

INTP–261 PAUSE.L (%s^4, %d^5) Array subscript missing

Cause: No array element number is specified.
Remedy: Specify an array element number.

INTP–262 PAUSE.L (%s^4, %d^5) Field name missing

Cause: No element name is specified.
Remedy: Specify an element name.

INTP–263 PAUSE.L (%s^4, %d^5) Invalid register type

Cause: The register type is not valid.
Remedy: Check the register type.

INTP–265 PAUSE.L (%s^4, %d^5) Invalid value for speed value

Cause: The indicated value cannot be used for the AF instruction.
Remedy: Specify a value in the range of 0 to 100.

INTP–266 ABORT.L (%s^4, %d^5) Mnemonic in interrupt is failed

Cause: The execution of mnemonic instructions in the KAREL interrupt program failed.
Remedy: insert a CANCEL or STOP instruction before calling an interrupt routine.

INTP–267 PAUSE.L (%s^4, %d^5) RUN stmt failed

Cause: Specified program is already running.
Remedy: Abort the specified program.

INTP–268 PAUSE.L (%s^4, %d^5) This statement only one in each line

Cause: A single line contains more than one application instruction. Only one of these statements can exist per line.
Remedy: Delete the extra statement.

INTP–269 PAUSE.L (%s^4, %d^5) Skip statement only one in each line

Cause: A single line contains more than one skip instruction. Only one Skip statement can exist per line.
Remedy: Delete the extra Skip statement.

INTP–270 PAUSE.L (%s^4, %d^5) different group cannot BWD

Cause: During backward execution, a move is encountered that has a different group number from the previous motion statement.
Remedy: Use FWD execution carefully.

INTP–271 WARN (%s^4, %d^5) Excessive torque limit value

Cause: The torque limit value was modified to exceed it's maximum value. The torque limit value was clamped at the upper torque limit.
Remedy: Set torque limit value less than or equal to the maximum value.

INTP–272 PAUSE.L (%s^4, %d^5) Unsupported operator

Cause: This operator is not supported.
Remedy: Check the operator.

INTP–274 (%s^4, %d^5) CH program error

Cause: This monitor statement cannot be executed.
Remedy: Refer to the error cause code.
Use MENU to display the Alarm Log screen.
Cause: If this alarm is raised together with the "MEMO–004 WARN SPECIFIED PROGRAM IN USE" alarm, the specified condition program is currently being edited.
Remedy: Select another program from the program list.
Cause: If this alarm is raised together with the "INTP–275 PAUSE.L INVALID SUB TYPE OF CH PROGRAM" alarm, the sub type of the specified condition program may not be CH, or that program may not exist.

INTP–275 Invalid sub type of CH program

Cause: The sub type of specified ch program cannot be used.
Remedy: Check the sub type of this CH program.

INTP–276 (%s^4, %d^5) Invalid combination of motion option

Cause: The motion option instructions (SKIP, TIME BEFORE/AFTER, and application instruction) cannot be taught together
Remedy: Delete the motion option instruction

INTP–277 (%s^4, %d^5) Internal MACRO EPT data mismatch

Cause: The EPT index in macro table doesn't point the program name defined in macro table. That is, the EPT index in macro table is incorrect.
Remedy: Please set the correct EPT index for the program name defined in macro table.

INTP–278 %s^7

Cause: The DI monitor alarm for auto error recovery function occurs.
Remedy: This is a user definded alarm.

INTP–279 (%s^4, %d^5) Application instruction mismatch

Cause: The application instruction was executed. But this application instruction doesn't match to the application process data of this program.
Remedy: Please change the application process data of this program to the adequate application for this application instruction.

INTP–280 (%s^4, %d^5) Application data mismatch

Cause: The application data of called program is different from that of the original program.
Remedy: Please change the structure of the program.

INTP0281 No application data

Cause: This program doesn't have the application data.
Remedy: Please define the application data in the program detail screen.

INTP–282 (%s^4, %d^5) Fast fault status mismatch

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

INTP–283 (%s^4, %d^5) Stack over flow for fast fault recovery

Cause: Stack over flow to record the fast fault recovery nesting data
Remedy: Reduce the nesting of the program

INTP–284 No detection of fast fault recovery

Cause: The point for the fast fault recover cannot detected
Remedy: Contact our service center serving your locality.

INTP–285 Karel program cannot entry in fast fautl recovery

Cause: The fast entry cannot be performed in the karel program.
Remedy: Use the TP program.

INTP–286 MAINT program isn't defined in fast fautl recovery

Cause: MAINT program is not defined in fast fault recovery.
Remedy: Set MAINT program.

INTP–287 Fail to execute MAINT program

Cause: It failed to execute MAINT program
Remedy: Confirm the MAINT program name is correct or MAINT program exist in acutual.

INTP–288 (%s^4, %d^5) Parameter does not exist

Cause: The parameter designated by AR register does not exist.
Remedy: Please confirm the index of AR register and the parameter in CALL/MACRO command in main program.

INTP–289 Can't save ffast point at program change

Cause: When fast fault is enabled, the program was paused at the part of program change
Remedy: Check whether the CONT terminaton exists at end of sub–program.
If it exist, change it to FINE.
This is the limitation of the fast fault recovery function.

INTP–290 Fast fault recovery position is not saved

Cause: During fast fault recovery sequence, any alarm occurs. So the fast fault recovery position is not saved.

INTP–291 (%s^4, %d^5) Index for AR is not correct

Cause: The AR register number is incorrect. At present, this alarm is not issued.
Remedy: Check the index of the AR register and the argument specified in the call/macro instruction of the main program.

INTP–292 more than 6 motion with DB executed

Cause: Six or more advanced execution (distance) motions overlapped each other.
Remedy: Modify the teaching so that six or more advanced execution (distance) motions do not overlap each other.

INTP–293 (%s,%d)DB too small(away)(%dmm)

Cause: The condition for advanced execution (distance) is not satisfied.
Remedy: Increase the specified distance value.

INTP–294 TPE parameter error

Cause: An incorrect argument is specified for call/macro instruction execution.
Remedy: Contact your FANUC customer service representative, and inform the representative of an alarm message displayed.

INTP–295 (%s,%d)DB too small(away)(%dmm)

Cause: The condition for advanced execution (distance) is not satisfied.
Remedy: Increase the specified distance value.

INTP–296 (%s,%d) $SCR_GRP[%d].$M_POS_ENB is FALSE

Cause: Advanced execution (distance) does not function when $SCR_GRP[].$M_POS_ENB is FALSE.
Remedy: Change $SCR_GRP[].$M_POS_ENB to TRUE.

INTP–297 (%s,%d)DB too small(done)(%dmm)

Cause: A motion statement ended before the condition for advanced execution (distance) is satisfied.
Remedy: Increase the specified distance value.

INTP–300 ABORT.L (%s^4, %d^5) Unimplemented P–code

Cause: KAREL program error. This KAREL statement cannot be executed.
Remedy: Check the KAREL translator software version.

INTP–301 ABORT.L (%s^4, %d^5) Stack underflow

Cause: KAREL program error. Execution entered into a FOR loop by the GOTO statement.
Remedy: A GOTO statement cannot be used to enter or exit a FOR loop. Check the label of the GOTO statement.

INTP–302 ABORT.L (%s^4, %d^5) Stack overflow

Cause:
1 A recursive program instruction was executed repeatedly without limit.
2 Too many programs are called at one time.

Remedy:
1 Before executing a recursive instruction, perform programming so that a call to the instruction can be cleared at any point of execution.
2 Reduce the number of programs to be called at any one time. For KAREL programs, the stack size can be increased.

INTP–303 ABORT.L (%s^4, %d^5) Specified value exceeds limit

Cause: KAREL program error. The specified value exceeds the maximum limit.
Remedy: Check the value.

INTP–304 ABORT.L (%s^4, %d^5) Array length mismatch

Cause: KAREL program error. The dimensions of the arrays are not the same.
Remedy: Check the dimensions of the arrays.

INTP–305 ABORT.L (%s^4, %d^5) Error related condition handler

Cause: KAREL program error. A condition handler error occurred.
Remedy: Refer to the error cause code.

INTP–306 ABORT.L (%s^4, %d^5) Attach request failed

Cause: KAREL program error. The ATTACH statement failed.
Remedy: Refer to the error cause code.

INTP–307 ABORT.L (%s^4, %d^5) Detach request failed

Cause: KAREL program error. The DETACH statement failed.
Remedy: Refer to the error cause code.

INTP–308 ABORT.L (%s^4, %d^5) No case match is encountered

Cause: KAREL program error. The CASE statement does not match any branches.
Remedy: Check the CASE value and branches.

INTP–309 ABORT.L (%s^4, %d^5) Undefined WITHCH parameter

Cause: KAREL program error. The specified parameter cannot be used in the with clause of the condition handler.
Remedy: Check the parameter.

INTP–310 ABORT.L (%s^4, %d^5) Invalid subscript for array

Cause: KAREL program error. The index of the array is invalid.
Remedy: Check the length of the array and index value.

INTP–311 PAUSE.L (%s^4, %d^5) Uninitialized data is used

Cause: KAREL program error. Untaught or uninitialized data was used.
Remedy: Teach or initialize the data before using it.

INTP–312 ABORT.L (%s^4, %d^5) Invalid joint number

Cause: KAREL program error. The wrong axis number was used.
Remedy: Check the axis number and the data value.

INTP–313 ABORT.L (%s^4, %d^5) Motion statement failed

Cause: KAREL program error. The MOTION statement cannot be executed.
Remedy: Refer to the error cause code.

INTP–314 ABORT.L (%s^4, %d^5) Return program failed

Cause: KAREL program error. Execution cannot be returned from the routine.
Remedy: Refer to the error cause code.

INTP–315 ABORT.L (%s^4, %d^5) Built–in execution failed

Cause: KAREL program error. A built-in routine error occurred.
Remedy: Refer to the error cause code.

INTP–316 ABORT.L (%s^4, %d^5) Call program failed

Cause: KAREL program error. The routine cannot be called.
Remedy: Verify the routine is loaded by referring to the error cause code.

INTP–317 ABORT.L (%s^4, %d^5) Invalid condition specified

Cause: KAREL program error. The specified condition was invalid.
Remedy: Check the condition.

INTP–318 ABORT.L (%s^4, %d^5) Invalid action specified

Cause: KAREL program error. The specified action was invalid.
Remedy: Check the action.

INTP–319 ABORT.L (%s^4, %d^5) Invalid type code

Cause: KAREL program error. The data type was invalid.
Remedy: Check the data type.

INTP–320 ABORT.L (%s^4, %d^5) Undefined built–in

Cause: KAREL program error. The built-in routine is not defined.
Remedy: Check the appropriate option is loaded.

INTP–321 ABORT.L (%s^4, %d^5) END stmt of a func rtn

Cause: KAREL program error. The END statement was executed in a function routine instead of a RETURN statement.
Remedy: Add a RETURN statement to the function routine.

INTP–322 ABORT.L (%s^4, %d^5) Invalid arg val for builtin

Cause: KAREL program error. The argument value of a built-in routine was wrong.
Remedy: Check the argument value.

INTP–323 ABORT.L (%s^4, %d^5) Value overflow

Cause: KAREL program error. The data value for the variable was too large.
Remedy: Check the variable's type and data value.

INTP–324 ABORT.L (%s^4, %d^5) Invalid open mode string

Cause: KAREL program error. The usage string in the OPEN FILE statement was invalid.
Remedy: Check the usage string in the OPEN FILE statement.

INTP–325 ABORT.L (%s^4, %d^5) Invalid file string

Cause: KAREL program error. The file string in the OPEN FILE statement was invalid.
Remedy: Check the file string.

INTP–326 ABORT.L (%s^4, %d^5) File var is already used

Cause: KAREL program error. The FILE variable is already being used.
Remedy: Close the file before reusing the FILE variable or add a new FILE variable.

INTP–327 ABORT.L (%s^4, %d^5) Open file failed

Cause: KAREL program error. The file could not be opened.
Remedy: Refer to the error cause code.

INTP–328 ABORT.L (%s^4, %d^5) File is not opened

Cause: KAREL program error. The specified file was not opened before operation.
Remedy: Open the file before operation.

INTP–329 ABORT.L (%s^4, %d^5) Write variable failed

Cause: KAREL program error.
Remedy: Refer to the error cause code.

INTP–330 ABORT.L (%s^4, %d^5) Write file failed

Cause: KAREL program error. Writing to the file failed.
Remedy: Refer to the error cause code.

INTP–331 ABORT.L (%s^4, %d^5) Read variable failed

Cause: KAREL program error. Reading the variable failed.
Remedy: Refer to the error cause code.

INTP–332 ABORT.L (%s^4, %d^5) Read data is too short

Cause: KAREL program error. Data read from the file is too short.
Remedy: Make sure the data in the file is valid.

INTP–333 ABORT.L (%s^4, %d^5) Invalid ASCII string for read

Cause: KAREL program error. The string read from the file is wrong.
Remedy: Check the data of the file.

INTP–334 ABORT.L (%s^4, %d^5) Read file failed

Cause: KAREL program error. Reading from the file failed.
Remedy: Refer to the error cause code.

INTP–335 ABORT.L (%s^4, %d^5) Cannot open pre–defined file

Cause: KAREL program error. A file pre-defined by the system cannot be opened.
Remedy: Use the file defined by the system without opening it.

INTP–336 ABORT.L (%s^4, %d^5) Cannot close pre–defined file

Cause: KAREL program error. A file pre-defined by the system cannot be closed.
Remedy: Do not try to close it.

INTP–337 ABORT.L (%s^4, %d^5) Invalid routine type

Cause: KAREL program error. This routine cannot be used.
Remedy: Make sure you have the correct routine type and name.

INTP–338 ABORT.L (%s^4, %d^5) Close file failed

Cause: KAREL program error. Closing the file failed.
Remedy: Refer to the error cause code.

INTP–339 ABORT.L (%s^4, %d^5) Invalid program name

Cause: KAREL program error. The program name is invalid.
Remedy: Make sure you have the correct program name.

INTP–340 ABORT.L (%s^4, %d^5) Invalid variable name

Cause: KAREL program error. The variable name is invalid.
Remedy: Make sure you have the correct variable name.

INTP–341 ABORT.L (%s^4, %d^5) Variable not found

Cause: KAREL program error. The variable cannot be found.
Remedy: Verify the program name and variable name.

INTP–342 ABORT.L (%s^4, %d^5) Incompatible variable

Cause: KAREL program error. The data type defined by the BYNAME function and the variable type are mismatched.
Remedy: Make sure you have the correct data type and variable type.

INTP–343 ABORT.L (%s^4, %d^5) Reference stack overflow

Cause: KAREL program error. Too many variables are passed using the BYNAME function.
Remedy: Decrease the number of BYNAME functions.

INTP–344 ABORT.L (%s^4, %d^5) Readahead buffer overflow

Cause: KAREL program error. The buffer to read ahead from the device overflowed.
Remedy: Increase the buffer size.

INTP–345 ABORT.L (%s^4, %d^5) Pause task failed

Cause: KAREL program error. The PAUSE statement cannot be executed.
Remedy: Refer to the error cause code.

INTP–346 ABORT.L (%s^4, %d^5) Abort task failed

Cause: KAREL program error. The ABORT statement cannot be executed.
Remedy: Refer to the error cause code.

INTP–347 ABORT.L (%s^4, %d^5) Read I/O value failed

Cause: KAREL program error. The digital input signal cannot be input.
Remedy: Refer to the error cause code.

INTP–348 ABORT.L (%s^4, %d^5) Write I/O value failed

Cause: KAREL program error. The digital output signal cannot be output.
Remedy: Refer to the error cause code.

INTP–349 ABORT.L (%s^4, %d^5) Hold motion failed

Cause: KAREL program error. The HOLD statement cannot be executed.
Remedy: Refer to the error cause code.

INTP–350 ABORT.L (%s^4, %d^5) Unhold motion failed

Cause: KAREL program error. The UNHOLD statement cannot be executed.
Remedy: Refer to the error cause code.

INTP–351 ABORT.L (%s^4, %d^5) Stop motion failed

Cause: KAREL program error. The STOP statement cannot be executed.
Remedy: Refer to the error cause code.

INTP–352 ABORT.L (%s^4, %d^5) Cancel motion failed

Cause: KAREL program error. The CANCEL statement cannot be executed.
Remedy: Refer to the error cause code.

INTP–353 ABORT.L (%s^4, %d^5) Resume motion failed

Cause: KAREL program error. The RESUME statement cannot be executed.
Remedy: Refer to the error cause code.

INTP–354 ABORT.L (%s^4, %d^5) Break point failed

Cause: KAREL program error. The break point function cannot be executed.
Remedy: Refer to the error cause code.

INTP–355 ABORT.L (%s^4, %d^5) AMR is not found

Cause: KAREL program error. The AMR operated by the RETURN_AMR built-in routine was not found.
Remedy: Check program operation.

INTP–356 ABORT.L (%s^4, %d^5) AMR is not processed yet

Cause: KAREL program error. The RETURN_AMR built-in routine cannot be used for an unoperated AMR.
Remedy: Operate the AMR using the WAIT_AMR built-in routine.

INTP–357 ABORT.L (%s^4, %d^5) WAIT_AMR is cancelled

Cause: KAREL program error. The execution of the WAIT_AMR built–in routine was cancelled.
Remedy: The program executing the WAIT_AMR must be restarted.

INTP–358 ABORT.L (%s^4, %d^5) Timeout at read request

Cause: KAREL program error. The READ statement timed out.
Remedy: Check the device being read.

INTP–359 ABORT.L (%s^4, %d^5) Read request is nested

Cause: KAREL program error. Another READ statement was executed while a READ statement was waiting for input.

Remedy: Remove nested reads.

INTP–360 ABORT.L (%s^4, %d^5) Vector is 0

Cause: KAREL program error. The vector value was invalid.

Remedy: Check the vector value.

INTP–361 PAUSE.L (%s^4, %d^5) FRAME:P2 is same as P1

Cause: KAREL program error. The X–axis direction cannot be calculated in the FRAME built-in routine because P1 and P2 are the same point.

Remedy: Teach P1 and P2 as different points.

INTP–362 PAUSE.L (%s^4, %d^5) FRAME:P3 is same as P1

Cause: KAREL program error. The X–Y plane cannot be calculated in the FRAME built-in routine because P1 and P3 are the same point.

Remedy: Teach P1 and P3 as different points.

INTP–363 PAUSE.L (%s^4, %d^5) FRAME:P3 exists on line P2–P1

Cause: KAREL program error. The X–Y plane cannot be calculated in the FRAME built-in routine because P3 is located in the X–axis direction.

Remedy: Teach P3 out of the X–axis direction.

INTP–364 ABORT.L (%s^4, %d^5) String too short for data

Cause: KAREL program error. The target string is too short.

Remedy: Increase the target string size.

INTP–365 ABORT.L (%s^4, %d^5) Predefined window not opened

Cause: KAREL program error. A FILE pre-defined by the system is not opened.

Remedy: Check the use of this file.

INTP–366 ABORT.L (%s^4, %d^5) I/O status is not cleared

Cause: KAREL program error. The last file operation failed.

Remedy: Reset the error using the CLR_IO_STAT built-in routine.

INTP–367 ABORT.L (%s^4, %d^5) Bad base in format

Cause: KAREL program error. I/O mode operates only from binary to hexdecimal.

Remedy: Check the specified mode.

INTP–368 PAUSE.L (%s^4, %d^5) Cannot use specified program

Cause: KAREL program error. The specified program cannot be used.

Remedy: Refer to the error cause code.

INTP–369 ABORT.L (%s^4, %d^5) Timeout at WAIT_AMR

Cause: KAREL program error. The WAIT_AMR built-in routine timed out.

Remedy: If an AMR was expected within the time–out value check logic in the task that should have posted the AMR

INTP–370 ABORT.L (%s^4, %d^5) Vision CPU not plugged in

Cause: KAREL program error. The vision CPU board is not plugged in.

Remedy: Plug in the vision CPU board.

INTP–371 ABORT.L (%s^4, %d^5) Vision built–in overflow

Cause: KAREL program error. The operation overflowed in the vision built-in routine.

Remedy: Modify your program so that fewer vision built-ins are executing at the same time.

INTP–372 ABORT.L (%s^4, %d^5) Undefined vision built–in

Cause: KAREL program error. The vision built-in routine is not defined.
Remedy: Check the appropriate option is loaded.

INTP–373 ABORT.L (%s^4, %d^5) Undefined vision parameter type

Cause: KAREL program error. The parameter to the vision built-in routine is invalid.
Remedy: Check the parameter of the vision built-in routine.

INTP–374 ABORT.L (%s^4, %d^5) Undefined vision return type

Cause: KAREL program error. The return value from the vision built-in routine is invalid.
Remedy: Check the return value from the vision built-in routine.

INTP–375 (%s^4, %d^5) System var passed using BYNAME

Cause: This alarm is related to the KAREL program. With the BYNAME function, no system variable can be used. Use Pass without BYNAME, GET_VAR, or SET_VAR.

INTP–376 ABORT.L (%s^4, %d^5) Motion in interrupt is failed

Cause: There is no CANCEL or STOP instruction.
Remedy: insert a CANCEL or STOP instruction before call a interrupt routine.

INTP–377 WARN (%s^4, %d^5) Local COND recovery failed

Cause: This local condition cannot be recovered.
Remedy: Refer to the error cause code.

INTP–378 WARN (%s^4, %d^5) Local variable is used

Cause: Local variable or parameter is used for the condition.
Remedy: Use global variable to recover local condition.

INTP–379 ABORT.L Bad condition handler number

Cause: An invalid condition handler number was used in either a condition handler definition, or with an ENABLE, DISABLE, or PURGE statement or action.
Remedy: Correct the condition handler number. Condition handler numbers must be in the range of 1–1000.

INTP–380 ABORT.L Bad program number

Cause: An invalid program number has been specified.
Remedy: Use a valid program number. Program numbers must be in the range of
1 – $SCR,$MAXNUMTASK + 2.

INTP–381 (%s^4, %d^5) Invalid Delay Time

Cause: An invalid delay time has been specified in DELAY statement.
Remedy: Use a valid delay time. Delay time must be in the range 0..86400000 .

INTP–382 (%s^4, %d^5) Invalid bit field value

Cause: An invalid value has been specified in bit field
Remedy: Use a valid value for the bit field.

INTP–383 (%s^4, %d^5) Path node out of range

Cause: The specified path node is out of range.
Remedy: Check the path node.

INTP–400 ABORT.L (%s^4, %d^5) Number of motions exceeded

Cause: Too many motions are executed at the same time.
Remedy: Decrease the number of motions executed at the same time. Execute the next motion after the completion of the last motion.

INTP–401 ABORT.L (%s^4, %d^5) Not On Top Of Stack

Cause: Paused motion exists after the motion was resumed.
Remedy: Resume the motion that was previously paused.

INTP–420 (%s^4, %d^5) OFIX is not available

Cause: The attitude fix instruction cannot be used.
Remedy: Check the motion format and the motion addition instruction.

INTP–421 (%s^4, %d^5) Stitch disable(S/S)

Cause: The single step mode is set.
Remedy: Cancel the single step mode.

INTP–422 (%s^4, %d^5) Stitch enable signal off

Cause: The stitch enable signal is set to OFF.
Remedy: Set the stitch enable signal to ON.

INTP–423 (%s^4, %d^5) Eq.condition signal error

Cause: The equipment condition signal is incorrect.
Remedy: Check the equipment condition signal.

INTP–424 (%s^4, %d^5) Stitch speed error

Cause: The stitch speed value is incorrect.
Remedy: Check the stitch speed value.

INTP–425 (%s^4, %d^5) Illegal motion type(J)

Cause: The stitch function cannot be used with joint motion.
Remedy: Make a change to linear motion.

INTP–426 (%s^4, %d^5) Another prog is in stitching

Cause: Another program is using the stitch function.
Remedy: Terminate the program that is using the stitch function.

INTP–450 (%s^4, %d^5) Cannot call KAREL program

Cause: From the master/slave/single slave program of the robot link, the KAREL program was called.
Remedy: Do not call the KAREL program from the master/slave/single slave program.

INTP–451 (%s^4, %d^5) Cannot call Motion program

Cause: From the master/slave/single slave program of the robot link, a normal program with a motion group was called.
Remedy: Do not call a normal program with a motion group from the master/slave/single slave program.

INTP–452 (%s^4, %d^5) Robot link type mismatch

Cause: From the master/slave/single slave program of the robot link, a program of a different type was called.
Remedy: Do not call a program of a different type from the master/slave/single slave program.

INTP–453 (%s^4, %d^5) Not in remote

Cause: The slave program of the robot link can be executed only in the remote mode.
Remedy: Ensure that the remote condition is satisfied.

INTP–454 (%s^4, %d^5) Illegal return occurred

Cause: In the robot link, the type of a calling program differs from the type of a called program.
Remedy: Match the type of a calling program with the type of a called program.

INTP–455 (%s^4, %d^5) Group mismatch(Link pattern)

Cause: The master program motion group of the robot link does not match the master robot motion group specified with the link pattern.
Remedy: Match the master program motion group with the master robot motion group specified with the link pattern.

INTP–456 (%s^4, %d^5) Group mismatch(Slave group)

Cause: The slave program motion group of the robot link does not match the slave robot motion group specified with the slave group.
Remedy: Match the slave program motion group with the slave robot motion group specified with the slave group.

INTP–457 (%s^4, %d^5) Master tool number mismatch

Cause: The tool coordinate system number currently selected by the master robot does not match the robot link data master tool coordinate system number of the slave program.

Remedy: Match the tool coordinate system number currently selected by the master robot with the robot link data master tool coordinate system number of the slave program.

INTP–458 (%s^4, %d^5) Robot is still moving

Cause: The robots are moving with the robot link, so that the master/slave/single slave cannot establish synchronization.

Remedy: After the robot stops, restart the program.

INTP–459 (%s^4, %d^5) Slave cannot JOINT motion

Cause: The motion statement in the slave program of the robot link specifies a joint motion.

Remedy: Change the motion statement of the slave program to orthogonal motion.

INTP–460 (%s^4, %d^5) Cannot use JOINT pos for Slave

Cause: The position data format of the slave program of the robot link is the joint format.

Remedy: Change the position data format of the slave program to the orthogonal format.

INTP–461 (%s^4, %d^5) Master TP is enabled

Cause: The master program of the robot link was activated from the teach pendant. The slave program stops temporarily.

Remedy: The master program was activated from the teach pendant, so that the slave program stops temporarily.

INTP–462 (%s^4, %d^5) Cannot start Robot Link

Cause: The setting of the robot link may be incorrect.

Remedy: Check the setting.

INTP–463 (%s^4, %d^5) Motion group is Master

Cause: The motion group of a program whose execution was attempted with the robot link is master.

Remedy: Cancel the setting of master, then restart the program.

INTP–465 (%s^4, %d^5) Tracking error

Cause: The robot link could not perform synchronous motion.

Remedy: Check the setting of the robot link.

INTP–466 (%s^4, %d^5) Robot link not calibrated

Cause: The robot link is not calibrated.

Remedy: Calibrate the robot link.

INTP–467 (%s^4, %d^5) Cannot use INC for Slave

Cause: In a motion statement of the slave program of the robot link, an incremental instruction is taught.

Remedy: The incremental instruction cannot be used in a motion statement of the slave program.

INTP–468 (%s^4, %d^5) Cannot use OFFSET for Slave

Cause: In a motion statement of the slave program of the robot link, a compensation instruction is taught.

Remedy: The compensation instruction cannot be used in a motion statement of the slave program.

INTP–469 (%s^4, %d^5) BWD is failed for Master

Cause: An attempt for BWD synchronization of the mater of the robot link failed.

Remedy: Place the slave in the synchronization wait state.

INTP–470 (%s^4, %d^5) Not support BWD for Slav

Cause: BWD synchronization is not supported for the slave program of the robot link.

Remedy: BWD synchronization is supported for the slave program.

INTP–471 (%s^4, %d^5) Robot is Master(Manual)

Cause: In the robot link, the robot is placed in the master (manual) state.

Remedy: In the master (manual) state, external activation is disabled.
For external activation, set the master (single) state on the manual operation screen.

INTP–472 (%s^4, %d^5) Robot is Slave(Manual)

Cause: In the robot link, the robot is placed in the slave (manual) state.

Remedy: In the slave (manual) state, other slaves cannot be executed. Hold the program, and cancel the slave (manual) state.

INTP–474 (%s^4, %d^5) Synchro ID mismatch

Cause: In the robot link, a program with a synchronous motion ID different from the synchronous motion ID of the currently executed program was executed.

Remedy: Programs with different synchronous motion IDs cannot be executed at the same time.

INTP–475 (%s^4, %d^5) Cannot single step

Cause: The slave program of the robot link cannot be executed in the single step mode.

Remedy: Cancel the single step mode.

INTP–476 (%s^4, %d^5) BWD is failed

Cause: In the robot link, BWD failed.

Remedy: BWD failed.

INTP–477 (%s^4, %d^5) Cannot run Slave directly

Cause: The slave program of the robot link cannot be activated directly.

Remedy: Execute the slave by calling from the normal program.

INTP–478 This group can not be MASTER

Cause: This alarm is issued when an attempt is made to specify as a mater a robot not set as a master in the robot link or when an attempt is made to specify as a master a group not set as a master on the manual operation screen.

Remedy: Specify another group as a master, or modify the setting.

INTP–479 Bad Hostname or Address(MASTER

Cause: An attempt was made to execute the robot link when a host name not registered is specified or the setting of an IP address is incorrect on the host communication screen or the master setting screen.

Remedy: Check the master in the robot link setting and host communication setting.

INTP–480 Bad Hostname or Address(SLAVE)

Cause: An attempt was made to execute the robot link when a host name not registered is specified or the setting of an IP address is incorrect in the host communication setting or link pattern setting.

Remedy: Check the slave in the robot link setting and host communication setting.

INTP–481 Bad Synchronization ID

Cause: In the robot link, a program–specified synchronous motion ID is incorrect.

Remedy: Correct the synchronous motion ID on the list screen.

INTP–482 Bad Link Pattern Number

Cause: In the robot link, a program–specified link pattern number is incorrect.

Remedy: Correct the link pattern number on the list screen.

INTP–483 Bad Master Number

Cause: In the robot link, a program–specified master number is incorrect.

Remedy: Correct the master number on the list screen.

INTP–484 Bad Group number (MASTER)

Cause: The group number of the master of the robot link is incorrect.

Remedy: Check the group number of the master.

INTP–485 Bad Group number (SLAVE)

Cause: The group number of the slave of the robot link is incorrect.
Remedy: Check the group number of the slave.

INTP–486 SLAVE is not calibrated

Cause: In the robot link, there is a slave not calibrated.
Remedy: Calibrate the slave robot.

INTP–488 RLINK communication timeout

Cause: In the robot link, communication initialization timed out.
Remedy: Increase the value of $RK_SYSCFG.$RMGR_PHTOUT by 100.

INTP–489 Bad Hostname or Address, Group

Cause: An attempt was made to execute the robot link when the setting of a host name, IP address, or motion group is incorrect in the host communication setting or robot link setting.
Remedy: Check the robot link setting and host communication setting.

INTP–490 Timeout for link start

Cause: An attempt was made to execute the robot link when the setting of a host name, IP address, or motion group is incorrect in the host communication setting or robot link setting or when the robot link program is not executed at the communication destination. So, a synchronization start timeout occurred.
Remedy: Check the robot link setting and host communication setting, and also check the state of the robot at the communication destination.

INTP–491 Linked robot or comm stopped

Cause: During robot link execution, the robot at the communication destination stopped program execution, or stopped communication for a cause such as a power failure.
Remedy: Check the state of the robot at the communication destination.

INTP–493 Slave program stopped

Cause: During slave program execution, the master program at the communication destination stopped.
Remedy: Check the state of the robot at the communication destination.

INTP–493 Slave program stopped

Cause: During robot link execution as the master, the slave program at the communication destination stopped.
Remedy: Check the state of the robot at the communication destination.

JOG Error Codes (ID = 19)

JOG–001 WARN Overtravel Violation

Cause: A robot overtravel has occurred.
Remedy: While holding down the shift key, press the alarm clear button to clear the alarm. Then, while holding down the shift key, perform jog feed to move the overtravel axis into the movable range.

JOG–002 WARN Robot not Calibrated

Cause: Robot has not been calibrated.
Remedy: Apply one the following methods for positioning:
1 Make positioning settings on the positioning screen.
2 Turn the power off and then on again.

JOG–003 WARN No Motion Control

Cause: Other program has motion control
Remedy: Abort the program that has motion control by pressing FCTN key then selecting ABORT.

JOG–004 WARN Illegal linear jogging

Cause: You cannot do more than one rotational jog at a time.
Remedy: Only press one rotational jog key at a time.

JOG–005 WARN Can not clear hold flag

Cause: The hold key or hold button is held down. Or, *HOLD input is off.
Remedy: Release the hold key or hold button. Or, turn on *HOLD input.

JOG–006 WARN Subgroup does not exist

Cause: No extended axis exist in this group with which to jog.
Remedy: No action is required.

JOG–007 WARN Press SHIFT key to jog

Cause: The SHIFT key is not pressed.
Remedy: You must press the SHIFT key when jogging the robot. Release the JOG key then hold the SHIFT key and press the JOG key to jog.

JOG–008 WARN Turn on TP to jog

Cause: Teach pendant is not enabled.
Remedy: Hold the DEADMAN and turn on the teach pendant before jogging the robot.

JOG–009 WARN Hold deadman to jog

Cause: The DEADMAN switch is not pressed.
Remedy: Press the DEADMAN switch, then press RESET key to clear the error.

JOG–010 WARN Jog pressed before SHIFT

Cause: The JOG key was pressed before the SHIFT key was pressed.
Remedy: Release the JOG key. Then hold down the SHIFT key and press the JOG key.

JOG–011 WARN Utool changed while jogging

Cause: The selected tool frame changed while jogging.
Remedy: Release the SHIFT key and the JOG key. The new TOOL frame will take effect automatically.

JOG–012 WARN manual brake enabled

Cause: The manual brake enabled.
Remedy: Engage all the brakes by pressing EMERGENCY STOP button, then press the RESET key.

JOG–013 WARN Stroke limit (Group:%d Axis:%x Hex)

Cause: Robot axis reaches its specified stroke limit.
Remedy: The robot already reach the stroke limit and cannot jog in the current direction any more. Extend the axis limit if it does not exceed the robot and software specifications.

JOG–014 WARN Vertical fixture position

Cause: Robot reaches its vertical fixture position on the LR-MATE system.
Remedy: To continue jogging, release the JOG key then press it again.

JOG–015 WARN Horizontal fixture position

Cause: Robot reaches its horizontal fixture position on the LR-MATE system.
Remedy: To continue jogging, release the JOG key then press it again.

JOG–016 SERVO Softfloat time out(Group:%d)

Cause: Follow-up time is over when softfloat is ON.
Remedy: Make the system variable $SFLT_FUPTIM larger.

JOG–017 At R–Theta robot posture

Cause: In remote TCP jogging, the robot assumed the R–Theta attitude.
Remedy: To continue jogging, release the jog key, then press the jog key again.

JOG–020 Can not PATH JOG now

Cause: PATH JOG has selected, but robot is not currently on a taught path, or tool Z direction is same teaching path, so Y direction can not be determined.
Can not PATH JOG
Remedy: Use shift–FWD to execute program path, or specify another jog frame.

JOG–021 Multi key is pressed

Cause: Use of multiple jog keys is not supported in PATH JOG
Remedy: Use only one jog key at a time.

JOG–022 Disabled in JOINT path

Cause: PATH jog is disabled in JOINT path
Remedy: PATH jog is available in LINEAR and CERCULAR path

JOG–023 Available only in PAUSE

Cause: PATH jog is available only in PAUSE status
Remedy: PATH jog is available only in PAUSE status

JOG–024 Currently this key is invalid

Cause: This key is currently disabled.
Remedy: Change the jog coordinate system.

JOG–025 J4 is not zero

Cause: J4 is not at the 0° position.
Remedy: To use attitude–fixed jogging, J4 needs to be at the 0° position.

JOG–026 J4 is zero

Cause: J4 is now at the 0° position.
Remedy: Attitude–fixed jogging is enabled.

JOG–027 Reverse direction from J4=0

Cause: The direction of jogging is opposite to the 0° direction of J4.
Remedy: Press the jog key for the opposite direction.

JOG–028 Attitude fix mode limit (TCP)

Cause: A linear motion range limit was reached.
Remedy: Change the target position, or switch to joint motion. A stroke limit in the TCP mode was reached.

JOG–029 OFIX jog error

Cause: Internal error
Remedy: Contact your FANUC customer service representative, and inform the representative of an alarm message displayed.

JOG–030 Can't jog as OFIX

Cause: Attitude–fixed jogging is disabled. An additional alarm is issued.
Remedy: Check the additional alarm on the alarm history screen.

TPIF Error Codes (ID = 9)

TPIF–001 to 003 WARN Mnemonic editor error (%s^1)

Cause: Illegal case occurred on software.
Remedy: Contact our service center serving your locality.

TPIF–004 WARN Memory write error

Cause: The instruction cannot be used because the corresponding software option is not provided.
Remedy: Install the software option.

TPIF–005 WARN Program is not selected

Cause: The program was not selected when the program was displayed at the edit screen.
Remedy: Select a program in the SELECT screen.

TPIF–006 WARN SELECT is not taught

Cause: This taught statement needed the SELECT statement before the current line.
Remedy: Teach the SELECT statement before the current line.

TPIF–007 WARN Robot is not calibrated

Cause: The calibration was not finished yet
Remedy: Finish the calibration.

TPIF–008 WARN Memory protect violation

Cause: The program's write protection is set to on.
Remedy: Release protection on select screen.

TPIF–009 WARN Cancel delete by application

Cause: The program cannot be deleted because program deletion is disabled by the application tool software.
Remedy: Enable program deletion on the application setting screen.

TPIF–010 WARN Cancel enter by application

Cause: The program cannot be edited because program editing is disabled by the application tool software.
Remedy: Enable program editing on the application setting screen.

TPIF–011 WARN Item is not found

Cause: Item is not found below this line
Remedy: Try another item or close search function

TPIF–012 WARN Kinematics solution is invalid

Cause: Can not translate position data
Remedy: Check the configuration of robot and $MNUTOOL/$MNUFRAM of system variables

TPIF–013 WARN Other program is running

Cause: Can not select the program when other is running or pausing.
Remedy: Select program after aborting the program which is running or pausing.

TPIF–014 WARN Teach pendant is disabled

Cause: Can not be edit a program when the Teach pendant is disabled.
Remedy: Edit program after Teach pendant is enabled.

TPIF–015 WARN Bad position register index

Cause: The specified position register index is invalid.
Remedy: Check the index of the position register.

TPIF–016 to 017 WARN Memory access failed (%s^1)

Cause: Illegal case occurred on software.
Remedy:
1 Select a program again.
2 Contact FANUC Robotics.

TPIF–018 WARN Unspecified index value

Cause: Specified index value is invalid.
Remedy: Check specified index value.

TPIF–019 WARN This item cannot be replaced

Cause: This item can not be replaced.
Remedy: Try another item or close replace function.

TPIF–020 NONE Mnaction search error

Cause: Illegal case occurred on software.
Remedy: Contact our service center serving your locality.

TPIF–023 WARN WJNT and RTCP are not compatible

Cause: Wjnt and RTCP are not compatible
Remedy: Remove Wjnt or RTCP before add the other

TPIF–030 WARN Program name is NULL

Cause: Program name was not entered.
Remedy: Enter program name.

TPIF–031 WARN Remove num from top of Program name

Cause: Top of program name is number.
Remedy: Remove number from top of program name.

TPIF–032 WARN Remove space from Program name

Cause: A space is included in the program name.
Remedy: Remove space from program name.

TPIF–033 WARN Remove comma from Program name

Cause: A comma is included in the program name.
Remedy: Remove comma from program name.

TPIF–034 WARN Remove dot from Program name

Cause: A dot is included in the program name.
Remedy: Remove dot from program name.

TPIF–035 WARN Remove minus from Program name

Cause: A minus is included in the program name.
Remedy: Remove minus from program name.

TPIF–036 WARN Memory is not enough

Cause: Not enough memory available.
Remedy: Delete unused programs.

TPIF–037 WARN Program must be selected by TP

Cause: Only the Teach Pendant default program can be edited on the CRT
Remedy: Please select the program on the Teach Pendant before editing on the CRT

TPIF–038 WARN Invalid char in program name

Cause: Invalid character in program name
Remedy: Please remove invalid character from program name

TPIF–040 WARN Label is already exist

Cause: Same label No. already exists.
Remedy: Change to different label No.

TPIF–041 WARN MNUTOOLNUM number is invalid

Cause: Specified MNUTOOLNUM number is invalid.
Remedy: Check MNUTOOLNUM number in SYSTEM variables.

TPIF–042 WARN MNUFRAMENUM number is invalid

Cause: Specified MNUFRAMNUM number is invalid.
Remedy: Check MNUFRAMNUM number in SYSTEM variables.

TPIF–043 WARN External change is valid

Cause: Can not change robot (group), because the function that select robot by external DI is valid.
Remedy: Set $MULTI_ROBO.CHANGE_SDI in SYSTEM variables to ZERO.

TPIF–044 WARN Program is unsuitable for robot

Cause: The group mask of program differs from selected robot (group).
Remedy: Check to select robot (group) or check group mask of program attribute.

TPIF–045 WARN Pallet number is over max

Cause: Palletizing instruction can not teach more than 16 in one program.
Remedy: Teach another program.

TPIF–046 WARN Motion option is over max

Cause: Too many motion option of default motion
Remedy: Please decrease motion option of default motion

TPIF–047 WARN Invalid program is selected

Cause: The program type is wrong.
Remedy: Select TPE program.

TPIF–048 WARN Running program is not found

Cause: The running program does not exist.

TPIF–049 WARN Port number is invalid

Cause: Port is not set for outside device.
Remedy: Set port for outside device.

TPIF–050 WARN Macro does not exist

Cause: A program is not assigned to this macro command.
Remedy: Assign a program to this macro command.

TPIF–051 WARN Program has been selected by PNS

Cause: When a program has been selected by PNS, you can not select program at SELECT screen.
Remedy: You have to turn off the signal of PNSTROBE.

TPIF–052 WARN FWD/BWD is disabled

Cause: When the Disabled FWD function has been selected, you can not execute the program by TP
Remedy: Please select the Disabled FWD in the function menu, then you can release it from the Disable FWD function.

TPIF–053 WARN Not editing background program

Cause: The program has not been selected by the BACKGROUND editing
Remedy: Please select the BACKGROUND program in the SELECT screen

TPIF–054 WARN Could not end editing

Cause: Memory is not enough or background program is invalid
Remedy: Delete any useless programs or confirm the background program

TPIF–055 WARN Could not recovery original program

Cause: Failed recovering original program which has been selected by the BACKGROUND
Remedy: End editing by the END_EDIT of [EDCMD] again before executing the origianl program which has been selected by the BACKGROUND

TPIF–056 WARN This program is used by the CRT

Cause: The program of BACKGROUND can not be selected by the CRT and TP at the same time
Remedy: End editing by the END_EDIT of [EDCMD] at the CRT

TPIF–057 WARN This program is used by the TP

Cause: The program of BACKGROUND can not be selected by the CRT and TP at the same time
Remedy: End editing by the END_EDIT of [EDCMD] at the TP

TPIF–060 WARN Can't record on cartesian (Group:%d)

Cause: This current position is in singularity
Remedy: You can record this position on joint type only Please select the function key

TPIF–061 WARN Group[%s] has not recorded

Cause: This position data has not been changed to displayed groups because you selected the function key which did not record the position, when checking in singularity
Remedy: Check this recorded position again before excution

TPIF–062 AND operator was replaced to OR

Cause: All AND operators on this line were replaced with OR operators.
Remedy: You cannot mix AND and OR operator on a the same line.
Verify that all logical operators on this line are the same before execution.

TPIF–063 OR operator was replaced to AND

Cause: All OR operator on this line were replaced by AND operators.
You cannot mix AND OR operaotr on a the same line
Remedy: Verify all logical operators on this line before execution

TPIF–064 Too many AND/OR operator(Max.4)

Cause: Too many AND/OR operators (Max.4 on a single line)
Remedy: Teach the logical operation on another line

TPIF–065 Arithmetic operator was unified to +– or */

Cause: Arithmetic operator on this line was changed to +– or */.
Cannot mix arithmatic + and – operators with * and /operators on the same line.
Remedy: Verify all arithmetic operators on this line before execution

TPIF–066 Too many arithmetic operator(Max.5)

Cause: Too many arithmetic operators (Max.5 on a single line)
Remedy: Teach the arithmetic operation on another line

TPIF–067 Too many arguments (Max.10)

Cause: Too many arguments (Max.10 for a program or a macro)
Remedy: Check arguments of the program/macro

TPIF–070 Cannot teach the instruction

Cause: Cannot teach the instruction.
Remedy: Check the sub type of the program.

TPIF–071 Cannot change sub type

Cause: Cannot change sub type
Remedy: Check sub type of the program

TPIF–072 Cannot change motion group

Cause: Cannot change motion group
Remedy: Check sub type of the program

TPIF–090 WARN This program has motion group

Cause: The program specified in $PWR_HOT, $PWR_SEMI and $PWR_NORMAL must not have motion group.
Remedy: Set * to all motion group in program detail screen on TP.

TPIF–091 WARN PREG access error

Cause: An error occurred when accessing a position register.
Remedy: Refer to the error cause code.

TPIF–092 Value %d expected %s

Cause: The value_array that was passed to a KAREL built–in was incorrectly specified.
Remedy: Make sure the value_array specifies the correct names for the variables and that the types expected are correct.

TPIF–093 USER menu must be selected

Cause: Software internal error.
Remedy: Consult our service representative.

TPIF–094 USER2 menu must be selected

Cause: Software internal error.
Remedy: Consult our service representative.

TPIF–095 WARN Execution history table error

Cause: Software internal error
Remedy: Please do controlled start(it isn't necessary to re–set the new item)

TPIF–097 WARN Running task's history can't display

Cause: The execution history of the executing program can not be displayed
Remedy: Please refer this screen when the program is paused or aborted

TPIF–098 WARN %s was not run

Cause: The program of $PWR_HOT, $PWR_SEMI or $PWR_NORMAL is not executed
Remedy: Read the cause code

TPIF–099 WARN This program is edited

Cause: The program specified in $PWR_HOT, $PWR_SEMI and $PWR_NORMAL is not executed, when the program is in editing.
Remedy: Select the other program

TPIF–100 WARN No vacant table space

Cause: Illegal case occured on software.
Remedy: Contact our service center serving your locality.

TPIF–101 WARN No such menu

Cause: Illegal case occured on software.
Remedy: Contact our service center serving your locality.

TPIF–102 WARN E.STOP is asserted

Cause: FWD execution is selected while, E.STOP is asserted.
Remedy: Turn the E.STOP off. Then select FWD execution

TPIF–103 WARN Dead man is released

Cause: When starting the program with the teach pendant, the deadman switch was released.
Remedy: Press and hold the deadman switch and start a program.

TPIF–104 WARN Teach Pendant is disabled

Cause: A program was not started because the teach pendant was disabled.
Remedy: After turning on the enable switch of the teach pendant, start a program.

TPIF–105 WARN Program is not selected

Cause: A program was started without selecting a program.
Remedy: After selecting a program, start the program.

TPIF–106 WARN Program is already running

Cause: While a program was running, starting from teach pendant was performed.
Remedy: Start a program after waiting for program's ending or aborting it.

TPIF–107 WARN FWD/BWD is disabled

Cause:
1. Starting a program was performed when the starting was prohibited such as entering the value into the message line.
2. A program was not selected.

Remedy:
1. After finishing the procedure of entering the value, start a program.
2. Select a program and then start a program.

TPIF–108 WARN Form error, line %d, item %d

Cause: The Form Manager detected an error on the specified line with the specified item.
Remedy: Refer to the cause code for the actual error.

TPIF–109 WARN %v not specified correctly

Cause: An internal software error occurred.
Remedy: Contact your FANUC Service Center.

TPIF–110 WARN Screen used by other device

Cause: An internal software error occurred.
Remedy: Contact your FANUC Service Center.

TPIF–111 op_global does not exist

Cause: Internal system error.
Remedy: Consult our service representative.

TPIF–112 op_sel does not exist

Cause: Internal system error.
Remedy: Consult our service representative.

TPIF–113 Illegal param in op menu

Cause: Internal system error.
Remedy: Consult our service representative.

TPIF–114 Illegal data in op menu

Cause: Internal system error.
Remedy: Consult our service representative.

TPIF–115 Data is full

Cause: Internal system error.
Remedy: Consult our service representative.

TPIF–116 System variable error: %s

Cause: System variable name is invalid
Remedy: Check the spelling and format of the name.

TPIF–117 Cannot backup to device: %s

Cause: The default device is not valid for backup
Remedy: Select a valid device and try again

TPIF–118 File error for %s

Cause: File error
Remedy: Perform a cold start:
1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that

TPIF–119 File compression failed

Cause: Failed creating compressed file
Remedy: Check backup device

TPIF–120 Device failure

Cause: Device failure
Remedy: Check device and try again

TPIF–121 Invalid copy. Use MOVE key.

Cause: Cannot COPY a file on a Memory device to the same Memory device.
Remedy: Use the MOVE key and try again

TPIF–122 Specified softpart ID is illegal

Cause: Internal system error.
Remedy: Consult our service representative.

TPIF–123 No active applications

Cause: Internal system error.
Remedy: Consult our service representative.

TPIF–124 Current application is nothing

Cause: Internal system error.
Remedy: Consult our service representative.

TPIF–125 Specified softpart ID is nothing

Cause: Internal system error.
Remedy: Consult our service representative.

TPIF–126 THKY ASLOAD is failed

Cause: Internal system error.
Remedy: Consult our service representative.

TPIF–127 TOPK ASLOAD is failed

Cause: Internal system error.
Remedy: Consult our service representative.

TPIF–128 Verify logic of pasted line(s)

Cause: The reverse motion copy function does not support the following motion option instruction.

1 Application command
2 Skip, Quick Skip
3 Incremental
4 Continuous turn
5 Ahead execution command

Remedy: Check the above motion option instruction. And modify the copied statement correctly.

TPIF–129 Group motion inst. is pasted

Cause: The group motion instruction is copied. The reverse motion copy function does not supported group motion instruction.
Remedy: Check the group motion instruction. And modify the copied statement correctly.

TPIF–130 Specified application has no EQ

Cause: Internal system error.
Remedy: Consult our service representative.

TPIF–131 Please set application mask data

Cause: This program has no application mask
Remedy: Please set the application mask in the program detail screen

TPIF–132 Can't recover this operation

Cause: Because the data for UNDO can not be saved, this operation can not recover by UNDO function
Remedy: Check the cause code. If the memory is full, please delete program or disable UNDO function.

TPIF–133 Can’t recover this command

Cause: Palletizing command and Compliance control command can not be recovered by UNDO function

MOTN Error Codes (ID = 15)

MOTN–001 to 008 STOP.G Internal error in osmkpkt

Cause: Internal system error.
Remedy: Cycle start the controller

MOTN–009 STOP.G Internal error for single step

Cause: The tool stopped at the midpoint of an arc in single step mode.
Remedy: Ignore this alarm.

MOTN–010 to 011 STOP.G Internal error in osathpkt

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–012 STOP.G Invalid softpart MIR

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–013 STOP.G Invalid softpart SEG

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–017 STOP.G Limit error (Group:%d^2, Axis:%x^3 Hex)

Cause: The specified position falls outside the joint movable range ($PARAM_GROUP.$LOWERLIMS, $PARAM_GROUP.$UPPERLIMS).
Axis j is defined in hexadecimal, as shown below.
Axis 1: 1, Axis 2: 2, Axis 3: 4, Axis 4: 8, Axis 5: 10, Axis 6: 20, Axis 7: 40, Axis 8: 80, Axis 9: 100
If two or more axes have caused this alarm, the total of their values, shown above, is indicated in hexadecimal.

Example	Axis 1 +	Axis 3 +	Axis 4 +	Axis 6 +	Axis 9	=	12D
	1	4	8	20	100		

Remedy:
1 Correct the position so that it falls within the movable range.
2 Change the movable range settings on the joint movable range screen, which is displayed by selecting 6 SYSTEM AXIS LIMITS.

MOTN–018 STOP.G Position not reachable

Cause: The position is not reachable or is near a singularity point.
Remedy: Reteach the position that is not reachable.

MOTN–019 WARN In singularity

Cause: The position is near a singularity point.
Remedy: Reteach the position that is near a singularity point.

MOTN–020 WARN Wristjoint warning

Cause: Wrist joint warning
Cause: Wrist joint warning

MOTN–021 STOP.G No kinematics error

Cause: No kinematics.
Remedy: Use joint motion.

MOTN–022 STOP.G Invalid limit number

Cause: Invalid limit number.
Remedy: Set limit number correctly.

MOTN–023 STOP.G In singularity

Cause: The position is near a singularity point.
Remedy: Reteach the position that is near a singularity point.

MOTN–024 STOP.G Kinematics not defined

Cause: Kinematics is not defined.
Remedy: Define Kinematics.

MOTN–030 to 046 STOP.G Internal error in MMGR:PEND

Cause: Internal system error.
Remedy: Contact our service center serving your lacality.

MOTN–047 Internal error in MMGR:PRST

Cause: Internal error
Remedy: Contact your FANUC customer service representative, and inform the representative of an alarm message displayed.

MOTN–049 STOP.G Attempt to move w/o calibrated

Cause: Robot not calibrated.
Remedy: Calibrate the robot.

MOTN–050 STOP.G Invalid spdlim (Group:%d^2 Axis:%x^3 H)

Cause: An internal software error occurred. The joint speed factor ($PARAM_GROUP.$SPEEDLIMJNT) is invalid. Axis j is defined in hexadecimal, as shown below.
Axis 1: 1, Axis 2: 2, Axis 3: 4, Axis 4: 8, Axis 5: 10, Axis 6: 20, Axis 7: 40, Axis 8: 80, Axis 9: 100
If two or more axes have caused this alarm, the total of their values, shown above, is indicated in hexadecimal.

Example	Axis 1	+	Axis 3	+	Axis 4	+	Axis 6	+	Axis 9	= 12D
	1		4		8		20		100	

Remedy: Correct the joint speed factor.

MOTN–051 to 53 STOP.G Speed out of range (Group:%d^2)

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–054 STOP.G Uninitialized dest pos (Group:%d^2)

Cause: Uninitialized destination position.
Remedy: Teach the destination position.

MOTN–055 STOP.G Uninitialized via pos (Group:%d^2)

Cause: Uninitialized via position.
Remedy: Teach the via position.

MOTN–056 WARN Speed limits used (Group:%d^2)

Cause: Speed limits used.
Remedy: This is just a notification. You do not have to do anything for this warning message.

MOTN–057 to 062 STOP.G Invalid mir (Group:%d^2)

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–063 STOP.G Position config change (Group:%d^2)

Cause: For path–controlled operation (linear or circular operation), different position data formats are set for the start point and end point.
Remedy:
1 Set the same position data format for the start point and end point.
2 Specify joint operation mode.
3 Specify a wrist joint operation instruction (operation addition instruction).

MOTN–064 and 065 STOP.G Rs orientation error (Group:%d^2)

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–066 STOP.G Degenerate circle (Group:%d^2)

Cause: For circular operation, the position data for the start point, passing point, and end point is invalid.
a Two of the start, passing, and end points overlap one another.
b All of the start, passing, and end points are arranged in a straight line.
Remedy: Specify appropriate start, passing, and end points for circular operation.

MOTN–067 to 072 STOP.G Ata2 error in circle (Group:%d^2)

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–073 STOP.G Error in orientype (Group:%d^2)

Cause: Internal error: planner received invalid orientype.
Remedy: Contact our service center serving your locality.

MOTN–074 to 079 STOP.G Error in speed (Group:%d^2)

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–080 STOP.G Via position required (Group:%d^2)

Cause: Missing via position for circular motion.
Remedy: Teach via position.

MOTN–081 STOP.G Extended position error (Group:%d^2)

Cause: No value is set for the additional axis.
Remedy: Set a value for the additional axis.

MOTN–082 to 087 STOP.G (Group:%d^2)

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–088 STOP.G Not cartesian move (Group:%d^2)

Cause: Motype is not cartesian.
Remedy: Must set motype to cartesian.

MOTN–089 to 091 STOP.G (Group:%d^2)

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–092 STOP.G Extended not supported (Group:%d^2)

Cause: Extended axes not supported
Remedy: Do not use extended axes.

MOTN–093 and 094 STOP.G Internal (Group:%d^2)

Cause: Internal plan error
Remedy: Contact our service center serving your locality.

MOTN–095 WARN Can't blend corner line:%d^5

Cause: Under acceleration vector control, the specified operation instruction results in an inconstant robot path.
Remedy: 1 Turn off acceleration vector control.
2 Correct the operation instruction so that it can be executed normally.

MOTN–096 STOP.G Cart rate not equal(Group:%d^2)

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–097 WARN INTR overrun %d^3 (Group:%d^2)

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–098 to 109 STOP.G INTR (Group:%d^2)

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–110 STOP.G Use FINE in last L (Group:%d^2)

Cause: During the execution of the specified operation instruction, joint operation could not be performed.
Remedy: Correct the operation instruction, according to the desired path–controlled operation.

MOTN–111 WARN Can’t switch filter(Group:%d^2)

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–112 Increment move turn Mismatch

Cause: Incremental motion causes turn number mismatch
Remedy: Change position to absolute position

MOTN–113 WARN Robot not calibrated

Cause: Robot not calibrated.
Remedy: Calibrate the robot.

MOTN–114 WARN Servo is on (Group:%d^2)

Cause: Servo in still on.
Remedy: Turn off servo.

MOTN–115 WARN Invalid brake mask (Group:%d^2)

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–116 WARN Invalid solution (Group:%d^2)

Cause: Invalid kinematics solution.
Remedy: Reteach position.

MOTN–117 WARN Robot not mastered (Group:%d^2)

Cause: Robot not mastered.
Remedy: Master the robot.

MOTN–118 WARN Robot in over travel (Group:%d^2)

Cause: Robot in overtravel.
Remedy: Reset over travel jog the robot outside of the overtravel position.

MOTN–119 WARN Servo is off (Group:%d^2)

Cause: Robot servo is on.
Remedy: Turn off servo power.

MOTN–120 to 121 WARN Invalid reference position (Group:%d^2)

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–122 STOP.G Dfilter not empty (Group:%d^2)

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–123 WARN Not enough node (Group:%d^2)

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–124 to 127 STOP.G INTR:Bad Mirpkt req_code(Group:%d^2)

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–128 STOP.G Group mtn not supported(Group:%d^2)

Cause: Group motion not supported.
Remedy: Document the events that led to the error and contact our service center serving your locality.

MOTN–129 and 130 STOP.G Local cond ptr conflict(Group:%d^2)

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–131 STOP.G In singularity

Cause: Position near by a singularity point.
Remedy: a. Move the target point well away from the singular point.
b. Use joint coordinates to specify the target point in joint operation mode.

MOTN–132 STOP.G Group circ not supported(Group:%d^2)

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–133 WARN Time after limit used(Group:%d^2)

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–134 STOP.G Can not move path backward (Group:%d^2)

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–135 STOP.G Last motype can’t be circular (Group:%d^2)

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–136 STOP.G Illegal filter switch line:%d^5

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–137 STOP.G No circular softpart (Group:%d^2)

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–138 STOP.G No joint short motion SP (Group:%d^2)

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–139 STOP.G No cart short motion SP (Group:%d^2)

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–140 STOP.G No KAREL motion softpart (Group:%d^2)

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–141 STOP.G No KAREL motion func. ptr (Group:%d^2)

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–142 STOP.G No Group Motion SP (Group:%d^2)

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–143 STOP.G No Motion Resume SP (Group:%d^2)

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–144 STOP.G No joint Turbo Move SP (Group:%d^2)

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–145 STOP.G No cart Turbo Move SP (Group:%d^2)

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MOTN–146 STOP.G INTR can't replan major axis(Group:%d^2)

Cause: Mismatch in major axis turn number.
Remedy: Reteach position.

MOTN–147 WARN L–>J replan joint slowdown (Group:%d^2)

Cause: Linear motions ignore turn numbers. Therefore, when a joint motion follows several linear motions, the turn number might be mismatched, causing the robot to slow down
Remedy: Change the current motion's motype to linear or change the previous motion's motype to joint. If the problem persists, re-teach the path.

MOTN–148 SWARN Can't move concurrently (Group:%d^2)

Cause: Two motion groups cannot synchronize with each other due to replanning of one group. This will cause slow down on both groups.
Remedy: If slow down is not acceptable, re-teach the path.

MOTN–149 STOP.G CF:rotspeedlim exceeded line:%d^5

Cause: CF:rotspeedlim exceeded.
Remedy:
1 Set system variable $cf_paramgp[].$cf_framenum to 1 or 2. Turn the power off and then on again.
2 Reduce the speed.
3 Specify positioning as the previous operation.

MOTN–161 (%s^4 L:%d^5) Can't look ahead

Cause: With the shortest time control function, program lines cannot be read in advance.
The following situations can be considered:
– The position register is used without locking.
– The if/selection instruction is used.
– Another program is called.

Remedy:
– Use the lock position register instruction.
– Remove the if/selection instruction.
– Integrate the programs into one.

MOTN–171 Overload

Cause: An overload is imposed.
Remedy: Reduce the load.

MOTN–172 Another robot is re–linked

Cause: The robot at a link destination was relinked, so that operation stopped.
Remedy: Stop all linked robots once, then restart the robots.

MOTN–173 Robot link configuration error

Cause: The robot link setting is incorrect.
Remedy: Check the host name and IP address in the host communication setting, and also check the robot link setting.

MOTN–174 No motion control

Cause: This alarm is issued, for example, when an operation for master or slave setting such as manual operation screen manipulation or program execution is performed while the robot is moving according to a program or jog operation.
Remedy: Perform an operation after the robot stops.

MOTN–175 Failed to be MASTER

Cause: At the time of switching to the master state by program execution or manual operation, robot motion is not completed, or the setting is incorrect.
Remedy: Modify the program, or check the robot link setting.

MOTN–176 Failed to be SLAVE

Cause: At the time of switching to the master state by program execution or manual operation, robot motion is not completed, or the setting is incorrect.
Remedy: Modify the program, or check the robot link setting.

MOTN–177 Failed to end sync motion

Cause: If the master and slave have not stopped or the setting is incorrect, synchronous motion cannot be completed.
Remedy: Check the motion instruction of the program, and the robot link setting.

MOTN–178 Link robot is HELD

Cause: After start of synchronous motion, it was detected that the robot at the communication destination lost synchronism for a cause such as program termination.
Remedy: The programs temporarily stop. Restart the programs of the master and slave.

MOTN–179 Robot link internal error

Cause: Internal error
Remedy: Contact your FANUC customer service representative, and inform the representative of an alarm message displayed.

MOTN–180 Robot link Calib–data not found

Cause: Calibration data cannot be found.
Remedy: Calibrate the robot link.

MOTN–181 Robot link Version mismatch

Cause: The robot link software differs between the master robot and slave robot.
Remedy: Match the software version of the master robot with that of the slave robot.

MOTN–182 Failed to get data from master

Cause: Communication data is not sent from the master robot.
Remedy: Check the Ethernet cable, cable connection, hub, main board, and robot link setting.

MOTN–184 Invalid MNUTOOL data array

Cause: The value of the system variable $MNUTOOL is invalid.
Remedy: Check the value of the system variable $MNUTOOL.

MOTN–185 Protect of ACK BF to be sent

Cause: The memory in the slave robot for communication from the slave robot to the master robot is protected.
Remedy: No particular action is required.

MOTN–186 Protect of BCST BF to be sent

Cause: The memory in the master robot for communication from the master robot to the slave robot is protected.
Remedy: No particular action is required.

MOTN–187 Protect of ACK BF to be read

Cause: The memory in the master robot for communication from the slave robot to the master robot is protected.
Remedy: No particular action is required.

MOTN–188 Protect of BCST BF to be read

Cause: The memory in the slave robot for communication from the master robot to the slave robot is protected.
Remedy: No particular action is required.

MOTN–189 Slave motion remained

Cause: In the slave robot, the amount of travel of the previous motion remains before the slave program is started.
Remedy: Restart after the previous motion is completed.

MOTN–190 Slave cannot use JOINT pos

Cause: The motion instruction data of the slave robot is in the joint format.
Remedy: Change the data to the orthogonal format.

MOTN–191 Slave cannot JOINT motion

Cause: The slave program cannot make a joint motion.
Remedy: Change the instruction to an orthogonal motion instruction.

MOTN–192 UT of MASTER was changed

Cause: In the master state, the tool coordinate system of the master robot was changed.
Remedy: Do not change the tool coordinate system in the master state.

MOTN–193 UT of SLAVE was changed

Cause: In the slave state, the tool coordinate system of the slave robot was changed.
Remedy: Do no change the tool coordinate system in the slave state.

MOTN–194 Machine Lock is ENABLED

Cause: In the master lock state, synchronous motion is disabled.
Remedy: Cancel the machine lock state.

MOTN–195 RLINK internal error %d^5

Cause: Internal error
Remedy: Contact your FANUC customer service representative, and inform the representative of an alarm message displayed.

MOTN–198 CRC Start–Via too close(L:%d^5)

Cause: The start point and center point of an arc are too close to each other.
Remedy: Reteach the robot. The taught points of an arc must be on the same plane. Otherwise, a minor modification to the teaching can cause a major change in motion.

MOTN–199 CRC Via–Dest too close(L:%d^5)

Cause: The intermediate point and end point of an arc are too close to each other.
Remedy: Reteach the robot. The taught points of an arc must be on the same plane. Otherwise, a minor modification to the teaching can cause a major change in motion.

MOTN–200 (%s^4, %d^5) Too long anticipate time

Cause: The value of advanced processing time (Timebefore) is too large.
Remedy:

– Reteach the previous taught point to increase the distance of motion.
– Decrease the advanced processing time.

MOTN–230 T1 rotspeed limit (G:%d^2)

Cause: The attitude change speed in the T1 mode was clamped.

Remedy: Decrease the speed of teaching. Alternatively, use a motion instruction in deg/sec or sec.

MOTN–231 T1 Speed restriction (G:%d^2)

Cause: When the taught speed is 250 mm/sec or less in the T1 mode, speed restriction processing was performed.

Remedy: The speed of the flange section exceeded 250 mm/sec because of a change in the tool attitude. This alarm is a warning, and does not represent a failure. However, the actual motion of this portion needs to be checked in the T2 mode.

MOTN–240 J4 is not zero

Cause: The J4–axis is not at the 0° position.

Remedy: Make a motion so that the J4–axis is at the 0° position.

MOTN–241 OFIX stroke limit

Cause: In attitude–fixed motion, a stroke limit was detected.

Remedy: Check the motion range, and reteach the robot so that the motion does not exceed the range.

MOTN–242 OFIX is disabled

Cause: The attitude–fixed motion instruction is disabled.

Remedy: Check if the robot supports attitude–fixed motion.

MOTN–243 OFIX error

Cause: The attitude–fixed motion instruction cannot be executed for another cause.

Remedy: Check the alarm history to see if other alarms are issued.

MOTN–244 OFIX Detect J4 is not 0

Cause: J4 at the motion start position or target position is not at the 0° position.

Remedy: Check the value of J4 at each position, and make modifications.

MOTN–245 OFIX Wrist config mismatch

Cause: The configuration differs between the motion start position and target position.

Remedy: Check the attitude and make modifications. If motion is still unsuccessful, use joint motion.

MOTN–246 OFIX Invalid rail vector

Cause: The attitude–fixed motion instruction is disabled.

Remedy: Check if the robot supports attitude–fixed motion.

MOTN–247 E–Effector is not vertical to rail

Cause: The flange surface is not parallel with the J1–axis.

Remedy: Cause the robot to assume such an attitude that the flange surface is parallel with the J1–axis.

MOTN–248 OFIX Too large tool rotation

Cause: In attitude–fixed motion, the rotation angle of the flange between the start point and end point exceeded the range allowable in one motion.

Remedy: Split the motion, and teach each split part of the motion.

MOTN–249 OFIX Too large tool spin

Cause: In attitude–fixed motion, the rotation angle of J6 between the start point and end point exceeded the range allowable in one motion.

Remedy: Split the motion, and teach each split part of the motion.

MOTN–250 Use CNT0/FINE for L/C before OFIX

Cause: A circular motion or a linear motion other than attitude–fixed motion continues to an attitude–fixed motion through a smooth motion.

Remedy: Change the positioning mode of the circular motion or linear motion to smooth 0 or positioning.

MOTN–251 Can't use OFIX with this motion

Cause: A motion statement, such as an incremental motion instruction or remote TCP, which cannot be used at the same time with the attitude fix instruction, is specified.

Remedy: Modify the instruction.

MOTN–252 OFIX: No plan data

Cause: Internal error

Remedy: Contact your FANUC customer service representative, and inform the representative of an alarm message displayed.

MOTN–253 OFIX: Motion type mismatch

Cause: Internal error

Remedy: Contact your FANUC customer service representative, and inform the representative of an alarm message displayed.

MOTN–254 OFIX: Detect large spin

Cause: The tool attitude change per motion is too large.

Remedy: Split the motion, and teach each split part of the motion.

MOTN–255 OFIX: Detect J4 is not 0

Cause: During motion, it was detected that J4 is not at the 0° position.
This alarm is issued because J4 is slightly shifted from the 0° position at the start point or end point of motion, or the angular change of the flange per motion is too large.

Remedy: Check the values of the start point and end point of J4.
If this alarm is issued even when J4 is taught to assume exactly the 0° position at both points, split the motion statement.

MOTN–256 OFIX: TCP config limit

Cause: A linear motion range limit was reached.

Remedy: Change the target position, or switch to joint motion.

MOTN–257 Wrist start angle mismatch

Cause: The motion start angle of the wrist axis does not match the internal calculation for attitude–fixed motion.

Remedy: Modify the teaching only so that the J4–axis moves completely with 0 degree.
Moreover, check if an application such as for tracking that compensates for a motion is used and check also if a motion addition instruction such as for compensating for a track is used.

MOTN–258 Not reached to dest rotation

Cause: At the end of attitude–fixed motion, the tool arrival attitude does not match the internal calculation for attitude–fixed motion.

Remedy: Modify the teaching only so that the J4–axis moves completely with 0 degree.
Moreover, check if an application such as for tracking that compensates for a motion is used and check also if a motion addition instruction such as for compensating for a track is used.

MOTN–259 Not reached to dest spin

Cause: At the end of attitude–fixed motion, the tool arrival attitude does not match the internal calculation for attitude–fixed motion.

Remedy: Modify the teaching only so that the J4–axis moves completely with 0 degree.
Moreover, check if an application such as for tracking that compensates for a motion is used and check also if a motion addition instruction such as for compensating for a track is used.

MOTN–300 CD not support:Use CNT L:%d^5

Cause: Term type CD is not supported.

Remedy: Change termtype FINE or CNT.

MOTN–301 Path to resume is changed(G:%d^2)

Cause: Can't resume motion.

Remedy: Abort and run program.

MOTN–302 Corner speed slowdown L:%d^5

Cause: Corner speed slows down automatically because of robot constraint.
Remedy: If slow down is not acceptable, re–teach the path.

MOTN–303 Can't maintain CDist L:%d^5

Cause: Can't maintain corner distance because path is short or speed is high.
Remedy: Lengthen path or reduce speed.

MOTN–304 CS:Prog speed achieved L:%d^5

Cause: SPD value does not affect corner speed anymore.
Remedy: This is just a notification. You do not have to do anything for this warning messsage.

MOTN–305 Can't maintain speed L:%d^5

Cause: Can't maintain program speed on the path because of robot constraint.
Remedy: This is just a notification. You do not have to do anything for this warning messsage.

MOTN–306 Can't replan (G:%d^2, A:%x^3 Hex)

Cause: Resume motion cannot reach stop position
Can't resume orginal path.
Remedy: Abort program and rerun

MOTN–307 Mismatch MMR (G:%d^2)

Cause: Internal error
Remedy: Contact your FANUC customer service representative, and inform the representative of an alarm message displayed.

MOTN–308 FINE termtype used L:%d^5

Cause: Can't generate corner between two motion because of motion instruction.
And CNT or CD is ignored.
Remedy: Use LOCK PREG instructiion when PR[] is used for positiion or OFFSET instruction is used.

MOTN–309 Circular speed reduced L:%d^5

Cause: Circular speed is reduced because of robot contraint
Remedy: Reduce program speed not to display.

MOTN–310 Pos. Cfg. change 2 (G:%d^2)

Cause: Configuration mismatch
Remedy: string matches the start position's configuration string.

MOTN–311 Path to resume is changed(G:%d^2)

Cause: Can't resume motion on the original path.
Remedy: Abort and run program.
Then, the resumed motion may not be on the original path.

MOTN–312 Can't resume in single step CJ

Cause: Can't resume motion in single step mode.
Remedy: Abort program and rerun.

MOTN–313 Can't resume motion CJ(2)

Cause: Can't resume motion on the original path.
Remedy: Abort and run program.
Then, the resumed motion may not be on the original path.

MOTN–314 Can't resume motion CJ(3)

Cause: Can't resume motion on the original path due to motion condition.
Remedy: Abort and run program.
Then, the resumed motion may not be on the original path.

MOTN–315 Command speed is changed CJ

Cause: Can't resume motion on the original path due to command speed change.
Remedy: Modify back the command speed, or abort program

MOTN–316 Override change not allowed

Cause: An override change was made when CJP was disabled and the program was being restarted.
Remedy: Make an override change before restarting the program. Do not make an override change immediately after the program is restarted.

MOTN–319 CRC large orient change (G:%d^2)

Cause: The small circle causes a large attitude change.
Remedy: Reteach the robot.

MOTN–320 Adj out of limit at line %s

Cause: During a fine adjustment check, a position that cannot be reached was detected.
Remedy: From the alarm message, identify the line that generated the alarm. Use CLR_Adj to clear the adjustment value.

MOTN–321 Posn unreachable at line %s

Cause: During a fine adjustment check, a position that cannot be reached was detected.
Remedy: From the alarm message, identify the line that generated the alarm. Use CLR_Adj to clear the adjustment value.

MOTN–340 Fast fault recovery

Cause: This is notification for application process enabled in the fast fault recovery when the alarm position is found.
Remedy: N/A

MOTN–341 NO Z offset for INC motion

Cause: A Z offset value cannot be applied to an incremental motion.
Remedy: Do not use an incremental motion.

MOTN–342 Override change not allowed

Cause: Change in teach pendant override setting while the program is running.
Remedy: Set Teach Pendant's override to the desired value and resume the program

PALT Error Codes (ID = 26)

PALT–001 ABORT.G Inadequate register value

Cause: Column/row/layer number is illegal.
Remedy: Please confirm palletizing register.

PALT–002 ABORT.G Computing shift vector

Cause: The position data of specified column/row/layer can not be calculated.
Remedy: Teach an appropriate stack point.

PALT–003 ABORT.G Computing rotation matrix

Cause: The position data of specified column/row/layer can not be calculated.
Remedy: Teach an appropriate stack point.

PALT–004 ABORT.G Increment value is ill

Cause: Increment value of parettaizing config screen is illegal.
Remedy: Please correct increment value

PALT–005 ABORT.G Application item unfound

Cause: Software internal error
Remedy: Teach a pallet instruction again.

PALT–006 ABORT.G Amr number is differnet
Cause: Software internal error
Remedy: Teach a pallet instruction again.

PALT–007 ABORT.G Configulation not decided
Cause: The array of column/row/layer of palletizing initial data is mistakes.
Remedy: Set the array of column/row/layer again.

PALT–008 ABORT.G Route pattern duplicated
Cause: Condition setting of the route pattern overlaps.
Remedy: Set the condition in the route pattern again.

PALT–009 ABORT.G Bottom point unfound
Cause: The correspoinding stack point does not exist.
Remedy: Teach the stack point again.

PALT–010 ABORT.G Route pattern unfound
Cause: Route pattern unfound.
Remedy: Check route pattren in palletizing route pattern screen.

PALT–011 ABORT.G Route point unfound
Cause: The correspoinding route point does not exist.
Remedy: Teach the route point again.

PALT–012 ABORT.G Route Bottom point unfound
Cause: The stack point of a present route pattern is not found.
Remedy: Teach the stack point of a present route pattern.

PALT–013 ABORT.G Memory id error
Cause: Setting of palletizing data is imcomplete.
Remedy: Contact our service center serving your locality.

PALT–014 ABORT.G Illegal bottom number
Cause: Software internal error
Remedy: Contact our service center serving your locality.

PALT–015 ABORT.G Illegal pattern number
Cause: Software internal error
Remedy: Contact our service center serving your locality.

PALT–016 ABORT.G Illegal rout number
Cause: Software internal error
Remedy: Contact our service center serving your locality.

PALT–017 ABORT.G Illegal current number
Cause: Software internal error
Remedy: Contact our service center serving your locality.

PALT–018 ABORT.G Illegal key input
Cause: Software internal error
Remedy: Contact our service center serving your locality.

PALT–019 ABORT.G Illegal point data
Cause: Software internal error
Remedy: Contact our service center serving your locality.

PALT–020 ABORT.G Function code unfound

Cause: Software internal error
Remedy: Contact our service center serving your locality.

PALT–021 ABORT.G Register index ill value

Cause: Software internal error
Remedy: Contact our service center serving your locality.

PALT–022 ABORT.G Item code cannot be found

Cause: Software internal error
Remedy: Contact our service center serving your locality.

PALT–023 ABORT.G Illegal element value

Cause: Software internal error
Remedy: Contact our service center serving your locality.

PALT–024 ABORT.G Calculation error occured

Cause: Palletizing data is imcomplete.
Remedy: Teach bottom point all ?
Teach rout point all ?

PALT–025 ABORT.G OS error occured

Cause: Software internal error

PALT–026 ABORT.G Cannot read/write to PL[]

Cause: Sofeware internal error.
Remedy: Check palletizing register index

PALT–027 ABORT.G Failed program close

Cause: Software internal error

PALT–028 ABORT.G Failed program open

Cause: Software internal error

PALT–029 ABORT.G Not exist application data

Cause: Software internal error

PALT–030 WARN Pallet number is over max

Cause: Don't teach palletizing instruction over 16
Remedy: palletizing instruction don't teach more than 16 in this program, Please teach another program.

PALT–031 WARN Can not be set FREE or INTER

Cause: In FREE configuation, can not be set INTER over two directions
Remedy: In FREE configuation, can be set INTER only one direction (ROW, COLUMN or LAYER)

PALT–033 This speed type isn't supported

Cause: Palletizing option doesn't support this speed type
Remedy: Please select the other motion statement.

PROG Error Codes

PROG–001 to 004 ABORT.L Invalid pointer is specified

Cause: System internal error.
Remedy: Contact our service center serving your locality.

PROG–005 WARN Program is not found

Cause: The specified program cannot be found.
Remedy: Check the program name.

PROG–006 WARN Line is not found

Cause: The specified line number cannot be found.
Remedy: Check the line number.

PROG–007 WARN Program is already running

Cause: The specified program is already being executed.
Remedy: Check the program name.

PROG–008 WARN In a rtn when creating a task

Cause: Execution cannot be started in sub-routine program.
Remedy: Check the line number.

PROG–009 WARN Line not same rtn as paused at

Cause: The program attempted to resume at a line different from the paused line.
Remedy: Check the line number.

PROG–010 WARN Not same prg as paused

Cause: A program, different from the paused program, attempted to resume.
Remedy: Check the program name.

PROG–011 PAUSE.L Cannot get the motion control

Cause: Motion control cannot be obtained.
Remedy: Check the teach pendant enable switch and other running programs to determine who has motion control.

PROG–012 WARN All groups not on the top

Cause: The program attempted to resume at a motion different from the paused motion.
Remedy: Resume the motion paused the last time.

PROG–013 WARN Motion is stopped by program

Cause: This motion was paused by the MOTION PAUSE instruction. Only the RESUME MOTION program instruction can resume the motion.
Remedy: Use the RESUME MOTION instruction in the program.

PROG–014 WARN Max task number exceed

Cause: The number of programs you attempted to start exceeded the maximum number allowed.
Remedy: Abort any unnecessary programs.

PROG–015 WARN Cannot execute backwards

Cause: Backward execution cannot be used.
Remedy: Do not use backward execution at this point.

PROG–016 WARN Task is not found

Cause: The specified task is not running or paused.
Remedy: Check the task name.

PROG–017 WARN Task is not running

Cause: The specified task is not running.
Remedy: Check the task name.

PROG–018 ABORTG Motion stack overflowed

Cause: Too many programs are paused.
Remedy: Resume or abort some programs.

PROG–019 WARN Ignore pause request

Cause: The request to pause the program was ignored.

PROG–020 WARN Task is already aborted

Cause: The specified program was already aborted.
Remedy: Check the program name.

PROG–021 WARN Ignore abort request

Cause: The request to abort the program was ignored.

PROG–022 WARN Invalid request type

Cause: Internal error
Remedy: Contact our service center serving your locality.

PROG–023 WARN Task is not paused

Cause: The specified program is not paused.
Remedy: Pause the program.

PROG–024 WARN Not have motion history

Cause: The motion path record is lost.
Remedy: Do not attempt backwards execution at this time.

PROG–025 WARN Cannot execute backwards

Cause: Backward execution cannot be used.
Remedy: Do not use backwards execution here.

PROG–026 WARN No more motion history

Cause: Backward execution cannot be used any more. The current line is on top of the memorized path.

PROG–027 to 033 WARN Invalid task number

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

PROG–034 WARN Routine not found

Cause: The specified routine cannot be found.
Remedy: Check the routine name and verify it is loaded.

PROG–035 WARN Not locked the specified group

Cause: Motion control for the specified group cannot be locked.
Remedy: Check the teach pendant enable switch and other running programs to determine who has motion control.

PROG–036 WARN The length of trace array is 0

Cause: Either there is not enough memory available, or the task attribute is set incorrectly.
Remedy: Set the trace buffer length using the KCL SET TASK TRACELEN command.

PROG–037 WARN No data in the trace array

Cause: There is no execution record in memory.
Remedy: Turn on tracing using the KCL SET TRACE ON command.

PROG–038 Inconsistency in task status

Cause: Internal system error.
Remedy: Consult our service representative.

PROG–039 WARN locked, but not get mctl

Cause: Motion control for the specified group was reserved, but it cannot be obtained.
Remedy: Check the teach pendant enable switch and other running programs to determine who has motion control.

PROG–040 PAUSE.L Already locked by other task

Cause: Motion control for the specified group was already reserved by another program.

Remedy: Check the other running programs to determine who has motion control.

PROG–041 WARN mctl denied because released

Cause: Motion control is released. The teach pendant currently has motion control. The robot cannot be started until motion control is obtained.

Remedy: Disable the teach pendant.

PROG–042 WARN Already released

Cause: Motion control was already released.

Remedy: If you had expected that the task may have already released the group, this may not be an error. Otherwise, check UNLOCK_GROUP usage.

PROG–043 WARN Already released by you

Cause: Motion control was already released by request of this program.

Remedy: If you had expected that the task may have already released the group, this may not be an error. Otherwise, check UNLOCK_GROUP usage.

PROG–044 WARN Arm has not been released yet

Cause: Motion control was not released yet.

Remedy: If you had expected that the task may have already locked the group, this may not be an error. Otherwise, check LOCK_GROUP usage.

PROG–045 WARN Other than requestor released

Cause: Motion control was already released by the request of another program.

Remedy: If you had expected that another task may have already released the group, this may not be an error. Otherwise, check UNLOCK_GROUP usage.

PROG–046 PAUSE.L TP is enabled while running (%s^7)

Cause: The teach pendant was enabled while the program was executing.

Remedy: Disable the teach pendant and resume the program.

PROG–047 PAUSE.L TP is disabled while running (%s^7)

Cause: The teach pendant was disabled while the program was executing.

Remedy: Enable the teach pendant and use shift–FWD to resume execution.

PROG–048 PAUSE.L Shift released while running (%s^7)

Cause: The shift key was released while the program was executing.

Remedy: Hold the shift key and press the FWD key to resume execution.

PROG–049 WARN Cannot release, robot moving

Cause: Motion control cannot be released because the robot is moving.

Remedy: Check the status of robot motion.

PROG–050 WARN Abort still in progress

Cause: The program is in the process of being aborted.

Remedy: Wait a few seconds.

PROG–051 WARN Cannot skip the return stmt

Cause: The specified lines to which a move was attempted exceed the number of lines in the program.

Remedy: Check the line number.

PROG–052 ABORT.L Process is aborted while executing

Cause: The user application task was forced to abort while the application was executing.

Remedy: This requires no special action for the user.

PROG–053 ABORT.L User AX is not running

Cause: The user application task was not executed.
Remedy: Start the user application task before executing the application.

PROG–054 FWD released while running (%s^7)

Cause: FWD key was released while the program is executing.
Remedy: Hold the FWD key with shift key to resume execution.

PROG–055 BWD released while running (%s^7)

Cause: BWD key was released while the program is executing.
Remedy: Hold the BWD key with shift key to resume execution.

PROG–056 Motion data out is enable

Cause: The machine lock function is disabled, and the motion data output function is enabled.
Remedy: On the test execution screen, disable the motion data output function.

MACR Error Codes (ID = 57)

MACR–001 WARN Can't assign to MACRO command

Cause: The conditions for assigning macros are not correct.
1 The allocation definition is duplicated.
2 The index is beyond the set range.
Remedy: Modify the device allocation.

MACR–003 WARN Can't assign motn_prog to UK

Cause: It is not possible to assign a program with MOTION lock group to the User Key(UK) button.
Remedy: Remove the motion lock group from the program.

MACR–004 WARN Can't execute motn_prog by UK

Cause: It is not possible to execute a program with MOTION lock group with the User Key(UK) button.
Remedy: 1 Remove all the motion groups from the group mask for detailed program information.
2 Allocate the program to other devices (SU, SP, and MF).

MACR–005 WARN Please enable teach pendant

Cause: It is not possible to execute a program when the teach pendant is disabled.
Remedy: Enable the teach pendant.

MACR–006 WARN Please disable teach pendant

Cause: It is not possible to execute a program when the teach pendant is enabled.
Remedy: Disable the teach pendant.

MACR–007 WARN The same macro type exists

Cause: The macro assign type already exists.
Remedy: Change the assign type.

MACR–008 WARN Remote–cond isn't satisfied

Cause: This assign type is only enabled at a REMOTE condition.
Remedy: Create a REMOTE condition.

MACR–009 WARN The index is out of range

Cause: This assign index is out of range.
Remedy: Change the assign index so that it is within the valid range.

MACR–010 WARN This SOP button is disabled

Cause: This SOP button is disable for macro execution.
Remedy: Change the value of the $MACRSOPENBL system variable.

MACR–011 WARN This UOP button is disabled

Cause: This UOP signal is disabled for macro execution.
Remedy: Change the value of the $MACRUOPENBL system variable.

MACR–012 WARN Number of DI+RI is over

Cause: The number of RI+DI is over the maximum number.
Remedy: First deassign the other RI or DI assignments. Then assign the new macro as RI or DI.

MACR–013 WARN MACRO execution failed

Cause: Cannot execute this MACRO.
Remedy: Refer to the error cause code.

MACR–016 WARN The macro is not completed

Cause: The macro aborted while executing.
Remedy: The macro will begin executing from the first line at the next execution.

MEMO Error Codes (ID = 7)

MEMO–001 WARN System error

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MEMO–002 WARN Specified program is in use

Cause: The specified program is editing or executing.
Remedy: Either abort the specified program or select it once more after selecting another program.

MEMO–003 WARN Specified program is in use

Cause: The specified program is editing or executing.
Remedy: Either abort the specified program or select it once more after selecting another program.

MEMO–004 WARN Specified program is in use

Cause: The specified program is editing or executing.
Remedy: Either abort the specified program or select it once more after selecting another program.

MEMO–006 WARN Protection error occurred

Cause: The specified program is protected by a user.
Remedy: Cancel the protection of the specified program.

MEMO–007 WARN Invalid break number

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MEMO–008 WARN Specified line no. not exist

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MEMO–009 WARN System error

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MEMO–010 WARN Program name error

Cause: The specified program name is different from the one in the P-code file.
Remedy: Specify the same program name.

MEMO–011 WARN System error

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MEMO–012 WARN System error

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MEMO–013 WARN Program type is different

Cause: The specified program type is different from that of the object being processed.
Remedy: Specify the same program type.

MEMO–014 WARN Specified label already exists

Cause: The specified label id already exists in the program.
Remedy: Specify another label number.

MEMO–015 WARN Program already exists

Cause: The specified program already exists in the system.
Remedy: Either specify another program name or delete the registered program.

MEMO–019 WARN Too many programs

Cause: The number of the programs and routines exceeded the maximum allowed (3200).
Remedy: Delete unnecessary programs and routines.

MEMO–020 to 024 WARN System error

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MEMO–025 WARN Label does not exist

Cause: Specified label does not exist.
Remedy: Set the index to an existing label.

MEMO–026 WARN Line data is full

Cause: The number of line data exceeded the maximum possible line number (65535).
Remedy: Delete unnecessary line data.

MEMO–027 WARN Specified line does not exist

Cause: The specified line data does not exist.
Remedy: Specify another line number.

MEMO–028 WARN System error

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MEMO–029 WARN The line data can't be changed

Cause: The specified line data cannot be changed. The size of modified data is different from that of original data when replacing it.
Remedy: Specify another line number or data of the same size.

MEMO–030 and 031 WARN System error

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MEMO–032 WARN Specified program is in use

Cause: The specified program is editing or executing.
Remedy: Either abort the specified program or select it once more after selecting another program.

MEMO–034 WARN The item can't be changed

Cause: The specified item is locked to change by system.
Remedy: Specify another item.

MEMO–035 to 037 WARN System error

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MEMO–038 WARN Too many programs

Cause: The number of programs exceeded the maximum allowed.
Remedy: Delete unnecessary programs.

MEMO–039 to 047 WARN System error

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MEMO–048 WARN Break point data doesn't exist

Cause: The specified break point data does not exist.
Remedy: Specify another break point.

MEMO–049 WARN System error

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MEMO–050 WARN Program does not exist

Cause: The specified program does not exist in the system.
Remedy: Specify another program or create the new program first.

MEMO–051 to 055 WARN System error

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MEMO–056 WARN Program does not exist

Cause: The specified program does not exist in the system.
Remedy: Specify another program or create the new program first.

MEMO–057 to 064 WARN System error

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MEMO–065 WARN Too many opened programs

Cause: Too many CALL instructions being used. The number of opened programs exceeded the maximum allowed (100).
Remedy: Abort any unnecessary programs or remove unnecessary CALL instructions.

MEMO–066 and 067 WARN System error

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MEMO–068 WARN Specified program is in use

Cause:
1 The specified program is editing or executing.
2 The specified program is tied to a MACRO.

Remedy:
1 Either abort the specified program or select it once more after selecting another program.
2 Remove the program from the MACRO entry.

MEMO–069 and 070 WARN System error

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MEMO–071 WARN Position does not exist

Cause: The specified position data does not exist.
Remedy: Specify another position.

MEMO–072 WARN Position data already exists

Cause: Position data already exists in the position you specified to move.
Remedy: Specify another position or delete the data in the specified position.

MEMO–073 WARN Program does not exist

Cause: The specified program does not exist in the system.
Remedy: Specify another program or create the new program first.

MEMO–074 WARN Program type is not TPE

Cause: The operation can be applied only to teach pendant programs.
Remedy: Select a teach pendant program.

MEMO–075 WARN Program can't be used

Cause: The program must be opened before attempting read or write operations.
Remedy: Open the program before reading or writing.

MEMO–076 to 077 WARN System error

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MEMO–078 WARN Program can't be used

Cause: The specified operation is not supported for program type.
Remedy: Specify a program whose program type matches the operation.

MEMO–079 WARN System error

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MEMO–080 WARN Protection error occurred

Cause: The specified program is protected by a user.
Remedy: Cancel the protection of the specified program.

MEMO–081 WARN Specified program is in use

Cause: The specified program is editing or executing.
Remedy: Abort the specified program or select it once more after selecting another program.

MEMO–082 to 087 WARN System error

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MEMO–088 WARN Program does not exist

Cause: The specified position data does not exist.
Remedy: Specify another position.

MEMO–089 to 092 WARN System error

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MEMO–093 WARN Specified program is in use

Cause: The specified program is editing or executing.
Remedy: Abort the specified program or select it once more after selecting another program.

MEMO–094 to 097 WARN System error

Cause: Internal system error.
Remedy: Contact our service center serving your locality.

MEMO–098 WARN EOF occurs in file access

Cause: EOF occurs in file access. When the P-code file was scanned, EOF occurs.

Remedy: The P-code data may be broken. Translate the specified KAREL program again. Then reload the P-code.

MEMO–099 WARN Program name is wrong

Cause: The length of the program name is different from that of the P-code data.

Remedy: Check the name of the specified program.

MEMO–100 to 102 WARN System error

Cause: Internal system error.

Remedy: Contact our service center serving your locality.

MEMO–103 WARN Check sum error occurred

Cause: The specified data was broken. This is the internal error.

Remedy: Contact our service center serving your locality.

MEMO–104 WARN Program already exists

Cause: The specified program already exists in the system.

Remedy: Specify another program name or delete the registered program.

MEMO–105 to 111 WARN System error

Cause: Internal system error.

Remedy: Contact our service center serving your locality.

MEMO–112 WARN Break data already exists

Cause: The specified break point data already exists in the program.

Remedy: Specify another break point.

MEMO–113 WARN File access error

Cause: The port that has the program you want to load is not connected.

Remedy: Check the port setting and the connected device.

MEMO–114 WARN Break point can't be removed

Cause: The break point data can not be overwritten because the program is protected by a user or is executing.

Remedy: Cancel the protection of the program or abort the program.

MEMO–115 WARN Break point can't be removed

Cause: The break point data can not be removed because the program is protected by a user or is executing.

Remedy: Cancel the protection of the program or abort the program.

MEMO–116 to 118 WARN System error

Cause: Internal system error.

Remedy: Contact our service center serving your locality.

MEMO–119 WARN Application data doesn't exist

Cause: The specified application data does not exist because the program does not correspond to the specified application.

Remedy: Specify another application data. Then create the program in the current system.

MEMO–120 WARN Application data doesn't exist

Cause: The specified application data does not exist because the program does not correspond to the specified application.

Remedy: Specify another application data. Create the program in the current system.

MEMO–121 to 122 WARN System error

Cause: Internal system error.

Remedy: Contact our service center serving your locality.

MEMO–123 WARN Application data doesn't exist

Cause: The specified application data does not exist because the program does not correspond to the specified application.

Remedy: Specify another application data. Then create the program in the current system.

MEMO–124 WARN Program version is too new

Cause: KAREL program version number is newer than that of the system.

Remedy: Translate the program with an older version of the Translator.

MEMO–125 WARN Program version is too old

Cause: KAREL program version number is older than that of the system.

Remedy: Translate the program with a newer version of the Translator.

MEMO–126 WARN No more available memory

Cause: Lack of the memory which can be used.

Remedy: Delete unnecessary programs.

MEMO–127 WARN Pos reference over 255 times

Cause: Reference of the same position exceeded the maximum count (256).

Remedy: Set new position ID for the referenced position.

MEMO–128 WARN %s parameters are different

Cause: A routine exists in memory with a different parameter definition than the routine in the PC file being loaded.

Remedy: Update the calling convention in the KAREL program being loaded or delete the obsolete routine from system memory.

MEMO–129 WARN System error

Cause: Internal system error.

Remedy: Contact our service center serving your locality.

MEMO–130 SYSTEM Please power up again

Cause: System data in CMOS has been broken.

Remedy: Turn power off and then back on.

MEMO–131 SYSTEM Please power up again

Cause: System data in CMOS has been broken.

Remedy: Turn power off and then back on.

MEMO–132 WARN %s has been broken

Cause: Program data has been broken at the power fail recover.

Remedy: Delete the program and create it again. Contact our service center serving your locality.

MEMO–133 SYSTEM Please power up again

Cause: System data in CMOS has been broken.

Remedy: Turn power off and then back on.

MEMO–134 WARN TPE program %s already exists

Cause: A teach pendant (TP) program with the same name already exists.

Remedy: Delete the teach pendant (TP) program. Then load the specified KAREL program again.

MEMO–135 WARN Cannot create TPE program here

Cause: The teach pendant (TP) program cannot be created in this start mode.

Remedy: On the auxiliary menu, switch the start mode to cold start or control start 2.

MEMO–136 WARN Cannot load P–code here

Cause: The KAREL program cannot be loaded in this start mode.

Remedy: Select the function menu to change the start mode, or power on again.

MEMO–137 WARN Load at Control Start Only

Cause: Specified KAREL program cannot be loaded in this mode. Because the same name program has already been loaded at controlled start.

Remedy: Load the program at controlled start.

MEMO–138 WARN Delete at Control Start Only

Cause: Specified program has already been loaded at controlled start. Because of this, you can only delete the program at controlled start.

Remedy: Delete the program at controlled start.

MEMO–139 to 143 WARN System error

Cause: Internal system error.

Remedy: Contact our service center serving your locality.

MEMO–144 WARN Header size too big

Cause: The teach pendant (TP) header size specified is too big. Must be less than 256.

Remedy: Change size to range of 1–256. If necessary, use multiple header records.

MEMO–145 TPE cannot have KAREL routine

Cause: A KAREL program with the same name already exists, so that a program with the specified name cannot be created.

Remedy: Change the name to a different one.

MEMO–146 Invalid variable is used

Cause: A KAREL program includes an invalid variable.

Remedy: Check the variable of the KAREL program.

MEMO–147 Flash File access error(write)

Cause: An attempt to write to the F–ROM failed.
Program data may be destroyed.

Remedy: The F–ROM may be faulty.

MEMO–148 Flash File access error(read)

Cause: An attempt to read from the F–ROM failed.
Program data may be destroyed.

Remedy: The F–ROM may be faulty.

MEMO–149 Specified program is broken

Cause: Program data is destroyed.

Remedy: Check the contents of the program.

MEMO–151 No more available memory(TEMP)

Cause: Temporary memory is insufficient.

Remedy: Delete unnecessary programs.

CMND Error Codes

CMND–001 WARN Directory not found

Cause: The specified directory can not be found.

Remedy: Check the device and path that you entered.

CMND–002 WARN File not found

Cause: The specified file could not be found.

Remedy: Check to make sure the file has been spelled correctly and that it exists. Also verify the device and path name are correct.

CMND–003 WARN File already exists

Cause: The file already exists and could not be overwritten.
Remedy: Make sure the overwrite option has been specified.

CMND–006 WARN Self copy not allowed

Cause: A file cannot be copied to itself.
Remedy: Change the name of the destination file so it is different from the source file.

CMND–009 WARN Position types are the same

Cause: Internal error.

CMND–010 WARN Source type code is invalid

Cause: Internal error.

CMND–011 WARN Destination type code is invalid

Cause: Internal error.

CMND–012 WARN Type codes do not match

Cause: Internal error.

CMND–013 WARN Representation mismatch

Cause: Internal error.

CMND–014 WARN Positions are not the same

Cause: Internal error.

CMND–015 WARN Both arguments are zero

Cause: Internal error.

CMND–016 WARN Division by zero

Cause: Internal error.

CMND–017 WARN Angle is out of range

Cause: Internal error.
Remedy: Make sure that the rotational angle is no greater than 100 times PI, or about 314.15926······.

CMND–018 WARN Invalid device or path

Cause: You have specified an invalid device or path.
Remedy: Check the device and path that you entered.

CMND–019 WARN Operation cancelled

Cause: The operation was cancelled because CTRL–C or CTRL–Y was pressed.
Remedy: Repeat the operation.

CMND–020 WARN End of directory

Cause: The directory listing is finished.
Remedy: This is a notification. You do not have to do anything for this warning message.

CMND–021 WARN Cannot rename file

Cause: The destination file name contained both alphanumeric characters and the global character '*'.
Remedy: Use only alphanumeric characters or a single global character when renaming a file.

CMND–022 STOP.G Time motion with dist before

Cause: A time–based motion was specified along with distance before.
Remedy: Do not use these options in combinations.

COND Error Codes

COND–001 WARN Condition does not exist

Cause: The number of a monitor to be enabled, disabled, or deleted is specified, but it is not found.
Remedy: Check the existing monitor numbers and specify one of them.

COND–002 WARN Condition handler superseded

Cause: The specified condition number already exists in the system, and has been superseded by the new condition.
Remedy: This is a notification. You do not have to do anything for this warning message.

COND–003 WARN Already enabled, no change

Cause: The specified condition is already enabled. No change has been made.
Remedy: This is just a notification, and you do not have to do anything for this warning massage.

COND–004 WARN Already disabled, no change

Cause: The specified condition is already disabled. No change has been made.
Remedy: This is just a notification, and you do not have to do anything for this warning message.

COND–009 WARN Break point encountered

Cause: Break point has been encountered.
Remedy: No action is required.

COND–010 WARN Cond exists, not superseded

Cause: The specified condition already exists. Condition was not superseded.
Remedy: Either renumber the condition handler or avoid re–defining the same condition handler.

COND–011 ABORT.G Scan time took too long

Cause: There are too many conditions defined. It took too long to scan them all.
Remedy: Reduce the number of conditions defined.

DICT Error Codes

DICT–001 WARN Dictionary already loaded

Cause: A dictionary cannot be reloaded if it was loaded into FROM.
Remedy: Load into a different language and use KCL SET LANG to set the language.

DICT–002 WARN Not enough memory to load dict

Cause: There is no more permanent memory available in the system to load another dictionary.
Remedy: Clear all unnecessary programs, dictionaries, or variables.

DICT–003 WARN No dict found for language

Cause: There are no dictionaries loaded for the specified language.
Remedy: Use the DEFAULT language or a language in which a dictionary has been loaded.

DICT–004 WARN Dictionary not found

Cause: The specified dictionary was not found.
Remedy: Use KCL LOAD DICT to load the dictionary into the DEFAULT language or the current language.

DICT–005 WARN Dictionary element not found

Cause: The dictionary element was not found.
Remedy: Check the dictionary or element number to be sure it is specified correctly.

DICT–006 WARN Nested level too deep

Cause: Only five levels of dictionary elements can be nested.
Remedy: Fix the dictionary text file to include fewer nested levels.

DICT–007 WARN Dictionary not opened by task

Cause: The dictionary was never opened.
Remedy: Remove the close operation.

DICT–008 WARN Dictionary element truncated

Cause: The dictionary element was truncated because the KAREL string array is not large enough to hold all the data.
Remedy: Increase either the size of the string or the number of strings in the array.

DICT–009 WARN End of language list

Cause: The language list has completed.
Remedy: This is a notification. You do not have to do anything for this warning message.

DICT–010 WARN End of dictionary list

Cause: The dictionary list has completed.
Remedy: This is a notification. You do not have to do anything for this warning message.

DICT–011 WARN Dict opened by too many tasks

Cause: Only five dictionaries can be open by one task at one time.
Remedy: Load the dictionary files into F–ROM or C–MOS memory, where file open processing is not required. Close any unused dictionary files.

DICT–012 WARN Low on FROM, loaded to memory

Cause: Not enough memory exists in FROM so the dictionary was loaded to CMOS.
Remedy: Store the dictionaries into C–MOS memory.

DICT–013 WARN Cannot open dictionary file

Cause: The dictionary file does not exist on the specified device or in the specified directory.
Remedy: Select the proper device/directory and try again.

DICT–014 WARN Expecting $ in dictionary file

Cause: The dictionary text incorrectly specifies an element without a $.
Remedy: Make sure all dictionary elements begin with a $.

DICT–015 WARN Reserved word not recognized

Cause: A reserved word was not recognized in the dictionary text.
Remedy: Check for misspellings or look up the correct word in the KAREL Reference Manual.

DICT–016 WARN Ending quote expected

Cause: The dictionary text incorrectly specifies an element without using quotes.
Remedy: Make sure all dictionary text is surrounded by double quotes. Use a backslash if you want an actual quote to appear in the text. For example, \"This is an example\" will produce "This is an example"

DICT–017 WARN Expecting element name or num

Cause: A reference to another element is expected.
Remedy: Use the element number to reference the element.

DICT–018 WARN Invalid cursor position

Cause: The cursor position is specified incorrectly or the values are outside the limits.
Remedy: Make sure the cursor position is valid. For example, use @1,1 for the first row and col respectively.

DICT–019 WARN ASCII character code expected

Cause: A series of digits are expected after the # to specify an ASCII character code.
Remedy: Remove the # or look up the ASCII character code in the KAREL Reference Manual.

DICT–020 WARN Reserved word expected

Cause: An identifier is expected after the & to specify a reserved word.
Remedy: Remove the & or look up the reserved word in the KAREL Reference Manual.

DICT–021 WARN Invalid character

Cause: An unexpected character was found in the dictionary text file.
Remedy: Make sure all dictionary text is correct.

DICT–022 WARN Dict already opened by task

Cause: The dictionary is already open by the task.
Remedy: This is a notification. You do not have to do anything for this warning message.

DICT–023 WARN Dict does not need to be opened

Cause: Dictionaries loaded to memory do not need to be opened.
Remedy: Do not try to open the dictionary file.

DICT–024 WARN Cannot remove dictionary file

Cause: Dictionaries loaded to FROM cannot be removed or a dictionary cannot be removed if another task has it opened.
Remedy: Do not try to remove a dictionary loaded to FROM. Remove the dictionary from the same task which loaded it.

DICT–025 Invalid state – internal error

Cause: Incorrect scanning.
Remedy: Correct the text of the dictionary.

DICT–028 WARN No FROM write, loaded to memory

Cause: Not enough memory exists in FROM so the dictionary was loaded to CMOS for R-J.
Remedy: This is a notification. You do not have to do anything for this warning message.

DICT–029 WARN Help element not found

Cause: The help dictionary element was not found.
Remedy: Check the dictionary to be sure the help dictionary element was specified correctly. The help dictionary element must be specified with a question mark (?) followed by the element number.

DICT–030 WARN Function key element not found

Cause: The function key dictionary element was not found.
Remedy: Check the dictionary to be sure the function key element was specified correctly. The function key element must be specified with a caret (^) followed by the element number.

LANG Error Codes

LANG–004 WARN File is not open

Cause:
1 A file having the same name already exists.
2 The specified file has already been opened.
3 The file is write–protected.
4 When a floppy disk is used, it has no free space.

Remedy:
1 Delete any unnecessary files, or rename the file.
2 Close the file.
3 Cancel write protection.
4 Use a new floppy disk. Or, delete any unnecessary files from the existing floppy disk to create sufficient free space to save the file.

LANG–005 WARN Program type is different

Cause: Only able to process teach pendant programs.
Remedy: Please select a teach pendant program.

LANG–006 Invalid or corrupted TP file

Cause: The data of a program file cannot be read correctly.
Remedy: Check the port setting.
Check the Handy File setting.
Check the floppy or memory card connection.
If the checks above cannot correct the error, the data of the file may be destroyed.

LANG–007 System Error

Cause: The data of a program file cannot be read correctly.

Remedy: Check the port setting.
Check the Handy File setting.
Check the floppy or memory card connection.
If the checks above cannot correct the error, the data of the file may be destroyed.

LANG–014 WARN Program already exists

Cause: The program that is about to load, already exists in the system.

Remedy: Before you load it, delete the program already in the system.

LANG–015 WARN Can not write file

Cause: 1 The file is write–protected.
2 Data of the specified size could not be written.

Remedy: 1 Cancel write protection.
2 The disk may be faulty. Replace the disk.

LANG–016 WARN Can not read file

Cause: Data of the specified size could not be read. Data communication failed.

Remedy: Check the connection of the device.

LANG–017 WARN File format is incorrect

Cause: The data you are trying to save to a file is either abnormal or broken, therefore the file cannot be loaded.

Remedy: The file cannot be loaded with the data as it is. The data must be normal to load the file.

LANG–018 WARN Group mask value is incorrect

Cause: When printing the program, there was an illegal position that did not match the group mask of the program.

Remedy: Reteach the position data so that the group number matches the group mask of the program.

LANG–050 WARN %s contains %s, program/file names must match

Cause: The file name and the program name are not same. Their names must match.

Remedy: Rename the file name to be same as the program name.

LANG–094 WARN File already exists

Cause: The specified file already exists on the floppy.

Remedy: Before you write the new file to the floppy, delete the file that already exists on the floppy.

LANG–095 WARN File does not exist

Cause: The specified file does not exist on the floppy.

Remedy: Check the file name or content of the floppy.

LANG–096 WARN Disk is full

Cause: The floppy disk has reached its limit and is full.

Remedy: Either use a new floppy disk or delete an necessary file in order to make room for saving to the floppy.

LANG–098 WARN Disk timeout

Cause: Could not access the disk.

Remedy: Check if the correct device is set to port and if it turns on.

LANG–099 WARN Write protection violation

Cause: The disk has write protection.

Remedy: Cancel the write protection.

LANG–100 WARN Device error

Cause: Could not access the device.

Remedy: Connect the correct device to the correct port.

MCTL Error Codes

MCTL–001 NONE TP is enabled

Cause: The teach pendant is enabled, and the motion control was not granted.
Remedy: Disable the teach pendant and try the operation again.

MCTL–002 NONE TP is disabled

Cause: The teach pendant is disabled, and the motion control was not granted.
Remedy: Enable the teach pendant and try the operation again.

MCTL–003 NONE system is in error status

Cause: The motion control was not granted because the system is in error status.
Remedy: Clear the error, and try the operation again.

MCTL–004 NONE motion is in progress

Cause: The motion is still in progress, and the motion control was not granted.
Remedy: Wait until the robot comes to a complete stop.

MCTL–005 NONE not in control of motion

Cause: The motion control was not granted because of some unknown reason.
Remedy: Clear the reason, and try the operation again.

MCTL–006 NONE TP has motion control

Cause: The motion control was not granted because the teach pendant currently has motion control.
Remedy: Disable the teach pendant, and try the same operation again.

MCTL–007 NONE PROG has motion control

Cause: The motion control was not granted because the program has motion control
Remedy: Pause or abort the program, and try the operation again.

MCTL–008 NONE Operator panel has motion control

Cause: The motion control was not granted because the operator panel has the motion control.
Remedy: Set the $RMT_MASTER system variable correctly, and try the operation again.

MCTL–009 NONE Other has motion control

Cause: Another device has motion control, and the motion control was not granted.
Remedy: Set the $RMT_MASTER system variable correctly, and try the operation again.

MCTL–010 Other than msrc is rel'ing

Cause: Internal error
Remedy: Contact your FANUC customer service representative, and inform the representative of an alarm message displayed.

MCTL–011 Due to error processing

Cause: Internal error
Remedy: Contact your FANUC customer service representative, and inform the representative of an alarm message displayed.

MCTL–012 subsystem code unknown

Cause: Internal error
Remedy: Contact your FANUC customer service representative, and inform the representative of an alarm message displayed.

MCTL–013 NONE ENBL input is off

Cause: ENBL input on the UOP is off.
Remedy: Set ENBL input ON.

MCTL–014 NONE Waiting for Servo ready

Cause: The motion control was not granted because servo was not up.
Remedy: Wait for a few seconds until servo is up and ready.

MCTL–015 NONE Manual brake enabled

Cause: The motion control was not granted because manual brake control is enabled.
Remedy: Disable the manual brake control.

PRIO Error Codes

PRIO–001 WARN Illegal iotype

Cause: Port type specified is invalid.
Remedy: Use one of the port types defined in IOSETUP.KL.

PRIO–002 WARN Illegal index

Cause: Port number is invalid or not presently assigned.
Remedy: Correct the port number.

PRIO–003 SYST No memory available

Cause: The memory required for this operation is not available.
Remedy: Delete KAREL programs and/or variables to free memory.

PRIO–004 WARN Too few ports on mod too few ports on mod

Cause: There are not enough ports on the specified board or module to make the specified assignments
Remedy: Correct either the first port number or the number of ports.

PRIO–005 WARN bad logical port no

Cause: The specified port number in an assignment is invalid. It must be in the range of 1 – 32767.
Remedy: Correct the logical port number, so that it is within the valid range.

PRIO–006 WARN bad log port number in asgt

Cause: The specified port number in an assignment is invalid. It must be in the range of 1 – 32767.
Remedy: Correct the logical port number, so that it is within the valid range.

PRIO–007 WARN no match in deassign call

Cause: Port being deassigned is not presently assigned.
Remedy: Correct the port number.

PRIO–008 WARN phys ports not found

Cause: Physical port being assigned to, does not exist.
Remedy: Correct the rack number, slot number, or port number.

PRIO–009 WARN n_ports invalid

Cause: The number of ports in an assignment is invalid. It must be in the range of 1 – 128.
Remedy: Correct the number of ports, so that it is within the valid range.

PRIO–010 WARN bad phys port number is asgt

Cause: Invalid physical port number in assignment request. It must be greater than 1.
Remedy: Correct the physical port number, so that it is greater than 1.

PRIO–011 WARN asgt overlaps existing asgt

Cause: The logical port numbers being assigned overlap existing assignments.
Remedy: Correct the first port number or number of ports.

PRIO–012 WARN bad board num

Cause: The specified rack and/or slot number is invalid or refers to an unused rack/slot number.
Remedy: Correct the rack and/or slot number.

PRIO–013 WARN no aiseq for bd

Cause: An attempt was made to delete an analog input sequence which has not been defined.
Remedy: Check the rack and/or slot number.

PRIO–014 WARN ai seq too long

Cause: The specified analog input sequence is too long. The sequence has from 1 to 15 port numbers.
Remedy: Supply a sequence of an appropriate length.

PRIO–016 WARN log port already asgnd

Cause: The specified logic number is already in use.
Remedy: Use another logic number.

PRIO–017 WARN I/O point not sim I/O point not sim

Cause: You attempted to set an input port that was not simulated.
Remedy: Use the I/O menu to set the port simulated or avoid setting the port.

PRIO–020 SYST SLC communications error %d %d %d %d

Cause: An unrecoverable error is detected in communication with a process I/O board.
Remedy: Check the cable between the main CPU board and the I/O unit.
Check the SLC2 mounted on the main CPU board or I/O unit.

PRIO–021 Unknown I/O hardware

Cause: An unknown device is connected to the I/O Link connector.
Remedy: Replace the device with a device that is compatible with the current software or install a version of software that recognizes the device.

PRIO–022 Too much I/O data on I/O link

Cause: The devices connected to the I/O Link exceed the I/O link capacity.
Remedy: Disconnect some devices.

PRIO–023 WARN no ports of this type

Cause: There are no ports of the specified type.
Remedy: Change the port type, or define ports (e.g., GIN or GOUT) of the specified type.

PRIO–032 WARN too many DIO modules

Cause: More than 31 I/O units are connected through an I/O link.
Remedy: Disconnect some of the I/O units so that no more than 31 are connected.

PRIO–063 WARN Bad IO asg: rack% d^1 slot %d^2

Cause: The I/O unit to which a signal is allocated is not found. There is no I/O unit corresponding to the rack and slot numbers subsequent to the alarm message. Possible causes are as follows:
(1) The I/O unit has been replaced with another type of I/O unit.
(2) The I/O unit fuse has blown.
(3) Power is not supplied to the I/O unit.
(4) The I/O link cable is either disconnected or not securely connected.
(5) The I/O link cable is broken.
(6) The I/O unit is faulty.
Remedy: (1) If the I/O unit has been replaced, apply the following procedure to clear the I/O allocation.
1 Press MENU and select I/O. Then, press F1 (TYPE) and select I/O LINK to display the I/O link screen.
2 Press F5 (INTER CONNECT).
3 In response to the prompt ”RECOVER ALL,” press F4 (YES).
4 Turn the power off and then back on.
In this case, even when power restoration is enabled, all output signals are turned off.
(2) Replace the I/O unit fuse.
(3) Check the power supply to the I/O unit.
(4) Ensure that the I/O link cable is connected securely.
(5) Replace the I/O link cable.
(6) Replace the I/O unit.

PRIO–072 WARN Pulse output is full

Cause: Max of pulse output is 255 at the same time.
Remedy: Check the count of pulse output.

PRIO–081 I/O is not initialized

Cause: This indicates that an severe error has occured during I/O initialization at controller power–up.
Remedy: Check other error messages displayed on the TP alarm screen.

PRIO–083 Digital I/O is not recovered

Cause: Digital output port states are not recovered when semi–hot start is enabled because I/O device configuration or assignments have changed.
Remedy: Initialize I/O.

PRIO–085 BUSY in SLC2 does not turn off

Cause: BUSY bit in SLC2 does not turn off.
Remedy: Check SLC2 on Main CPU board or I/O device and I/O link cable.

PRIO–100 Model B comm fault %srack:%d slot:%d

Cause: Communication between the Model B interface unit and DI/DO units, or between DI/DO units, is lost.
Remedy: Check the power and cabling from Model B interface unit and DI/DO unit, or between DI/DO units.

PRIO–119 Too many DIGITAL I/O ports

Cause: There are too many DIGITAL I/O ports.
Remedy: Disconnect some DIGITAL I/O devices.

PRIO–125 SLC2 initialization error

Cause: The SLC2 is in an error state at the end of initialization.
Remedy: Check SYSFAIL of the other PCB.
Also check the main PCB.

ROUT Error Codes

ROUT–022 PAUSE.G Bad index in ORD

Cause: Internal error of software.
Remedy: Contact our service center serving your locality.

ROUT–023 PAUSE.G Bad index in SUBSTR

Cause: Internal error of software.
Remedy: Contact our service center serving your locality.

ROUT–024 PAUSE.G SUBSTR length less than 0

Cause: Internal error of software.
Remedy: Contact our service center serving your locality.

ROUT–025 ABORT.G Illegal semaphore number

Cause: Incorrect number is specified for semaphore id.
Remedy: Specify a number between 1 to 255.

ROUT–026 WARN Illegal group number

Cause: Invalid group number is specified.
Remedy: Specify existing group number.

ROUT–027 WARN String size not big enough

Cause: Specified string variable does not have enough room to hold the return data.
Remedy: Specify a larger size string variable.

ROUT–028 ABORT.G Illegal file attribute number

Cause: Incorrect file attribute id was specified.
Remedy: Specify a correct file attribute id.

ROUT–029 ABORT.G Illegal file attribute value

Cause: Incorrect file attribute value was specified.
Remedy: Specify a correct attribute value.

ROUT–030 WARN Non existent register number

Cause: A register number, that does not exist, is specified.
Remedy: Specify a correct register number.

ROUT–031 WARN Illegal register type

Cause: Incorrect register type is specified.
Remedy: Specify the correct register type for the attempted operation.

ROUT–032 ABORT.G Position type mismach

Cause: Position type is not correct for the operation.
Remedy: Specify correct position type.

ROUT–033 ABORT.G Illegal attribute type

Cause: Illegal attribute id was specified.
Remedy: Specify correct attribute id.

ROUT–034 WARN Not a TPE program

Cause: A non-teach pendant program is specified.
Remedy: Specify a program name other than a KAREL program.

ROUT–035 WARN Value is out of range

Cause: Internal error of software.
Remedy: Contact our service center serving your locality.

ROUT–037 ABORT.G Bad TPE header size

Cause: Value used in SET_HEAD_TPE for bfr_size is invalid.
Remedy: Use buffer size in the range 1–255.

ROUT–038 PAUSE.G Uninitialized TPE position

Cause: It indicates that the position data in the specified line of the specified TP program has not been recorded.
Remedy: Confirm the contents of position data.

ROUT–039 WARN Executing motion exists

Cause: Cannot unlock group while motion is executing.
Remedy: Wait until executing motion has completed.

ROUT–040 WARN Stopped motion exists

Cause: Cannot unlock group while stopped motion exists.
Remedy: Resume stopped motion and wait until motion has completed or cancel stopped motion.

ROUT–041 Dym. disp. var. not static

Cause: Internal system error.
Remedy: Consult our service representative.

ROUT–042 TPE parameters do not exist

Cause: The parameter designated by param_no does not exist.
Remedy: Confirm the param_no and the parameter in CALL/MACRO command in main TPE program.

SCIO Error Codes

SCIO–001 System Error

Cause: Internal error

Remedy: Contact your FANUC customer service representative, and inform the representative of an alarm message displayed.

SCIO–002 System Error

Cause: Internal error

Remedy: Contact your FANUC customer service representative, and inform the representative of an alarm message displayed.

SCIO–003 out buffer size is not enough

Cause: Internal system error.

SCIO–004 System Error

Cause: Internal error

Remedy: Contact your FANUC customer service representative, and inform the representative of an alarm message displayed.

SCIO–005 System Error

Cause: Internal error

Remedy: Contact your FANUC customer service representative, and inform the representative of an alarm message displayed.

SCIO–006 System Error

Cause: Internal error

Remedy: Contact your FANUC customer service representative, and inform the representative of an alarm message displayed.

SCIO–007 System Error

Cause: Internal error

Remedy: Contact your FANUC customer service representative, and inform the representative of an alarm message displayed.

SCIO–008 System Error

Cause: Internal error

Remedy: Contact your FANUC customer service representative, and inform the representative of an alarm message displayed.

SCIO–009 System Error

Cause: Internal error

Remedy: Contact your FANUC customer service representative, and inform the representative of an alarm message displayed.

SCIO–010 System Error

Cause: Internal error

Remedy: Contact your FANUC customer service representative, and inform the representative of an alarm message displayed.

SCIO–011 System Error

Cause: Internal error

Remedy: Contact your FANUC customer service representative, and inform the representative of an alarm message displayed.

SCIO–012 System Error

Cause: Internal error

Remedy: Contact your FANUC customer service representative, and inform the representative of an alarm message displayed.

SCIO–013 System Error

Cause: Internal error

Remedy: Contact your FANUC customer service representative, and inform the representative of an alarm message displayed.

SCIO–014 System Error

Cause: Internal error

Remedy: Contact your FANUC customer service representative, and inform the representative of an alarm message displayed.

SCIO–015 System Error

Cause: Internal error

Remedy: Contact your FANUC customer service representative, and inform the representative of an alarm message displayed.

SCIO–016 WARN This option does not exist

Cause: This option does not exist

Remedy: Confirm the bought option.

SCIO–017 System Error

Cause: Internal error

Remedy: Contact your FANUC customer service representative, and inform the representative of an alarm message displayed.

SCIO–018 System Error

Cause: Internal error

Remedy: Contact your FANUC customer service representative, and inform the representative of an alarm message displayed.

SCIO–019 System Error

Cause: Internal error

Remedy: Contact your FANUC customer service representative, and inform the representative of an alarm message displayed.

SCIO–020 WARN LBL[%d] exists in line %d:

Cause: This label number exists in another line.

Remedy: Select another label number.

SCIO–021 System Error

Cause: Internal error

Remedy: Contact your FANUC customer service representative, and inform the representative of an alarm message displayed.

SCIO–022 System Error

Cause: Internal error

Remedy: Contact your FANUC customer service representative, and inform the representative of an alarm message displayed.

SCIO–023 System Error

Cause: Internal error

Remedy: Contact your FANUC customer service representative, and inform the representative of an alarm message displayed.

SCIO–024 System Error

Cause: Internal error

Remedy: Contact your FANUC customer service representative, and inform the representative of an alarm message displayed.

SCIO–030 JOINT motion in slave program

Cause: The single slave execution program and slave program of the robot link cannot use a joint motion instruction.

Remedy: For teaching, use a linear or circular motion instruction.

SCIO–031 JOINT position in slave program

Cause: The single slave execution program and slave program of the robot link cannot use the joint position format for motion instruction position data.

Remedy: Use the orthogonal position format.

SCIO–032 Master UT mismatch

Cause: The current tool coordinate system number of the master robot does not match the tool coordinate system number specified on the program detail screen.

Remedy: Modify the tool coordinate system number of the master robot. Alternatively, modify the tool coordinate system number of the master robot on the program detail screen.

SCIO–033 Slave can have ony one motion line

Cause: The robot link slave program allows only one line of motion instructions to be taught.

Remedy: Ensure that the slave program contains only one line of motion instructions.

SRIO Error Codes

SRIO–002 SERIAL PORT NOT OPEN

Cause: Serial port is not opened.

Remedy: Open serial port before using it.

SRIO–003 SERIAL PORT ALREADY OPEN

Cause: Serial port has already been opened, and it was tried to be opened again.

Remedy: Do not try to open the serial port which has already be opened.

SRIO–004 SERIAL PORT NOT INITIALIZE

Cause: Serial port is not initialized.

Remedy: Initialize the serial port before using it.

SRIO–005 SERIAL PORT DSR OFF

Cause: Serial port DSR is off.

Remedy: Check if serial port setup is correct.
Check if cable is broken. Check if there exists a noise source near controller.
Check target device status.

SRIO–006 SERIAL PORT PARITY ERROR

Cause: Serial port parity error occured.

Remedy: Check if serial port setup is correct.
Check if cable is broken. Check if there exists a noise source near controller.

SRIO–007 SERIAL PORT OVERRUN ERROR

Cause: Serial port overrun error occured.

Remedy: Check if serial port setup is correct.
Check if cable is broken. Check if there exists a noise source near controller.

SRIO–008 SERIAL PORT FRAME ERROR

Cause: Serial port frame error occured.

Remedy: Check if serial port setup is correct.
Check if cable is broken. Check if there exists a noise source near controller.

SRIO–009 S. PORT PARITY & OVERRUN

Cause: Serial port parity error and overrun error occured.

Remedy: Check if serial port setup is correct.
Check if cable is broken. Check if there exists a noise source near controller.

SRIO–010 S. PORT PARITY & FRAME

Cause: Serial port parity error and frame error occured

Remedy: Check if serial port setup is correct.
Check if cable is broken. Check if there exists a noise source near controller.

SRIO–011 S. PORT OVERRUN & FRAME

Cause: Serial port overrun error and frame error occured.

Remedy: Check if serial port setup is correct.
Check if cable is broken. Check if there exists a noise source near controller.

SRIO–012 S. PORT PRTY & OVRRN & FRM

Cause: Serial port parity error, overrun error, and frame error occured.

Remedy: Check if serial port setup is correct.
Check if cable is broken. Check if there exists a noise source near controller.

SRIO–013 S. PORT DSR OFF & HARDWARE ERR

Cause: Serial port DSR is off and harware error occured.

Remedy: Check if serial port setup is correct.
Check if cable is broken. Check if there exists a noise source near controller.
Check target device status.
Check the hardware.

FLPY Error Codes

FLPY–001 End of directory reached

Cause: Your listing has reached the end of the directory.

Remedy: This is a notification. You do not have to do anything for this warning message.

FLPY–002 File already exists

Cause: The file name you are trying to create already exists on this device.

Remedy: Delete the file of this name or choose a different file name.

FLPY–003 File does not exist

Cause: The file you are trying to open does not exist on this device.

Remedy: Open a file that does exist on the device.

FLPY–004 Unsupported command

Cause: Operation is not supported on floppy disk.

Remedy: Use only operations supported on floppy disk.

FLPY–005 Disk is full

Cause: The disk file capacity has been reached.

Remedy: Delete some unneeded files or use a disk with sufficient free space.

FLPY–006 End of file reached

Cause: The end of the file was reached while reading.

Remedy: Do not attempt to read beyond the end of a file.

FLPY–008 Only one file may be opened

Cause: An attempt was made to open more than one file.

Remedy: Do not attempt to open more than one file at a time.

FLPY–009 Communications error

Cause: The protocol format was invalid.

Remedy: Retry the operation.

FLPY–015 Write protection violation

Cause: The disk has write protection enabled.
Remedy: Remove write protection from the disk or use a disk that is not write protected.

FLPY–100 Directory read error

Cause: The directory information is corrupted and unreadable.
Remedy: Try another disk or reformat the disk.

FLPY–101 Block check error

Cause: The checksum data is bad.
Data is corrupted on disk and can not be read.
Remedy: Try another disk, or reformat the disk

FLPY–103 Seek error

Cause: There is a bad sector or track on the disk.
Remedy: Clean the disk drive, try another disk, or reformat the disk.

FLPY–104 Disk timeout

Cause: The drive did not respond to a command.
Remedy: Check the cable to the drive and make sure drive power is on.

FLP–105 Write protection violation

Cause: The disk has write protection enabled.
Remedy: Remove write protection from the disk or use a disk that is not write protected.

FLPY–106 Memory Card hardware error

Cause: Memory Card hardware error is detected.
Remedy: Check Memory Card I/F unit connection or battery of the card.

FLPY–107 Not formatted card

Cause: The Memory Card is not formatted.
Remedy: Format the card with UTILITY menu on FILE screen.

FILE Error Codes

FILE–001 Device not ready

Cause: Specified file device is not ready.
Remedy: Check if the device is mounted and ready to use.

FILE–002 Device is Full

Cause: Device is full. There is no more space to store data on the device.
Remedy: Delete any unnecessary files or change to a new device.

FILE–003 Device is protected

Cause: Device is protected. So, you cannot write to the device.
Remedy: Release the device protection.

FILE–005 Device not mounted

Cause: Device is not mounted. You should mount the device before using it.
Remedy: Mount the correct file device.

FILE–006 Device is already mounted

Cause: You tried to mount the device which had been already mounted.
Remedy: Mount device only once.

FILE–008 Illegal device name

Cause: Device name contains an illegal character.
Remedy: Check spelling and validity of device name.

FILE–009 Illegal logical unit number

Cause: Illegal LUN is used.
Remedy: This is an internal error. Check the validity of the logical unit number.

FILE–010 Directory not found

Cause: Specified directory does not exist
Remedy: Check validity of directory name.

FILE–011 Directory full

Cause: Directory is full.
You tried to create a file in the root directory which execeeded the maximum number of files allowed on the device.
Remedy: Delete unnecessary files in the root directory.

FILE–012 Directory is protected

Cause: You tried to write to a write protected directory.
Remedy: Release the protection to the directory.

FILE–013 Illegal directory name

Cause: Directory name contains an illegal character.
Remedy: Check spelling of directory name.

FILE–014 File not found

Cause: The specified file was not found.
Remedy: Check that the file exists and that the file name was spelled correctly.

FILE–015 File is protected

Cause: You tried to access a protected file.
Remedy: Release the protection from file.

FILE–017 File not open

Cause: You tried to access a file which is not open.
Remedy: Open the file before accessing.

FILE–018 File is already opened

Cause: You tried to create/delete/rename a file which is already opened.
Remedy: Close file before such operations.

FILE–019 File is locked

Cause: You tried to access a file which is locked.
Remedy: Release the lock.

FILE–020 Illegal file size

Cause: File size is invalid.
Remedy: Change file size to be correct.

FILE–021 End of file

Cause: End of file was detected.

FILE–022 Illegal file name

Cause: File name contains an illegal character.
Remedy: Check spelling of file name.

FILE–023 Illegal file number

Cause: File number is illegal.
Remedy: Use a valid file number which is the ID returned from an open request.

FILE–024 Illegal file type

Cause: File type contains an illegal character.
Remedy: Check the spelling and validity of the file type.

FILE–025 Illegal protection code

Cause: File protection code is illegal.
Remedy: Check if the protection code is correct.

FILE–026 Illegal access mode

Cause: File access mode is illegal.
Remedy: Check if the access mode is correct.

FILE–027 Illegal attribute

Cause: File attribute in the SET_ATTRIBUTE request is illegal.
Remedy: Check that attribute specified is valid.

FILE–028 Illegal data block

Cause: Data block is broken which is used in FIND_NEXT request.
Remedy: You should keep the data block which is returned from the previous FIND_FIRST or FIND_NEXT request.

FILE–029 Command is not supported

Cause: Illegal request command is specified.
Remedy: Check if the request code is corect.

FILE–030 Device lun table is full

Cause: Device management table is full.
Remedy: Dismount any unnecessary devices.

FILE–031 Illegal path name

Cause: Path name contains an illegal character.
Remedy: Check if the path name is correct.

FILE–032 Illegal parameter

Cause: Illegal parameter is detected.
Remedy: Check that all parameters for the request are valid.

FILE–033 System file buffer full

Cause: File management buffer is full.
Remedy: Close unnecessary files.

FILE–034 Illegal file position

Cause: Illegal file position is specified.
Remedy: Check that the file position parameter from SEEK request is positive and not beyond the end of file.

FILE–035 Device not formatted

Cause: You tried to access a unformatted device.
Remedy: Format the device before using it.

FILE–036 File already exist

Cause: You tried to rename a file to an already existing file name.
Remedy: Change the new file name to be unique or delete the existing file.

FILE–037 Directory not empty

Cause: You tried to remove a subdirectory which contains some files or directories.
Remedy: Remove all files and directories in the subdirectory before removing subdirectory.

FILE–038 File locked by too many tasks

Cause: There are too many lock requests to same file.
Remedy: Unlock any unnecessary file lock requests.

FILE–039 Directory already exists

Cause: You tried to create a sub–directory that already exists.
Remedy: Use a unique name for new sub–directory

FILE–040 Illegal file access mode

Cause: You tried to read from a write only opened file or tried to write to a read only opened file.
Remedy: Open a file with correct access mode.

FILE–041 File not locked

Cause: You tried to unlock file which you had not locked.
Remedy: Don't unlock a file that is not locked.
You can only unlock files which YOU have locked.

SSPC Error Codes

SSPC–001 Waiting until space gets clear

Cause: Special checking space is not clear.
Remedy: No remedy

SSPC–002 Occer dead lock condition

Cause: The priority of space is invalid
Remedy: Set the priority valid

SSPC–003 AccuPath not allowed

Cause: Space Check function is not compatible with AccuPath.
AccuPath is not allowed.
Remedy: Not use AccuPath or disable space check function

SSPC–004 CTV option not allowed

Cause: Space Check function is not compatible with Continuous Turn CTV option. The CTV motion option is not allowed.
Remedy: Remove CTV option or disable space check function

SSPC–011 APDT error (i)

Cause: Internal error
Remedy: Contact your FANUC customer service representative, and inform the representative of the character string indicated in (i) of the message.

SSPC–012 Invalid element (s:i j)

Cause: The setting of model elements is incorrect.
Example of display: "Invalid element (G:1 6)" The sixth model element of group1 is set incorrectly.
"Invalid element (H:2 1)"The first model element of hand 2 is set incorrectly.
Remedy: Check the setting of model elements. Check that the setting of a link number and link type is correct.

SSPC–013 Invalid hand num (G:i UT:j)

Cause: The hand number assigned to tool coordinate system number (UT:j) of group (G:i) is invalid.
Remedy: On the model setting screen, check the hand number assignment.

SSPC–014 Common frame setting (G:i)

Cause: The inter–robot calibration of group (G:i) is not completed.
Remedy: Perform inter–robot calibration.

SSPC–015 Not calibrated (G:i)

Cause: The calibration of group (G:i) is not completed.
Remedy: Perform calibration.

SSPC–016 Invalid comb type (C:i s)

Cause: The model type on the (s) side (L [left], R[right]) of combination setting (C:i) is invalid.
Remedy: On the model combination setting screen, check the model type.

SSPC–017 Invalid comb index (C:i s)

Cause: The model number on the (s) side (L[left], R[right]) of combination setting (C:i) is invalid.
Remedy: On the model combination setting screen, check the model number.

SSPC–018 APDT is not supported (G:i)

Cause: The robot of group (G:i) does not support the proximity stop function.
Remedy: On the model combination setting screen, check the model type and model number.

SSPC–019 (G:i) is close to target

Cause: An interference was detected.
Remedy: The alarm can be released by an ordinary reset operation.

SSPC–020 Invalid fixture obj (F:i)

Cause: The teach group number of jig model (F:i) is invalid.
Remedy: On the jig model setting screen, check the teach group number.

SSPC–021 Too many settings

Cause: There are an excessive number of model settings or combination settings.
Remedy: Reduce the number of settings.

SSPC–101 (G:i) is close to target

Cause: Proximity was detected. (i: Group number)
Remedy: The alarm can be released by an ordinary reset operation.

SSPC–101 SSPC–102 (G:i) is close to target(qstop)

Cause: Proximity was detected. (i: Group number)
Remedy: The alarm can be released by an ordinary reset operation.

SSPC–103 (G:i) is near to target

Cause: A gradual stop occurred. (i: Group number)
Remedy: The alarm can be released by an ordinary reset operation.

SSPC–104 APDT error (i)

Cause: Internal error (i: Error number)
Remedy: This alarm is not issued usually. Inform your FANUC customer service representative of the numeric value indicated in (i) of the message.

SSPC–105 Too many settings

Cause: There are an excessive number of model settings or combination settings.
Remedy: Reduce the number of settings.

SSPC–106 Failed to get dist (j,C:i)

Cause: The distance between model elements could not be calculated.
Remedy: SSPC–111. This alarm is not issued usually. Inform your FANUC customer service representative of the numeric value indicated in (j,C:i) of the message.

SSPC–111 Invalid comb type (ST,C:i,s)

Cause: The model type on the (s) side (L[left], R[right]) of combination number (C:i) in the proximity stop combination setting is invalid.

Remedy: On the proximity stop combination setting screen, check the model type.

SSPC–112 Invalid comb index(ST,C:i,s)

Cause: The model number on the (s) side (L[left], R[right]) of combination number (C:i) in the proximity stop combination setting is invalid.

Remedy: On the proximity stop model combination setting screen, check the model number.

SSPC–113 APDT isn't supported (ST,G:i)

Cause: The robot of group (G:i) does not support the proximity stop function.

Remedy: On the proximity stop combination setting screen, check the model type and model number.

SSPC–114 Not calibrated (ST,G:i)

Cause: The calibration of group (G:i) is not completed.

Remedy: Perform calibration.

SSPC–115 Invalid utool number (ST,G:i)

Cause: The tool coordinate system number of group (G:i) is invalid.

Remedy: Check the tool coordinate system number.

SSPC–116 Invalid hand num(ST,G:i,UT:j)

Cause: The hand number assigned to tool coordinate system number (UT:j) of group (G:i) is invalid.

Remedy: On the model setting screen, check the hand number assignment.

SSPC–117 Common frame setting (ST,G:i)

Cause: The inter–robot calibration of group (G:i) is not completed.

Remedy: SSPC–111. Perform inter–robot calibration.

SSPC–118 Invalid element (ST,s:i,j)

Cause: The setting of model elements is incorrect.
Example of display: "Invalid element (ST,G:1 6) "The sixth model element of group1 is set incorrectly.
"Invalid element (ST ,H:2 1)"The first model element of hand 2 is set incorrectly.

Remedy: Check the setting of model elements. Check that the setting of a link number and link type is correct.

SSPC–119 Can't get elem pos(ST,G:i,j)

Cause: The current position of a model element could not be calculated.

Remedy: This alarm is not issued usually. Inform your FANUC customer service representative of the numeric value indicated in (ST,G:i, j) of the message.

SSPC–120 Invalid fixture obj (ST,F:i)

Cause: The teach group number of jig model (F:i) is invalid.

Remedy: On the jig model setting screen, check the teach group number.

SSPC–131 Invalid comb type (WT,C:i,s)

Cause: The model type on the (s) side (L[left], R[right]) of combination number (C:i) in the proximity wait combination setting is invalid.

Remedy: On the proximity wait combination setting screen, check the model type.

SSPC–132 Invalid comb index(WT,C: i,s)

Cause: The model number on the (s) side (L[left], R[right]) of combination number (C:i) in the proximity wait combination setting is invalid.

Remedy: On the proximity wait combination setting screen, check the model number.

SSPC–133 APDT isn't supported (WT,G:i)

Cause: The robot of group (G:i) does not support the proximity wait function.

Remedy: On the proximity wait combination setting screen, check the model type and number.

SSPC–134 Not calibrated (WT,G:i)

Cause: The calibration of group (G:i) is not completed.

Remedy: Perform calibration.

SSPC–135 Invalid utool number (WT,G:i)

Cause: The tool coordinate system number of group (G:i) is invalid.

Remedy: Check the tool coordinate system number.

SSPC–136 Invalid hand num(WT,G:i,UT:j)

Cause: The hand number assigned to tool coordinate system number (UT:j) of group (G:i) is invalid.

Remedy: On the model setting screen, check the hand number assignment.

SSPC–137 Common frame setting (WT,G:i)

Cause: The inter–robot calibration of group (G:i) is not completed.

Remedy: Perform inter–robot calibration.

SSPC–138 Invalid element (WT,s:i,j)

Cause: The setting of model elements is incorrect.
Example of display: "Invalid element (WT,G:1 6)"The sixth model element of group1 is set incorrectly.
"Invalid element (WT,H:2 1)"The first model element of hand 2 is set incorrectly.

Remedy: Check the setting of model elements. Check that the setting of a link number and link type is correct.

SSPC–139 Can't get elem pos(WT,G:i,j)

Cause: The current position of a model element could not be calculated.

Remedy: This alarm is not issued usually. Contact your FANUC customer service representative of the numeric value indicated in (WT,G:i, j) of the message.

SSPC–140 Invalid fixture obj (WT,F:i)

Cause: The teach group number of jig model (F:i) is invalid.

Remedy: On the jig model setting screen, check the teach group number.

SSPC–151 App_STOP (ST,C:i) is disabled

Cause: An attempt was made to temporarily disable invalid proximity stop combination (C:i) on the setting screen with a program instruction.

Remedy: Enable the proximity stop combination on the setting screen before using it.

SSPC–152 App_STOP (ST,C:i) is disabled

Cause: An attempt was made to temporarily disable invalid proximity stop combination (C:i) on the setting screen with a program instruction.

Remedy: Enable the proximity stop combination on the setting screen before using it.

SSPC–153 (WT,C:i) is enabled by other

Cause: An attempt was made to enable/disable proximity wait condition number (C:i) already enabled by another task.

Remedy: The proximity wait condition number is currently used by another program. Use it after it is freed.

SSPC–154 (ST,C:i) is disabled by other

Cause: The proximity stop instruction was used for proximity stop condition number (C:i) being used by another task.

Remedy: The proximity stop condition number is currently used by another program. Use it after it is freed.

SSPC–155 Invalid host name (ST,C:i)

Cause: In a specified proximity stop combination, an invalid host name is set.

Remedy: On the host communication screen of the setting screen, check the host name.
Set a correct host name.

SSPC–156 Invalid host name (WT,C:i)

Cause: In a specified proximity wait combination (C:i), an invalid host name is set.

Remedy: On the host communication screen of the setting screen, check the host name.
Set a correct host name.

SSPC–157 Intrupt signal (WT,C:i)

Cause: In the proximity wait state, the proximity wait halt signal was input.

Remedy: If the halt is unexpected, check the setting of the proximity wait halt signal. Moreover, check if the same signal is used for another purpose.

SSPC–158 App_WAIT timeout (WT,C:i)

Cause: In the proximity wait state, the time set in time–out has elapsed.

Remedy: Adjust the wait time. To wait infinitely, set 0 in ”time–out” on the proximity wait combination setting screen.

SSPC–159 App_WAIT can't be used(WT,G:i)

Cause: In the following operations, automatic stop/restart based on the proximity wait function cannot be performed:

- When slave robot follow–up operation is being performed based on robot link synchronization
- When the continuous rotation function is being used

Remedy: Do not use any of the two functions above at the time of automatic stop/restart operation.

SSPC–160 App_STOP is TMP_DISed(ST,C:i)

Cause: This message is output for confirmation when combination (C:i) is temporarily disabled by the proximity stop instruction.

Remedy:

SSPC–161 App_STOP is enabled (ST,C:i)

Cause: This alarm is issued in the following cases:

- Case where the program is temporarily stopped with combination (C:i) temporarily disabled by the proximity stop instruction, then is restarted by changing the line
- Case where the program is temporarily stopped with combination (C:i) temporarily disabled by the proximity stop instruction, then is executed after retraction

After this alarm is issued, the combination is not temporarily disabled even if the program is restarted.

Remedy: When the program is restarted by changing the line, reply NO in response to the confirmation message to prevent execution from being disabled.

SSPC–162 App_WAIT is enabled (WT,C:i)

Cause: This message is issued for confirmation when combination (C:i) is enabled by the proximity wait instruction.

Remedy:

SSPC–163 App_WAIT is disabled (WT,C:i)

Cause: This alarm is issued in the following cases:

- Case where the program is temporarily stopped with combination (C:i) enabled by the proximity wait instruction, then is restarted by changing the line
- Case where the program is temporarily stopped with combination (C:i) enabled by the proximity wait instruction, then is executed after retraction

After this alarm is issued, the proximity wait function is disabled even if the program is restarted.

Remedy: When the program is restarted by changing the line, reply NO in response to the confirmation message to prevent execution from being disabled.

SSPC–168 (s,i) invalid group number

Cause: An invalid group number is specified with the proximity stop sensitivity instruction.

Remedy: Specify a correct group number. (s: Program name, i: Line number)

SSPC–169 PAUSE.G (s, i) invalid rate value”

Cause: An invalid sensitivity is specified with the proximity stop sensitivity instruction. (s: Program name, i: Line number)

Remedy: Enter a correct value (0 to 100).

SSPC–181 Comm init error i s

Cause: An error occurred at communication initialization time. (i: Error cause number, s: Control unit name)

Remedy: For the control unit name indicated by the error message, check the address setting, host name, and communication line state.

SSPC–182 Invalid hostname (s)

Cause: After the setting of a new control unit name with a proximity stop or proximity wait combination, an attempt was made to enable the setting before the power is turned off then back on.
Alternatively, an invalid control unit name is specified. (s: Control unit name)

Remedy: When a new control unit name is specified, the power must be turned off then back on for the setting to become effective.
Moreover, check the control unit name on the host communication setting screen.

SSPC–183 Invalid address (s)

Cause: For the control unit name for which this alarm is issued, the communication address setting is incorrect. (s: Control unit name)

Remedy: The control unit name and its address must be checked and modified as required. Next, the power must be turned off then back on for the setting to become effective.

SSPC–184 Number of host exceed limit

Cause: The number of control units specifiable for proximity stop setting and proximity wait setting on one control unit exceeded the limit.

Remedy: Delete unused control units, if any, from the proximity stop setting screen and the proximity wait setting screen.
Alternatively, reduce the number of control units specified.

SSPC–185 Number of element exceed limit

Cause: The number of elements whose settings can be enabled on one control unit exceeded the limit.

Remedy: Check the settings of elements, and disable the settings of those elements that may not be used.
Alternatively, reduce the number of elements whose settings are enabled.

SSPC–186 Invalid element (s,i,j)

Cause: The setting of an element of the control unit indicated by the control unit name in this alarm message is invalid.
(s: Control unit name, i: Element type, j: Element number)
Element type 1 represents the robot, element type 2 represents the hand, and element type 3 represents the jig.

Remedy: Check and modify the setting of the element.

SSPC–187 Receive invalid data i s

Cause: Data received from another control unit contains an error.

Remedy: Check if an error has occurred on the source control unit or hub.

SSPC–188 Invalid data for send i

Cause: Data to be sent to another control unit contains an error.

Remedy: SSPC–111. Open the element setting screen, then check if the settings are correct and also check if all data is displayed correctly. If this alarm is issued even when all elements are set correctly, contact your FANUC customer service representative.

SSPC–189 Timeout element (s,i,j)

Cause: The position data of a remote element received from another control unit is obsolete.
Communication with the control unit may have been disconnected.
(s: Control unit name, i: Element type, j: Element number)

Remedy: If communication is disconnected, the no–response alarm is usually issued.
Turn off the power, then turn on the power.

SSPC–190 No communication (s)

Cause: This alarm is issued when no response is received from another control unit. (s: Control unit name) This function does not communicate with a control unit not specified as a proximity stop or proximity wait target. So, this alarm is issued also when the control unit indicated in the alarm does not have a proximity stop or proximity wait setting made for the target control unit.

Remedy: Check the communication line cabling, and the address, host name, and hub settings.
Moreover, on the remote control unit as well, make a proximity stop or proximity wait setting for the target control unit.

SSPC–191 Target elem not exist(ST,C:i,s)

Cause: The element on the (s) side (L[left], AR[right]) of proximity stop combination (C:i) contains an error. There may be one of the following errors:

- A nonexistent element type or number is specified.
- A nonexistent group is specified.
- All elements of the target $IA_GRP.$ROBOT, $IA_HAND, or $IA_FOBJ are disabled.
- The communication destination control unit does not have a target set correctly for a cause indicated above or because of nonexecution of calibration.

Remedy: Check the items listed above.

SSPC–192 Target elem not exist(PA,C:i,s)

Cause: The element on the (s) side (L[left], AR[right]) of proximity wait combination (C:i) contains an error. There may be one of the following errors:

- A nonexistent element type or number is specified.
- A nonexistent group is specified.
- All elements of the target $IA_GRP.$ROBOT, $IA_HAND, or $IA_FOBJ are disabled.
- The communication destination control unit does not have a target set correctly for a cause indicated above or because of nonexecution of calibration.

Remedy: Check the items listed above.

SSPC–193 IAL detect overload (i)

Cause: The operation, communication processing, proximity stop processing, and proximity wait processing in the control unit are causing an overload.
For the current robot setting, the interpolation period may be is too short.

Remedy: This alarm is not issued usually. Contact your FANUC customer service representative.

CNTR Error Codes

CNTR–004 WARN No cnir pointer

Cause: Internal software error.

Remedy: Contact your FANUC service center.

CNTR–005 WARN Wrong CN Axis/N1 or N2 (G:i)

Cause: The number of the continuous rotation axis is invalid.

Remedy: Set a valid axis number.

CNTR–006 WARN Unable to Allocate Memory

Cause: Internal software error.

Remedy: Contact your FANUC service center.

CNTR–007 STOP.G Serious Internal error (G:i)

Cause: Internal software error.

Remedy: Contact your FANUC service center.

CNTR–008 STOP.G Invalid dest.angle (G:i)

Cause: An operation option that cannot be used with the continuous rotation function is specified.

Remedy: Check the operation option.

CNTR–009 WARN Warn–Cont Vel too high (G:i)

Cause: The continuous rotation speed is relatively high.
Remedy: This does not present any problem. Ignore this message.

CNTR–010 STOP.G Ind.EV option not allowed

Cause: Both an additional axis speed instruction and continuous rotation instruction are used.
Remedy: Delete either instruction.

CNTR–011 STOP.G Axis speed exceeds lim (G:i)

Cause: The continuous rotation speed exceeds the upper limit.
Remedy: Decrease the continuous rotation speed.

CNTR–012 STOP.G Ending Cont Rot on Rel Motion

Cause: The continuous rotation speed instruction ended with a relative operation.
Remedy: Check the operation add instruction used together with the continuous rotation instruction.

RTCP Error Codes

RTCP–001 Wrist Joint is not allow

Cause: Wrist Joint is used on the resume motion.
Remedy: RTCP does not coexist with Wrist Joint. Normally, Wrist Joint is used with the resume motion, so this error always occurs. Change the setting about the resume motion and do not use wrist joint.

D. SYSTEM VARIABLES

This part of this manual describes the names, functions, standard settings, and valid ranges of system variables.

☐ Contents of this appendix

D.1 Format of a System Variable Table

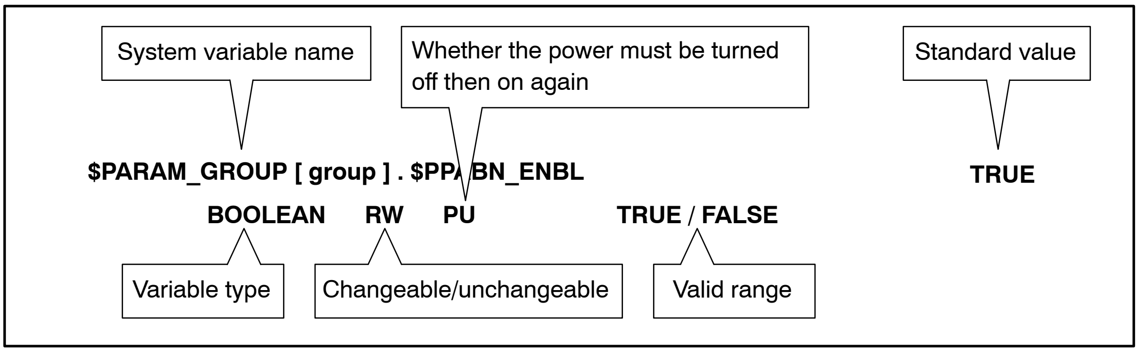


Table D–1. Format of a system variable table

System variable name		
Standard value	* Intrinsic value for each model	
Variable type	BOOLEAN	True/false type (TRUE/FALSE)
	BYTE	Integer (0 to 255)
	SHORT	Integer (–32768 to 32767)
	INTEGER	Integer (–1000000 to 1000000)
	REAL	Real number (–10000000000 to 1000000000)
	CHAR	Character string (“abcdefg”)
	XYZWPR	Cartesian coordinates
Changeable/unchangeable	RW	Changeable
	RO	Unchangeable
Whether the power must be turned off then on again	PU	Indicates that the power must be turned on again.
Valid range (unit)		

Procedure D–1 Setting a system variable

Step

1 Press the MENUS key.

2 Select 0 (NEXT), then select 6 (SYSTEM).

3 Press the F1 (TYPE) key.

4 Select Variables. Then, the system variable screen is displayed.

```
 9 USER
 0 -- NEXT --
```

MENUS

```
 5 POSITION
 6 SYSTEM
 7
```

```
  Variables

  TYPE
```

F1

```
SYSTEM Variables                JOINT 10%
                                    1/98
 1  $AP_MAXAX           536870912
 2  $AP_PLUGGED         4
 3  $AP_TOTALAX         16777216
 4  $AP_USENUM          [12] of Byte
 5  $AUTOINIT           2
 6  $BLT                19920216
 7  $CRT_DEFPROG        *uninit*
 8  $CSTOP              TRUE
 9  $DEFPULSE           4
10  $DEVICE             'P3:'

[TYPE]
```

5 To change the value of a system variable, move the cursor to a desired item, enter a new value, then press the ENTER key or select a desired item by pressing the corresponding function key.

6 When a system variable contains multiple system variables, move the cursor to a desired item and press the ENTER key. Then, the low–order system variables are displayed.

ENTER

```
SYSTEM Variables                JOINT 10%
                                   49/98
47  $ORIENTTOL          10.000
48  $OVRDSLCT           OVRDSLCT_T
49  $PARAM_GROUP        MRR_GRP_T
50  $PASSWORD           PASSWORD_T
```

```
SYSTEM Variables                JOINT 10%
$PARAM_GROUP                       49/98
 1 $BELT_ENABLE         FALSE
 2 $CART_ACCEL1         192
 3 $CART_ACCEL2         0
 4 $CIRC_RATE           1
 5 $CONTAXISNUM         0
 6 $EXP_ENBL            TRUE

[TYPE]
```

7 After changing the setting of the system variable for which PU is specified, turn off the power, then turn it on again. (PU is specified for all $PARAM_GROUP system variables.)

NOTE The setting of a system variable for which RO (unchangeable) is specified cannot be changed.

D.2 System Variables

Hot start

$SEMIPOWERFL **FALSE**

BOOLEAN RW TRUE/FALSE

[Description] Specifies whether to perform a hot start when power is recovered. After a hot start, the robot is restored near the status immediately before a power failure.

TRUE: Performs a hot start after power recovery.

FALSE: Does not perform a hot start. Instead, performs a cold start.

Power failure recovery

$PARAM_GROUP[group] . $SV_OFF_ALL **TRUE**

BOOLEAN RW PU TRUE / FALSE

[Function] Enables or disables the break control function

[Description] Specifies how the brakes are applied.

TRUE: It puts on/off breaks for all axes at same time, i.e. it does not put on all breaks till all axes finish to move and it puts off all breaks when one axis start to move.

FALSE: It puts on/off breaks independently, i.e. it puts on each break which axis finish to move and it never put off each break which axis starts to move.

Mastering

$MASTER_ENB **0**

ULONG RW 1 / 0

[Function] Displays positioning screen

[Description] When this variable is enabled, the positioning screen [6 (SYSTEM).Master/Cal] is displayed on the teach pendant.

0: Positioning screen not displayed.
1: Positioning screen displayed.

$DMR_GRP[group]. $MASTER_DONE **TRUE**

BOOLEAN RW TRUE / FALSE

[Function] Indicates if mastering is completed.

[Description] Indicates if mastering has been completed.

[Setting] On the positioning screen [6 (SYSTEM).Master/Cal]

$DMR_GRP[group]. $MASTER_COUN[1] *****

$DMR_GRP[group]. $MASTER_COUN[2] *****

$DMR_GRP[group]. $MASTER_COUN[3] *****

$DMR_GRP[group]. $MASTER_COUN[4] *****

$DMR_GRP[group]. $MASTER_COUN[5] *****

$DMR_GRP[group]. $MASTER_COUN[6] *****

$DMR_GRP[group]. $MASTER_COUN[7] *****

$DMR_GRP[group]. $MASTER_COUN[8] *****

$DMR_GRP[group]. $MASTER_COUN[9] *****

INTEGER RW 0 to 100000000 (pulse)

[Function] Store mastering pulse counts

[Description] Pulse coder count at zero degree position is stored. This value is calculated from current count at mastering and current position.

$PARAM_GROUP[group]. $MASTER_POS[1]   *
$PARAM_GROUP[group]. $MASTER_POS[2]   *
$PARAM_GROUP[group]. $MASTER_POS[3]   *
$PARAM_GROUP[group]. $MASTER_POS[4]   *
$PARAM_GROUP[group]. $MASTER_POS[5]   *
$PARAM_GROUP[group]. $MASTER_POS[6]   *
$PARAM_GROUP[group]. $MASTER_POS[7]   *
$PARAM_GROUP[group]. $MASTER_POS[8]   *
$PARAM_GROUP[group]. $MASTER_POS[9]   *

REAL RW PU –100000 to 100000 (deg)

[Function] Store jig position for jig mastering

[Description] Jig position for jig mastering is stored. Mastering pulse count is calculated from this data.

Quick mastering

$DMR_GRP[group]. $REF_DONE   FALSE

BOOLEAN RW TRUE / FALSE

[Function] Indicates if setting of the reference point for quick mastering is completed.

[Description] When the reference point of simple mastering is set, the pulse coder count and coordinate values of the reference position are stored.

[Setting] On the positioning screen [6 (SYSTEM).Master/Cal]

$DMR_GRP[group]. $REF_COUNT[1]   0
$DMR_GRP[group]. $REF_COUNT[2]   0
$DMR_GRP[group]. $REF_COUNT[3]   0
$DMR_GRP[group]. $REF_COUNT[4]   0
$DMR_GRP[group]. $REF_COUNT[5]   0
$DMR_GRP[group]. $REF_COUNT[6]   0
$DMR_GRP[group]. $REF_COUNT[7]   0
$DMR_GRP[group]. $REF_COUNT[8]   0
$DMR_GRP[group]. $REF_COUNT[9]   0

INTEGER RW 0 to 100000000 (pulse)

[Function] Store reference point mastering count

[Description] Store the count of the pulse coder when the robot is positioned at the reference point.

$DMR_GRP[group]. $REF_POS[1]   0
$DMR_GRP[group]. $REF_POS[2]   0
$DMR_GRP[group]. $REF_POS[3]   0
$DMR_GRP[group]. $REF_POS[4]   0
$DMR_GRP[group]. $REF_POS[5]   0
$DMR_GRP[group]. $REF_POS[6]   0
$DMR_GRP[group]. $REF_POS[7]   0
$DMR_GRP[group]. $REF_POS[8]   0
$DMR_GRP[group]. $REF_POS[9]   0

REAL RW –100000 to 100000 (deg)

[Function] Store reference point to be set during quick mastering

[Description] Store the reference point to be set during quick mastering.

Positioning

$MOR_GRP[group]. $CAL_DONE   TRUE

BOOLEAN RW TRUE / FALSE

[Function] Indicates if calibration is completed.

[Description] To check the current position of the robot, the count of the pulse coder issued and the current position is calculated using mastering count. This check is usually performed when the power is turned on.

[Setting] On the positioning screen [6 (SYSTEM).Master/Cal]

Specifying coordinate systems

$MNUFRAMENUM[group] **0**

BYTE RW 0 to 9

[Function] Specifies user coordinate system number

[Description] Specifies the number of the user coordinate system currently used.

0: World coordinate system
1 to 9: User coordinate system

[Setting] On the tool coordinate system setting & screen [6 SYSTEM, Coordinate, User]

$MNUFRAME[group, 1] **XYZWPR**

$MNUFRAME[group, 2] **XYZWPR**

$MNUFRAME[group, 3] **XYZWPR**

$MNUFRAME[group, 4] **XYZWPR**

$MNUFRAME[gropu, 5] **XYZWPR**

$MNUFRAME[group, 6] **XYZWPR**

$MNUFRAME[group, 7] **XYZWPR**

$MNUFRAME[group, 8] **XYZWPR**

$MNUFRAME[group, 9] **XYZWPR**

POSITION RW XYZWPR

[Function] Specifies user coordinates system number

[Description] Specifies the Cartesian coordinates in the user coordinate system. Up to nine user coordinate systems can be registered.

$MNUTOOLNUM[group] **0**

BYTE RW 0 to 9

[Function] Specifies tool coordinate system number

[Description] Specifies the number of the tool coordinate system currently used.

0: Mechanical interface coordinate system
1 to 9: Tool coordinate system

[Setting] On the tool coordinate system setting screen [6 SYSTEM.Coordinate.Tool]

$MNUTOOL[group, 1] **XYZWPR**

$MNUTOOL[group, 2] **XYZWPR**

$MNUTOOL[group, 3] **XYZWPR**

$MNUTOOL[group, 4] **XYZWPR**

$MNUTOOL[group, 5] **XYZWPR**

$MNUTOOL[group, 6] **XYZWPR**

$MNUTOOL[group, 7] **XYZWPR**

$MNUTOOL[group, 8] **XYZWPR**

$MNUTOOL[group, 9] **XYZWPR**

POSITION RW XYZWPR

[Function] Specifies the tool coordinate system

[Description] Specify the Cartesian coordinates in the tool coordinate system. Nine tool coordinate systems can be registered.

$JOG_GROUP[group]. $JOG_FRAME **XYZWPR**

POSITION RW XYZWPR

[Function] Specifies the jog coordinate system

[Description] Specifies the Cartesian coordinates in the jog coordinate system.

[Setting] On the jog coordinate system setting screen [6 SYSTEM, Coordinate, jog]

Setting motors

$SCR_GRP[group]. $AXISORDER[1]	**1**
$SCR_GRP[group]. $AXISORDER[2]	**2**
$SCR_GRP[group]. $AXISORDER[3]	**3**
$SCR_GRP[group]. $AXISORDER[4]	**4**
$SCR_GRP[group]. $AXISORDER[5]	**5**
$SCR_GRP[group]. $AXISORDER[6]	**6**
$SCR_GRP[group]. $AXISORDER[7]	**0**
$SCR_GRP[group]. $AXISORDER[8]	**0**
$SCR_GRP[group]. $AXISORDER[9]	**0**

BYTE RW 0 to 16

[Function] Specify axis order

[Description] Specifies the order of axes by assigning the physical number of a servo motor controlled by the servo amplifier (servo register) to the logical number of a robot joint axis specified in software (Jx–axis). For instance, when $AXISORDER[1] = 2, servo motor 2 is assigned to the J1–axis. When $AXISORDER[1] = 0, no servo motor is assigned as the J1–axis.

$SCR_GRP[group]. $ROTARY_AXS[1]	*
$SCR_GRP[group]. $ROTARY_AXS[2]	*
$SCR_GRP[group]. $ROTARY_AXS[3]	*
$SCR_GRP[group]. $ROTARY_AXS[4]	*
$SCR_GRP[group]. $ROTARY_AXS[5]	*
$SCR_GRP[group]. $ROTARY_AXS[6]	*
$SCR_GRP[group]. $ROTARY_AXS[7]	*
$SCR_GRP[group]. $ROTARY_AXS[8]	*
$SCR_GRP[group]. $ROTARY_AXS[9]	*

BOOLEAN RO TRUE / FALSE

[Function] Specify axis type

[Description] Specifies whether joint axes of the robot are rotational or linear.

TRUE: Rotational
FALSE: Linear

$PARAM_GROUP[group]. $MOSIGN[1]	*
$PARAM_GROUP[group]. $MOSIGN[2]	*
$PARAM_GROUP[group]. $MOSIGN[3]	*
$PARAM_GROUP[group]. $MOSIGN[4]	*
$PARAM_GROUP[group]. $MOSIGN[5]	*
$PARAM_GROUP[group]. $MOSIGN[6]	*
$PARAM_GROUP[group]. $MOSIGN[7]	*
$PARAM_GROUP[group]. $MOSIGN[8]	*
$PARAM_GROUP[group]. $MOSIGN[9]	*

BOOLEAN RW PU TRUE / FALSE

[Function] Specify direction of rotation around axes

[Description] Specify whether the robot moves in the positive or negative direction when the motor rotates positively for each axis.

TRUE: The robot moves in a positive direction when the motor rotates positively.
FALSE: The robot moves in a negative direction when the motor rotates positively.

$PARAM_GROUP[group]. $ENCSCALES[1] *
$PARAM_GROUP[group]. $ENCSCALES[2] *
$PARAM_GROUP[group]. $ENCSCALES[3] *
$PARAM_GROUP[group]. $ENCSCALES[4] *
$PARAM_GROUP[group]. $ENCSCALES[5] *
$PARAM_GROUP[group]. $ENCSCALES[6] *
$PARAM_GROUP[group]. $ENCSCALES[7] *
$PARAM_GROUP[group]. $ENCSCALES[8] *
$PARAM_GROUP[group]. $ENCSCALES[9] *

REAL RW PU –10000000000 to 10000000000 (pulse/deg, pulse/mm)

[Function] Specify unit of pulse coder count

[Description] Specify how many pulses are required for the pulse coder when the robot moves around a joint axis one degree or the robot moves along a joint axis 1 mm.
Rotation axis: $ENCSCALES = 2E19 x deceleration ratio/360

$PARAM_GROUP[group]. $MOT_SPD_LIM[1] *
$PARAM_GROUP[group]. $MOT_SPD_LIM[2] *
$PARAM_GROUP[group]. $MOT_SPD_LIM[3] *
$PARAM_GROUP[group]. $MOT_SPD_LIM[4] *
$PARAM_GROUP[group]. $MOT_SPD_LIM[5] *
$PARAM_GROUP[group]. $MOT_SPD_LIM[6] *
$PARAM_GROUP[group]. $MOT_SPD_LIM[7] *
$PARAM_GROUP[group]. $MOT_SPD_LIM[8] *
$PARAM_GROUP[group]. $MOT_SPD_LIM[9] *

INTEGER RW PU 0 to 100000 (rpm)

[Function] Specify maximum motor speed

[Description] Specifies the maximum speed of each servo motor for the robot for each axis. When the robot moves around or along a certain axis at a speed exceeding the maximum speed, a warning is issued. Then, the robot decelerates and moves at a speed not exceeding the maximum speed. In this case, the robot may not trace the specified path.

Override

$SHIFTOV_ENB 0

ULONG RW 0 / 1

[Function] Enables or disables shift override

[Description] The shift override function changes the feedrate override in five steps. To change the feedrate override, press and hold down the SHIFT key, then press the override key as many times as necessary to select the desired override.

1: Enables shift override.

0: Disables shift override.

Press and hold down the SHIFT key, then press the override key: The feedrate override changes in the order: VFINE → FINE → 5% → 50% → 100%.

$MCR. $GENOVERRIDE 10

INTEGER RW 0 to 100 (%)

[Function] Specifies the rate of change in feedrate override

[Description] Specifies the rate of changes in the robot feedrate in percentage.

The feedrate changes in this order:
FINE → VFINE → 0% → 50% → 100%. From 0% to 100% it changes in 5% increments.

[Setting] Use the override keys on the teach pendant.

$MCR. $PROGOVERRIDE 100

INTEGER RW 0 to 100 (%)

[Function] Specifies program override

[Description] Specifies the percentage of the robot feedrate while the program is being played back.

$SCR_GRP . $JOGLIM 12

INTEGER RO 0 to 100%

[Function] Maximum speed scale for coordinate jogging

[Description] Percentage of the maximum speed when jogging the robot in the x, y, or z directions using XYZ or TOOL frame. The maximum speed at linear motion is specified in $PARAM_GROUP[group].$SPEEDLIM.

$SCR . $JOGLIMROT 12

INTEGER RO 0 to 100%

[Function] Maximum speed scale for orientation jogging

[Description] Percentage of the maximum speed when jogging the robot about the x, y, or z axes using XYZ or TOOL frame. The maximum speed at orientation motion is specified in $PARAM_GROUP[group].$ROTSPEEDLIM.

$SCR_GRP[group]. $JOGLIM_JNT[1] *

$SCR_GRP[group]. $JOGLIM_JNT[2] *

$SCR_GRP[group]. $JOGLIM_JNT[3] *

$SCR_GRP[group]. $JOGLIM_JNT[4] *

$SCR_GRP[group]. $JOGLIM_JNT[5] *

$SCR_GRP[group]. $JOGLIM_JNT[6] *

$SCR_GRP[group]. $JOGLIM_JNT[7] *

$SCR_GRP[group]. $JOGLIM_JNT[8] *

$SCR_GRP[group]. $JOGLIM_JNT[9] *

INTEGER RO 0 to 100 (%)

[Function] Specify joint jog override

[Description] The joint jog override function specifies the percentage of the robot feedrate for each axis during jog feed. Specify a low jog override because it is generally unnecessary to move the robot at high speed, and because it is always prudent to avoid danger.

$SCR. $COLDOVRD 10

INTEGER RO 0 to 100 (%)

[Function] Specifies maximum feedrate override after a cold start

[Description] The feedrate override is set to this value after a cold start.

$SCR. $COORDOVRD **10**

INTEGER RO 0 to 100 (%)

[Function] Specifies maximum feedrate override when the manual–feed coordinate system is changed

[Description] The feedrate override is set to this value or less when the manual–feed coordinate system is changed.

$SCR. $TPENBLEOVRD **10**

INTEGER RO 0 to 100 (%)

[Function] Specifies the maximum feedrate override when the teach pendant is enabled

[Description] The feedrate override is set to this value when the teach pendant is enabled.

$SCR. $JOGOVLIM **100**

INTEGER RO 0 to 100 (%)

[Function] Specifies the maximum feedrate override during jog feed

[Description] The feedrate override is set to this value or less during jog feed.

$SCR. $RUNOVLIM **50**

INTEGER RO 0 to 100 (%)

[Function] Specifies the maximum feedrate override when the program is executed

[Description] The feedrate override is set to this value or less when the program is executed.

$SCR. $FENCEOVRD

INTEGER RO 0 to 100 (%)

[Function] Maximum feedrate override when the safety fence is open

[Description] When the safety fence is opened (*SFSPD input is turned off), the feedrate override is set to this value or below.

$SCR. $SFJOGOVLIM **50**

INTEGER RO 0 to 100 (%)

[Function] Maximum feedrate override of jog feed when the safety fence is open

[Description] If jog feed is performed while the safety fence is open, the feedrate override is set to this value or below.

$SCR. $SFRUNOVLIM **30**

INTEGER RO 0 to 100 (%)

[Function] Maximum feedrate override of program execution while the safety fence is open

[Description] When a program is executed with the safety fence open (*SFSPD input set off), the feedrate override is set to this value or below.

$SCR. $RECOV_OVRD **FALSE**

BOOLEAN RW TRUE/FALSE

[Function] Function to restore feedrate override when the safety fence is closed

[Description] When the safety fence is closed (*SFSPD input set on), the previous feedrate override is restored. Then, automatic operation can be started immediately.
This function is enabled when the following conditions are satisfied:
1 $SCR.$RECOV_OVRD is set to TRUE.
2 The system is in the remote state.
3 The feedrate override is not changed while the safety fence is open.
If the safety fence is closed while the above conditions are not satisfied, the previous override cannot be restored.

[Setting] General item setting screen [6 SETTING, GENERAL]

Feedrate

$PARAM_GROUP[group]. $JNTVELLIM[1] *

$PARAM_GROUP[group]. $JNTVELLIM[2] *

$PARAM_GROUP[group]. $JNTVELLIM[3] *

$PARAM_GROUP[group]. $JNTVELLIM[4] *

$PARAM_GROUP[group]. $JNTVELLIM[5] *

$PARAM_GROUP[group]. $JNTVELLIM[6] *

$PARAM_GROUP[group]. $JNTVELLIM[7] *

$PARAM_GROUP[group]. $JNTVELLIM[8] *

$PARAM_GROUP[group]. $JNTVELLIM[9] *

REAL RW PU 0 to 100000 (deg/sec, mm/sec)

[Function] Specify the maximum joint speed

[Description] Specify the maximum joint speed for each axis. When the robot moves around or along a certain axis at a speed exceeding the maximum joint speed, a warning is issued. Then, the robot decelerates and moves at a speed not exceeding the maximum joint speed.

$PARAM_GROUP[group]. $SPEEDLIM 2000

REAL RW PU 0 to 3000 (mm/sec)

[Function] Specifies the maximum linear feedrate

[Description] Specifies the maximum feedrate during linear or circular motion under path control.

$PARAM_GROUP[group]. $ROTSPEEDLIM 90

REAL RW PU 0 to 1440 (deg/sec)

[Function] Specifies the maximum circular feedrate

[Description] Specifies the maximum feedrate during circular motion under attitude control.

Jog feedrate (joint feed) =

$$\text{Maximum joint speed} \times \frac{\text{Joint jog override}}{100} \times \frac{\text{Feedrate override}}{100}$$

Jog feedrate (linear feed) (mm/sec) =

$$\text{Maximum linear feedrate} \times \frac{\text{Jog override}}{100} \times \frac{\text{Feedrate override}}{100}$$

Jog feedrate (circular feed) (mm/sec) =

$$\text{Maximum circular feedrate} \times \frac{\text{Jog override}}{100} \times \frac{\text{Feedrate override}}{100}$$

Joint jog override	$SCR_GRP . $JOGLIM_JNT [i] (%)
Jog override	$SCR . $JOGLIM (%)
Maximum joint speed	$PARAM_GROUP . $JNTVELLIM
Maximum linear feedrate	$PARAM_GROUP . $SPEEDLIM (mm/sec)
Maximum circular feedrate	$PARAM_GROUP . $ROTSPEEDLIM (deg/sec)

Operation speed (joint motion) =

$$\text{Maximum joint speed} \times \frac{\text{Coefficient of joint speed}}{200} \times \frac{\text{Programmed speed}}{100} \times \frac{\text{Programmed override}}{100} \times \frac{\text{Feedrate override}}{100}$$

Operation speed (linear motion) (mm/sec) =

$$\text{Programmed speed} \times \frac{\text{Programmed override}}{100} \times \frac{\text{Feedrate override}}{100}$$

Operation speed (circular motion) (deg/sec) =

$$\text{Programmed speed} \times \frac{\text{Programmed override}}{100} \times \frac{\text{Feedrate override}}{100}$$

Programmed override $MCR_GRP . $PROGOVERRIDE (%)
Coefficient of joint speed $PARAM_GROUP . $SPEEDLIMJNT

$PARAM_GROUP[group]. $LOWERLIMS[1] *
$PARAM_GROUP[group]. $LOWERLIMS[2] *
$PARAM_GROUP[group]. $LOWERLIMS[3] *
$PARAM_GROUP[group]. $LOWERLIMS[4] *
$PARAM_GROUP[group]. $LOWERLIMS[5] *
$PARAM_GROUP[group]. $LOWERLIMS[6] *
$PARAM_GROUP[group]. $LOWERLIMS[7] *
$PARAM_GROUP[group]. $LOWERLIMS[8] *
$PARAM_GROUP[group]. $LOWERLIMS[9] *

REAL RW PU –100000 to 100000 (deg, mm)

[Function] Specify the lower limit of the joint operating area

[Description] Specify the lower limit of the joint operating area which is the limit of the motion in the negative direction.

[Setting] Joint operating area screen [6 (SETTING).Joint Area]

$PARAM_GROUP[group]. $UPPERLIMS[1] *
$PARAM_GROUP[group]. $UPPERLIMS[2] *
$PARAM_GROUP[group]. $UPPERLIMS[3] *
$PARAM_GROUP[group]. $UPPERLIMS[4] *
$PARAM_GROUP[group]. $UPPERLIMS[5] *
$PARAM_GROUP[group]. $UPPERLIMS[6] *
$PARAM_GROUP[group]. $UPPERLIMS[7] *
$PARAM_GROUP[group]. $UPPERLIMS[8] *
$PARAM_GROUP[group]. $UPPERLIMS[9] *

REAL RW PU –100000 to 100000 (deg, mm)

[Function] Specify the upper limit of the joint operating area

[Description] Specify the upper limit of the joint operating area, which is the limit of the motion in the positive direction.

[Setting] Joint operating area screen [6 (SETTING).Joint Area]

Payload specification

If load information has not been set up on the load setting screen, it is necessary to enter the following information. If no load setting condition number has been selected, be sure to enter the information correctly, since the robot uses it when it runs. Therefore, be particularly careful when setting these values.

- $GROUP[group].$PAYLOAD
- $PARAM_GROUP[group].$PAYLOAD
- $PARAM_GROUP[group].$PAYLOAD_X
- $PARAM_GROUP[group].$PAYLOAD_Y
- $PARAM_GROUP[group].$PAYLOAD_Z
- $PARAM_GROUP[group].$PAYLOAD_IX
- $PARAM_GROUP[group].$PAYLOAD_IY
- $PARAM_GROUP[group].$PAYLOAD_IZ
- $PARAM_GROUP[group].$AXISINTERTIA[1 to 9]
- $PARAM_GROUP[group].$AXISMOMENT[1 to 9]
- $PARAM_GROUP[group].$AXIS_IM_SCL
- $PARAM_GROUP[group].$ARMLOAD[1 to 3]

$GROUP [group] . $PAYLOAD *

REAL RW 0 to 10000(kgf)

[Function] Payload

[Description] Specify a payload. If the load varies during an operation, specify the maximum value.

$PARAM_GROUP [group] . $PAYLOAD *

REAL RW PU 0 to 10000(kgf)

[Function] Payload

[Description] Specify a payload. If the load varies during an operation, specify the maximum value.

$PARAM_GROUP[group]. $PAYLOAD_X *

$PARAM_GROUP[group]. $PAYLOAD_Y *

$PARAM_GROUP[group]. $PAYLOAD_Z *

REAL RW PU –100000 to 10000(cm)

[Function] Load gravity center distance

[Description] Center of gravity of load viewed on the mechanical interface coordinate system (default tool coordinate system). The center of gravity of a load is measured along the X–axis, Y–axis, and Z–axis of the mechanical interface coordinate system.

$PARAM_GROUP[group]. $PAYLOAD_IX *

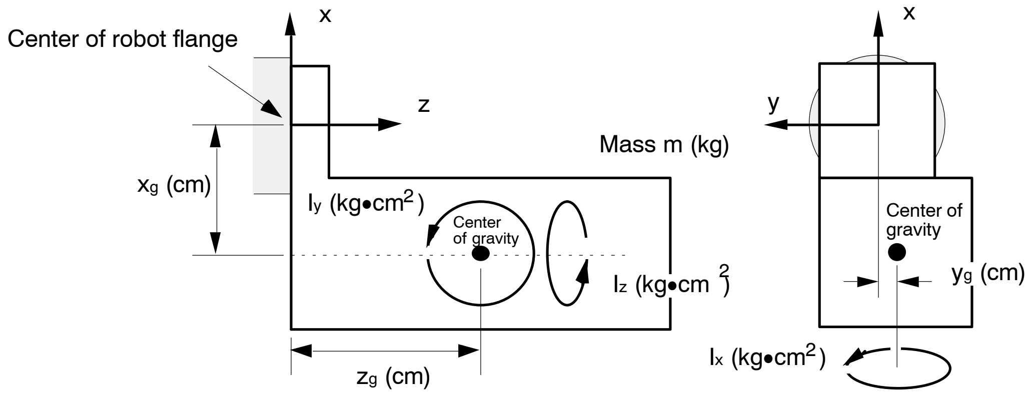
$PARAM_GROUP[group]. $PAYLOAD_IY *

$PARAM_GROUP[group]. $PAYLOAD_IZ *

REAL RW PU 0 to 10000(kg • cm2)

[Function] Load gravity center inertia

[Description] Inertia around the center of gravity of load. The inertia of a heavy load is calculated around the X–axis, Y–axis, and Z–axis of the mechanical interface coordinate system.
The meaning of $PARAM_GROUP[group].$PAYLOAD_* is as illustrated below:



xg (cm)	: PARAM_GROUP[group].$PAYLOAD_X
yg (cm)	: PARAM_GROUP[group].$PAYLOAD_Y
zg (cm)	: PARAM_GROUP[group].$PAYLOAD_Z
Ix (kg•cm^2)	: PARAM_GROUP[group].$PAYLOAD_IX
Iy (kg•cm^2)	: PARAM_GROUP[group].$PAYLOAD_IY
Iz (kg•cm^2)	: PARAM_GROUP[group].$PAYLOAD_IZ

$PARAM_GROUP [group]. $AXISINERTIA[1] *
$PARAM_GROUP [group]. $AXISINERTIA[2] *
$PARAM_GROUP [group]. $AXISINERTIA[3] *
$PARAM_GROUP [group]. $AXISINERTIA[4] *
$PARAM_GROUP [group]. $AXISINERTIA[5] *
$PARAM_GROUP [group]. $AXISINERTIA[6] *
$PARAM_GROUP [group]. $AXISINERTIA[7] *
$PARAM_GROUP [group]. $AXISINERTIA[8] *
$PARAM_GROUP [group]. $AXISINERTIA[9] *

SHORT RW PU 0 to 32767 (kgf · cm · sec^2)

[Function] Payload inertia

[Description] For each axis, specify an integer as the value of the inertia resulting from the applied payload. The values for the 1st to 3rd axes are calculated automatically; therefore, they need not be specified. (Set a value for each of the 4th, 5th, and 6th axes.)
The inertia for each axis is calculated using the following expression:

$$\$AXISINERTIA[\,i\,] = \frac{\text{payload} \times (\text{l\_max}[\,i\,])^2}{g} \quad (\text{kgf} \cdot \text{cm} \cdot \text{sec}^2)$$

payload : Payload [kgf]

l_max[i]: Maximum distance from the rotation center of the axis (axis i) to the mass center of the load on the robot [cm]
For the 4th and 5th axes, the distance may vary depending on the angle of the other axes. In such a case, set the maximum distance than can be achieved.

g: Gravity acceleration (= 980 [cm/sec2])

NOTE When specifying or changing this variable, refer to the explanation of $PARAM_GROUP[].$AXIS_IM_SCL, below.

$PARAM_GROUP [group]. $AXISMOMENT[1] *
$PARAM_GROUP [group]. $AXISMOMENT[2] *
$PARAM_GROUP [group]. $AXISMOMENT[3] *
$PARAM_GROUP [group]. $AXISMOMENT[4] *
$PARAM_GROUP [group]. $AXISMOMENT[5] *
$PARAM_GROUP [group]. $AXISMOMENT[6] *
$PARAM_GROUP [group]. $AXISMOMENT[7] *
$PARAM_GROUP [group]. $AXISMOMENT[8] *
$PARAM_GROUP [group]. $AXISMOMENT[9] *

SHORT RW PU 0 to 32767 (kgf · m)

[Function] Axis moment

[Description] For each axis, specify an integer as the moment value resulting from the applied payload. The values for the 1st to 3rd axes are calculated automatically; therefore, they need not be specified. (Set a value for each of the each of 4th, 5th, and 6th axes.)
The moment value for each axis is calculated using the following expression:

$AXISMOMENT [i] = payload × l_max [i] (kgf · m)

payload : Payload [kgf]
l_max[i]: Maximum distance from the rotation center of the axis (axis i) to the mass center of the load on the robot [m]
For the 4th and 5th axes, the distance may vary depending on the angle of the other axes. In such a case, set the maximum distance than can be achieved.

NOTE When specifying or changing this variable, refer to the explanation of $PARAM_GROUP[].$AXIS_IM_SCL, below.

[Function] Inertia and moment value adjustment scale

[Description] This scale is used to set up a number in decimal places for the inertia and moment values of each axis stated above.

NOTE It is usually unnecessary to re–set this variable.
Actually, the following inertia and moment values are used.

$$\text{(Inertia value)} = \frac{\text{\$PARAM\_GROUP[group].\$AXISINERTIA[i]}}{\text{\$PARAM\_GROUP[group].\$AXIS\_IM\_SCL}}$$

$$\text{(Moment value)} = \frac{\text{\$PARAM\_GROUP[group].\$AXISMOMENT[i]}}{\text{\$PARAM\_GROUP[group].\$AXIS\_IM\_SCL}}$$

It is therefore necessary to assign $AXISINERTIA[i] and $AXISMOMENT[i] with values that match the setting of this variable.
To enter the value "1.23," for example, as the inertia value for the fourth axis of the robot:

- Set up $PARAM_GROUP[group].$AXIS_IM_SCL = 100
- Set up $PARAM_GROUP[group].$AXISINERTIA[4] = 123
- Change these inertia and moment values for other axes according to the value of $AXIS_IM_SCL.

$PARAM_GROUP [group]. $ARMLOAD[1] *
$PARAM_GROUP [group]. $ARMLOAD[2] *
$PARAM_GROUP [group]. $ARMLOAD[3] *

REAL RW PU 0 to 10000 (kgf)

[Function] Equipment weight

[Description] When equipment such as welding equipment is installed on a robot axis, specify the payload incurred by that equipment.
$ARMLOAD[1]: Specify the weight of the equipment installed on the 3rd–axis arm.
$ARMLOAD[2]: Specify the weight of the equipment installed on the 2nd–axis base.
$ARMLOAD[3]: Not used.

Executing a program

$DEFPULSE **4**

SHORT RW 0 to 255 (100 msec)

[Function] Specifies the standard DO output pulse width

[Description] This value is used when the pulse width is not specified for the output of a DO signal pulse.

Automatic operation

$RMT_MASTER **0**

INTEGER RW 0 to 3

[Function] Specifies which remote unit is used

[Description] Specifies which remote unit is used. The specified remote unit has the right to start the robot.

0: Peripheral unit (remote controller)
1: CRT/keyboard
2: Host computer
3: No remote unit

Deleting the warning history

$ER_NOHIS **0**

BYTE RW 0 / 3

[Function] Warning history delete function

[Description] WARN alarms, NONE alarms and resets can be deleted from the alarm history.
0: Disables the function. (All alarms and resets are recorded in the history.)
1: Does not record WARN and NONE alarms in the history.
2: Does not record resets.
3: Does not record resets, WARN alarms, and NONE alarms.

Disabling alarm output

$ER_NO_ALM. $NOALMENBL **0**

BYTE RW 0 / 1

[Function] Enables the no–alarm output function

[Description] When this function is enabled, the LEDs on the teach pendant and the machine operator's panel corresponding to the alarms specified with system variable $NOALM_NUM do not light. In addition, the peripheral I/ O alarm signal (FAULT) is not output.

$ER_NO_ALM. $NOALM_NUM **5**

BYTE RW 0 to 10

[Function] Specifies the number of alarms not output

[Description] Specifies the number of alarms that are not output.

$ER_NO_ALM. $ER_CODE1 11001
$ER_NO_ALM. $ER_CODE2 11002
$ER_NO_ALM. $ER_CODE3 11003
$ER_NO_ALM. $ER_CODE4 11007
$ER_NO_ALM. $ER_CODE5 11037
$ER_NO_ALM. $ER_CODE6 0
$ER_NO_ALM. $ER_CODE7 0
$ER_NO_ALM. $ER_CODE8 0
$ER_NO_ALM. $ER_CODE9 0
$ER_NO_ALM. $ER_CODE10 0

INTEGER RW 0 to 100000

[Function] Specify the alarms not output

[Description] Specify the alarms that are not output.

Setting : 11 002 (Meaning: SERVO–002 alarm)
Alarm ID Alarm number

Error code output

$ER_OUT_PUT. $OUT_NUM 0

LONG RW 0 to 512

[Function] SDO start number for error code output

[Description] Specify the start number for the SDOs used for error code output. An error code is output, in binary format, using 33 SDOs starting from that having the specified number. If 0 is specified, no error code is output.

$ER_OUT_PUT. $IN_NUM 0

LONG RW 0 to 512

[Function] SDI number for error code output request

[Description] Every time the SDI specified in this variable is set to ON, an error code is output to the SDOs specified in $ER_OUTPUT.$OUT_NUM, explained above.

User alarm

$UALRM_SEV[ ] 6

BYTE RW 0 to 255

[Function] User alarm severity

[Description] Sets the user alarm severity. $UALRM_SEV[i] corresponds to the severity of user alarm [i].

0 WARN
6 STOP.L
38 STOP.G
11 ABORT.L
43 ABORT.G

The initial severity for each user alarm is 6 (STOP.L).

Jogging

$JOG_GROUP . $FINE_DIST 0.5

REAL RW 0.0 to 1.0 (mm)

[Function] Move distance for linear step jogging

[Description] Specify an amount of travel in low–speed linear step feed by Cartesian/tool manual feed. The amount of travel in very low speed step feed is one tenth of the value specified here.

$SCR . $FINE_PCNT 10

INTEGER RO 1 to 100 %

[Function] Move distance for joint or orientation step jogging

[Description] Specify an amount of travel for step feed in attitude rotation by axial manual feed or Cartesian/tool manual feed. Specify manual feed with a precentage and an override of 1%.

I/O setting

$OPWORK . $UOP_DISABLE *****

BYTE RW 0 / 1

[Function] Enable/disable UOP I/O

[Description] Specify whether the peripheral equipment input signal is enabled or disabled. If the peripheral equipment input signal is enabled when the robot is operated without any peripheral equipment connected, an alarm cannot be cleared. By disabling the signal with this setting, the alarm can be cleared. When any peripheral equipment is connected, set this variable to 0 before using that equipment.

$SCR . $RESETINVERT **FALSE**

BOOLEAN RW TRUE / FALSE

[Function] FAULT_RESET input signal detection.

[Description] When you set this value to "TRUE", an error is reset by rising edge of FAULT_RESET input signal. If "FALSE" is set, an error is reset by falling edge is detected.

TRUE: Check rising edge of reset input signal.

FALSE: Check falling edge of reset input signal.

$PARAM_GROUP . $PPABN_ENBL **FALSE**

BOOLEAN RW TRUE / FALSE

[Function] Enable/disable pressure abnormal *PPABN input

[Description] Specifies if pressure abnormal signal is detected or not. If you want to use *PPABN input, you should set this variable to TRUE.

TRUE: Enable

FALSE: Disable

$PARAM_GROUP. $BELT_ENBLE **FALSE**

BOOLEAN RW TRUE / FALSE

[Function] Belt rupture signal enabled/disabled

[Description] Specify whether the belt rupture signal (RDI[7]) is detected. For a robot utilizing the belt rupture signal (A–510, L–1000), this value is automatically set to TRUE.
TRUE: Belt rupture signal enabled
FALSE: Belt rupture signal disabled

Software version

$ODRDSP_ENB **0**

ULONG RW 1 / 0

[Function] Display of an order file

[Description] An order listing, showing the configuration of the software components installed in the controller can be displayed on the display (order file screen) of the teach pendant.

Soft float function

$SFLT_ERRTYP **0**

INTEGER RW 1 to 10

[Function] Flag for specifying the alarm to be generated when time–out occurs during follow–up processing of the soft float function

[Description] This variable specifies the alarm (a servo alarm or program pause alarm) to be generated if a time–out occurs during follow–up processing of the soft float function.
0: Generates servo alarm "SRVO–111 Softfloat time out."
1: Generates program pause alarm "SRVO–112 Softfloat time out."

$SFLT_DISFUP **FALSE**

BOOLEAN RW TRUE / FALSE

[Function] Specifies whether to perform follow–up processing at the start of each motion instruction.

[Description] Specify whether to perform follow–up processing of the soft float function at the start of each program motion instruction.
TRUE: Does not perform follow–up processing at the start of each program motion instruction.
FALSE: Performs follow–up processing at the start of each program motion instruction.

Saving files

$FILE_APPBCK

[Function] On the file screen, displays the name of a file to be saved as Application.

$FILE_SYSBCK

[Description] On the file screen, displays the name of a file to be saved as System file.

Register speed specification function

$RGSPD_PREXE **FALSE**

BOOLEAN RO TRUE/FALSE

[Function] Advanced register speed read enabled or disabled

[Description] Specify whether an advanced read of operation statement is performed (enabled) or not (disabled) when the movement speed specified by an operation statement is held in a register.
TRUE: Advanced read enabled
FALSE: Advanced read disabled

> **NOTE**
> When an advanced register speed read is enabled with the setting indicated above, the timing at which the register value is changed is important. With some timings, a change in the register value may not be reflected in the operation speed, and the register value existing before the change may be applied to the movement. To enable advanced register speed read, some consideration is needed: The value of a register used for the movement speed during program execution should not be changed; An interlock should be provided.

Specifying an output signal of the BLAL/BZAL alarm

$BLAL_OUT.$DO_INDEX **0**

INTEGER RW 0 to 256

[Description] When a non–zero number is specified, SDO corresponding to that number is turned on at the occurrence of BLAL/BAZL. SDO stays on until the voltage is restored by the replacement backup battery or some other means. (If a program or the I/O screen is used to turn off SDO forcibly, SDO is turned back on immediately.)

$BLAL_OUT.$BATALM_OR **FALSE**

INTEGER RW TRUE/FALSE

[Description] Specifies whether to set BATALM, a dedicated output signal, so that it has also the BZAL/BLAL function.

Index

[I]

[J]

[L]

[M]

[N]

[O]

[P]

[Q]

[R]

[S]

[T]

[U]

[V]

[W]

Revision Record

FANUC Robot series (R–J3 *i* Mate) LR HANDLING TOOL OPERATOR'S MANUAL (B–81524EN)

01	Aug., 2001				
Edition	Date	Contents	Edition	Date	Contents